



# **DIMS Administrator Guide**

***Release 0.1.18***

**David Dittrich**

**Dec 05, 2017**



---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Overview . . . . .	3
<b>2</b>	<b>Referenced documents</b>	<b>5</b>
<b>3</b>	<b>Onboarding Developers</b>	<b>7</b>
3.1	Initial Account Setup . . . . .	7
3.2	GPG Encryption Keys for Email, etc. . . . .	8
3.3	Creating accounts . . . . .	8
3.4	Installing initial SSH key(s) . . . . .	8
3.5	Remote Account Setup . . . . .	9
3.6	JIRA Onboarding . . . . .	13
<b>4</b>	<b>Installation of DIMS Components on “Bare-metal”</b>	<b>19</b>
4.1	Control and Target Prerequisites . . . . .	19
4.2	Setting up a DIMS Developer Laptop . . . . .	20
<b>5</b>	<b>Installation of DIMS Components Using Virtual Machines</b>	<b>31</b>
5.1	DIMS on Virtual Machines . . . . .	31
5.2	Prerequisites for Instantiating Virtual Machines . . . . .	32
5.3	VM Build Workflow . . . . .	33
5.4	Run Directory Helper Makefile Targets . . . . .	34
<b>6</b>	<b>Installation of a Complete DIMS Instance</b>	<b>39</b>
6.1	Cluster Foundation Setup . . . . .	39
6.2	Bootstrapping User Base . . . . .	40
<b>7</b>	<b>Trident</b>	<b>41</b>
7.1	Installing Trident manually . . . . .	41
7.2	Installing Trident with Ansible . . . . .	46
7.3	Trident Prerequisites . . . . .	48
7.4	Install Trident . . . . .	58
7.5	Running Trident . . . . .	58
7.6	Using tcli on the command line . . . . .	59
7.7	Configuring Trident via web app . . . . .	63
7.8	Upgrading configuration across Trident versions . . . . .	105
7.9	Emails and other non-official documentation . . . . .	110

<b>8</b>	<b>AMQP and RabbitMQ</b>	<b>113</b>
8.1	RabbitMQ use in DIMS . . . . .	113
8.2	Basic Service Administration . . . . .	113
8.3	Managing RabbitMQ . . . . .	114
8.4	Management with Ansible playbooks . . . . .	124
<b>9</b>	<b>RaspberryPi and Docker</b>	<b>125</b>
9.1	Installing HypriotOS w/Docker . . . . .	125
9.2	Installing a Persistent Docker Container . . . . .	128
<b>10</b>	<b>Docker Datacenter</b>	<b>135</b>
10.1	Initial Inquiry . . . . .	135
10.2	Docker Trusted Repository Issues . . . . .	135
10.3	Further Information . . . . .	135
<b>11</b>	<b>Managing Long-running Services</b>	<b>141</b>
11.1	Services using supervisord . . . . .	142
11.2	Services using Upstart . . . . .	142
<b>12</b>	<b>Diagnosing System Problems and Outages</b>	<b>157</b>
12.1	Using dimscli . . . . .	157
12.2	Debugging Vagrant . . . . .	164
12.3	Other Tools for Diagnosing System Problems . . . . .	167
<b>13</b>	<b>Managing CoreOS with Systemd and Other Tools</b>	<b>171</b>
13.1	State of systemd . . . . .	171
13.2	State of systemd units . . . . .	177
13.3	Managing systemd units . . . . .	189
<b>14</b>	<b>Managing Virtualbox VMs</b>	<b>191</b>
14.1	Remotely Managing Virtualbox . . . . .	191
<b>15</b>	<b>Appendices</b>	<b>195</b>
15.1	Add New Connection to Apache Directory Studio . . . . .	195
<b>16</b>	<b>Contact</b>	<b>199</b>
<b>17</b>	<b>License</b>	<b>201</b>



This document (version 0.1.18) covers issues related to system administration of DIMS components from an administrator's perspective.



This chapter introduces the system administration policies, methodology for configuration file management, automated installation and configuration of DIMS components using Ansible, and use of continuous integration mechanisms used for deployment and testing of DIMS components.

This document is closely related to the [DIMS Developer Guide v 1.0.0](#), which covers a number of related tasks and steps that will not be repeated here (rather, will be cross-referenced using [intersphinx](#) links.)

- All documentation for the DIMS project is written using restructured text (reST) and Sphinx. Section [Documenting DIMS Components](#) of the [DIMS Developer Guide v 1.0.0](#) covers how to use these tools for producing professional looking and cross-referenced on-line (HTML) and off-line (PDF) documentation.
- DIMS software – including Ansible playbooks for installation and configuration of DIMS system components, Packer, Vagrant, and Docker subsystem creation scripts, are all maintained under version control using Git and the HubFlow methodology and tool set. Section [Source Code Management with Git](#) of the [DIMS Developer Guide v 1.0.0](#) covers how these tools are used for source code, documentation, and system configuration files.
- Changes to source code that are pushed to Git repositories trigger build processes using the Jenkins continuous integration environment. These triggers build and/or deploy software to specified locations, run tests, and/or configure service components. In most cases, Ansible is used as part of the process driven by Jenkins. Section [Continuous Integration](#) of the [DIMS Developer Guide v 1.0.0](#) provides an overview of how this works and how to use it in development and testing DIMS components.
- System software installation and configuration of DIMS components are managed using Ansible playbooks that are in turn maintained in Git repositories. Only a bare minimum of manual steps are required to bootstrap a DIMS deployment. After that, configuration changes are made to Git repositories and those changes trigger continuous integration processes to get these changes into the running system. Section [Deployment and Configuration](#) of the [DIMS Developer Guide v 1.0.0](#) covers how to use this framework for adding or managing the open source components that are used in a DIMS deployment.

## 1.1 Overview

This document is focused on the system administrative tasks that are involved in adding open source software components to the DIMS framework, how to convert installation instructions into Ansible playbooks or `Dockerfile`

instructions that can be used to instantiate a service or microservice, how a complete DIMS instance (i.e., a complementary set of service and microservice components that function together as a coherent system) is installed, configured, debugged and/or tuned, and kept in running order over time.

## CHAPTER 2

---

### Referenced documents

---

1. [DIMS Developer Guide v 1.0.0](#)
2. `ansibleinventory:ansibleinventory`
3. `ansibleplaybooks:ansibleplaybooks`
4. `dimsdockerfiles:usingdockerindims`
5. `dimsdockerfiles:dockerincoreos`
6. `dimspacker:dimspacker`
7. `dimsciutils:dimsciutilities`
8. `dimssr:dimssystemrequirements`
9. [DIMS Architecture Design v 2.10.0](#)
10. `dittrich:homepage` home page.



---

## Onboarding Developers

---

This chapter covers the process for onboarding new developers to provide them access to DevOps components necessary to work on elements of a DIMS deployment. In short, developers (and system administrators) will need the following:

- An account in the Trident portal system for access to email lists, etc.
- A GPG/PGP key pair. The public key will be loaded into the Trident portal so others can access the key and so it can be used for encrypted email.
- A Google account for OpenID Connect authentication used for single-signon access to internal resources, along with an LDAP database entry that links to this Google account.
- SSH public/private key pairs allowing access to Git repositories, Ansible control host, DIMS system components, etc.
- Initial copies of Git repositories used to develop and build a DIMS deployment instance.

Once all of these resources have been procured, developers or system administrators are ready to work on a DIMS instance.

### 3.1 Initial Account Setup

The first step in adding a new DIMS developer is getting them set up with an account on our internal `ops-trust` portal instance.

---

**Note:** We will transition to using Trident, rather than the old Ops-Trust portal code base initially set up for this project, as soon as we are able. Trident has an internal wiki, so the FosWiki server mentioned here will also be retired.

---

Our FosWiki server has a page that was dedicated to the steps necessary for [Provisioning New DIMS Users](#).

**Caution:** The FosWiki page [Provisioning New DIMS Users](#) looks like it may be out of date, or include steps that may not be necessary for just adding a new user. It has a huge number of steps that should be made more streamlined or added to the DIMS web app to simplify the process of adding and removing DIMS users in concert with the `ops-trust` portal at the center of DIMS.

Once the user has been given their password to the `ops-trust` portal, they need to change their `MemberID` to match the account name that should be used within DIMS. (E.g., Dave Dittrich may be given the `MemberID` of `davedittrich2475` by the portal, but the desired account name within DIMS subsystems should be `dittrich`.)

## 3.2 GPG Encryption Keys for Email, etc.

Each `ops-trust` portal account holder needs a GPG key to be able to send/receive encrypted emails. In normal operation, one's `ops-trust` portal account is not fully enabled until the user has uploaded their GPG key.

One of the easiest ways to process GPG-encrypted email is using [Enigmail](#) with the [The GNU Privacy Guard](#) from the [Thunderbird](#) email client. Follow the [Enigmail Quick Start Guide](#) to install, configure, and generate a GPG key for use with [Thunderbird](#) (which is supported on Mac, Linux, and Windows, and is installed by default on the DIMS Ubuntu developer laptops).

After you have set up [The GNU Privacy Guard](#) and uploaded your key, log in to the `ops-trust` portal and select `PGP Keys` from the menu on the left of the screen to download all GPG keys for other portal users and all email lists to which you subscribe.

---

**Note:** This step will only download keys that are in the system at the time you press the link, which means they will get out-of-date with respect to new users, regenerated keys, and/or new email lists that may be created over time. Get in the habit of updating your GPG key ring regularly, or at least remember that failure to encrypt/decrypt and email may be due to your keyring being out of date and needing a refresh.

---

## 3.3 Creating accounts

After a new user has successfully set up their `ops-trust` portal account and modified their `MemberID` to align with their desired DIMS account name, they must be added to the `dims_users` array in the `$GIT/ansible-playbooks/group_vars/all` file. Once added, the Ansible playbook roles that generate DIMS user accounts (e.g., `dims-users-create`) can be played to create accounts as needed.

## 3.4 Installing initial SSH key(s)

Before someone can clone Git repositories, or use SSH to log in to DIMS systems for interactive shell access, they must (a) have a DIMS SSH key, and (b) have the public key and `authorized_keys` file(s) on target systems set up properly.

1. Create the user's DIMS SSH key pair...
2. Generate accounts using Ansible playbook (`$whatever`), which creates the accounts and installs their public key.
3. Copy their key pair into the account on the system where they will be doing their development (i.e., a DIMS developer laptop, Vagrant virtual machine, or bare-metal workstation.) Also make sure their key is included in the `authorized_keys` file in the `git` account on `git.devops.develop` in order for them to be able to read/write source code using Git.



4. Trigger a Jenkins build job for `public-keys-configure` to push the new user's key to all DIMS-DevOps and DIMS-OPS systems.
5. Set the password on the account they are supposed to use so they can log in to it, and/or securely transfer their public SSH key to them so they can use it to access the account without needing a password.

---

**Note:** They will need a password on the account for `sudo` on commands like `dims-ci-utils.install`. user that ask for the `sudo` password in order to pass it to Ansible.

---

Use command `passwd <username>`.

```
[dimsenv] mboggess@b52:~ () $ passwd mboggess
Changing password for mboggess.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

## 3.5 Remote Account Setup

This section details how to set up a new account for a current developer on a remote machine, after being logged in to the remote machine.

### 3.5.1 Change password

Use command `passwd <username>`.

```
[dimsenv] mboggess@b52:~ () $ passwd mboggess
Changing password for mboggess.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

### 3.5.2 Transfer SSH Keys to Remote Machine

- Once logged in to remote machine, check `~/.ssh/authorized_keys` file for public key:

```
[dimsenv] mboggess@b52:~ () $ cd .ssh
[dimsenv] mboggess@b52:~/.ssh () $ ls
authorized_keys  config  known_hosts
[dimsenv] mboggess@b52:~/.ssh () $ vim authorized_keys
```

- Securely transfer DIMS RSA keys from local machine to remote machine

Keys are located in `~/.ssh/` and should be named:

- `dims_${dimsusername}_rsa` for private key
- `dims_${dimsusername}_rsa.pub` for public key
- `dims_${dimsusername}_rsa.sig` for signature

Copy all three files from local machine with DIMS RSA keys:

```
[dimsenv] mboggess@dimsdev2:~ () $ cd .ssh
[dimsenv] mboggess@dimsdev2:~/ssh () $ scp dims_mboggess_rsa* mboggess@b52.
↪tacoma.uw.edu:/home/mboggess/.ssh/
dims_mboggess_rsa                100% 1675      1.6KB/s  ↪
↪00:00
dims_mboggess_rsa.pub            100%  403      0.4KB/s  ↪
↪00:00
dims_mboggess_rsa.sig            100%   82      0.1KB/s  ↪
↪00:00
```

Check on remote machine:

```
[dimsenv] mboggess@b52:~/ssh () $ ls
authorized_keys  dims_mboggess_rsa      dims_mboggess_rsa.sig
config           dims_mboggess_rsa.pub  known_hosts
```

---

**Note:** This solves the “second hop issue”: a user can access machines one hop away because the necessary keys are available on their local machine, but when trying to go one hop further, keys are not available. For example, I can log in to b52 just fine, but when I try to run `dims.git.syncrepos`, which requires access to `git.devops.develop`, I ran into trouble because my keys were not on b52.

---

### 3.5.3 Sync Repos on Remote Machine

There probably will not be a `.mrconfig` file on the remote machine, so you must create an empty file with that name before you sync repos or the command will fail.

Failure when running `dims.git.syncrepos` because no `.mrconfig`:

```
<snip>

[+++] Adding Repo[49] umich-botnets to /home/mboggess/dims/.mrconfig and checking it
↪out.
cp: cannot stat '/home/mboggess/dims/.mrconfig': No such file or directory

[+++] Updated 49 of 49 available repos.
[+++] Summary of actions for repos that were updated:
- Any changes to branches at origin have been downloaded to your local repository
- Any branches that have been deleted at origin have also been deleted from your
↪local repository
- Any changes from origin/master have been merged into branch 'master'
- Any changes from origin/develop have been merged into branch 'develop'
- Any resolved merge conflicts have been pushed back to origin
[+++] Added 49 new repos: ansible-inventory ansible-playbooks cif-client cif-java
↪configs dims dims-ad dims-adminguide dims-asbuilt dims-ci-utils dims-dashboard dims-
↪db-recovery dims-devguide dims-dockerfiles dims-dsdd dims-jds dims-keys dims-ocd
↪dims-packer dims-parselogs dims-sample-data dims-sr dims-supervisor dims-svd
↪dimssysconfig dims-test-repo dims-tp dims-tr dims-vagrant ELK fuse4j ipgrep java-
↪native-loader java-stix-v1.1.1 mal4s MozDef ops-trust-openid ops-trust-portal
↪poster-deck-2014-noflow prisem prisem-replacement pygraph rwfind sphinx-autobuild
↪stix-java ticketing-redis tsk4j tupelo umich-botnets
[+++] Updating repos took 00:00:00
```

Looking in `~/dims/` for `.mrconfig`:

```
[dimsenv] mbogges@b52:~ () $ cd dims
[dimsenv] mbogges@b52:~/dims () $ ls -a
.  ..  git
```

- Create .mrconfig

```
[dimsenv] mbogges@b52:~/dims () $ touch .mrconfig
[dimsenv] mbogges@b52:~/dims () $ ls -a
.  ..  git  .mrconfig
```

- Run dims.git.syncrepos

```
[dimsenv] mbogges@b52:~/dims () $ cd ..
[dimsenv] mbogges@b52:~ () $ dims.git.syncrepos
[+++] Found 49 available repos at git@git.devops.develop
[+++] Adding Repo[1] ansible-inventory to /home/mbogges/dims/.mrconfig and
↳checking it out.
mr checkout: /home/mbogges/dims/git/ansible-inventory
Cloning into 'ansible-inventory'...
remote: Counting objects: 481, done.
remote: Compressing objects: 100% (387/387), done.
remote: Total 481 (delta 237), reused 122 (delta 65)
Receiving objects: 100% (481/481), 62.36 KiB | 0 bytes/s, done.
Resolving deltas: 100% (237/237), done.
Checking connectivity... done.
Using default branch names.

Which branch should be used for tracking production releases?
- master
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []

mr checkout: finished (1 ok)

<snip>

[+++] Updated 49 of 49 available repos.
[+++] Summary of actions for repos that were updated:
- Any changes to branches at origin have been downloaded to your local repository
- Any branches that have been deleted at origin have also been deleted from your
↳local repository
- Any changes from origin/master have been merged into branch 'master'
- Any changes from origin/develop have been merged into branch 'develop'
- Any resolved merge conflicts have been pushed back to origin
[+++] Added 49 new repos: ansible-inventory ansible-playbooks cif-client cif-java
↳configs dims dims-ad dims-adminguide dims-asbuilt dims-ci-utils dims-dashboard
↳dims-db-recovery dims-devguide dims-dockerfiles dims-dsdd dims-jds dims-keys
↳dims-ocd dims-packer dims-parselogs dims-sample-data dims-sr dims-supervisor
↳dims-svd dimssysconfig dims-test-repo dims-tp dims-tr dims-vagrant ELK fuse4j
↳ipgrep java-native-loader java-stix-v1.1.1 mal4s MozDef ops-trust-openid ops-
↳trust-portal poster-deck-2014-noflow prisem prisem-replacement pygraph rwfind
↳sphinx-autobuild stix-java ticketing-redis tsk4j tupelo umich-botnets
```

```
[+++] Updating repos took 00:07:19
```

### 3.5.4 Build Python Virtual Environment on Remote Machine

- When logged in to remote machine, change directories to location of virtual environment build scripts:

```
[dimsenv] mboggess@b52:~ () $ cd $GIT/ansible-playbooks
```

- Run the DIMS command to build the system virtualenv for access to system DIMS commands:

```
[dimsenv] mboggess@b52:~/dims/git/ansible-playbooks (develop) $ ./dimsenv.install.  
↪system
```

- Run `exec bash` to refresh:

```
[dimsenv] mboggess@b52:~/dims/git/ansible-playbooks (develop) $ exec bash  
[+++] DIMS shell initialization [ansible-playbooks v1.2.107]  
[+++] Sourcing /opt/dims/etc/bashrc.dims.d/bashrc.dims.network ...  
[+++] OpenVPN status:  
* VPN '01_uwapl_daveb52' is running  
* VPN '02_prsm_dave-prisem-2' is running  
[+++] Sourcing /opt/dims/etc/bashrc.dims.d/bashrc.dims.virtualenv ...  
[+++] Activating virtual environment (/home/mboggess/dims/envs/dimsenv) [ansible-  
↪playbooks v1.2.107]  
[+++] (Create file /home/mboggess/.DIMS_NO_DIMSENV_ACTIVATE to disable)  
[+++] Virtual environment 'dimsenv' activated [ansible-playbooks v1.2.107]  
[+++] Installed /home/mboggess/dims/envs/dimsenv/bin/dimsenv.install.user  
[+++] Installed /home/mboggess/dims/envs/dimsenv/bin/dimsenv.install.system  
[+++] Sourcing /opt/dims/etc/bashrc.dims.d/git-prompt.sh ...  
[+++] Sourcing /opt/dims/etc/bashrc.dims.d/hub.bash_completion.sh ...
```

Line “Activating virtual environment” should have path to `dimsenv/` via `$HOME/dims`.

- Run DIMS command to build user virtualenv:

```
[dimsenv] mboggess@b52:~/dims/git/ansible-playbooks (develop) $ ./dimsenv.install.  
↪user
```

- Run `exec bash` to refresh again.
- Check `$HOME/dims/envs/` for `dimsenv/` and activation scripts:

```
[dimsenv] mboggess@b52:~/dims/git/ansible-playbooks (develop) $ ls $HOME/dims/envs  
dimsenv          initialize      postdeactivate  postmkvirtualenv  preactivate      ↵  
↪premkproject    prermvirtualenv  
get_env_details  postactivate   postmkproject   postrmvirtualenv  predeactivate    ↵  
↪premkvirtualenv
```

### 3.5.5 Transfer Config Files

- Your account personalization files need to be transferred to the remote machine as well, including `.gitconfig`, `.vimrc`, and `.bash_aliases`.

From the local machine:

```
[dimsenv] mboggess@dimsdev2:~ () $ scp .bash_aliases mboggess@b52.tacoma.uw.edu:/
↪home/mboggess/
.bash_aliases                                100% 510      0.5KB/s   00:00
[dimsenv] mboggess@dimsdev2:~ () $ scp .gitconfig mboggess@b52.tacoma.uw.edu:/
↪home/mboggess/
.gitconfig                                100% 847      0.8KB/s   00:00
[dimsenv] mboggess@dimsdev2:~ () $ scp .vimrc mboggess@b52.tacoma.uw.edu:/home/
↪mboggess/
.vimrc                                    100% 314      0.3KB/s   00:00
```

On the remote machine, check for files and refresh bash:

```
[dimsenv] mboggess@b52:~ () $ ls -a
.      .ansible      .bash_history .bashrc  dims      .gitconfig .profile_
↪      .ssh          .vimrc
..     .bash_aliases .bash_logout .cache   examples.desktop .mrtrust   .python-
↪eggs  .viminfo
[dimsenv] mboggess@b52:~ () $ exec bash
```

## 3.6 JIRA Onboarding

### 3.6.1 Adding LDAP Entries for Users

We have an OpenLDAP server which serves as an authorization backend for our LemonLDAP SSO. Authentication is provided by OpenID Connect. It also serves as the user directory for JIRA.

**Note:** You will need an application to be able to edit/add directory information. [Apache Directory Studio](#) is cross platform and recommended. Ideally, the Trident portal would directly feed these records, rather than requiring someone follow the lengthy steps outlined below using a more laborious graphical user interface.

An Ansible role `apache-directory-studio` is used to install this application. Once this role has been applied, you can start the GUI with the following command:

```
$ apache-directory-studio &
```

The first time the program is run, a connection must be configured for the project LDAP server. Follow the instructions in [Add New Connection to Apache Directory Studio](#) to create the initial connection.

#### Attention:

When starting Adobe Directory Studio from the command line, you *must* add the `&` to run the program in the background. Since this is not a terminal program that takes input at the command line, failing to background the process will result in the shell not returning to a command prompt until after you quit the application, which novice Linux users unfamiliar with command shells and background processes will interpret as the terminal window being “hung” or “frozen”.

After Adobe Directory Studio has been installed and configured, start the application. You should see the initial connection in the list:

1. Click on the connection in the **Connections** list. (If you followed the instructions in [Add New Connection to Apache Directory Studio](#), the connection you want is labelled `ldap.devops.develop`.)

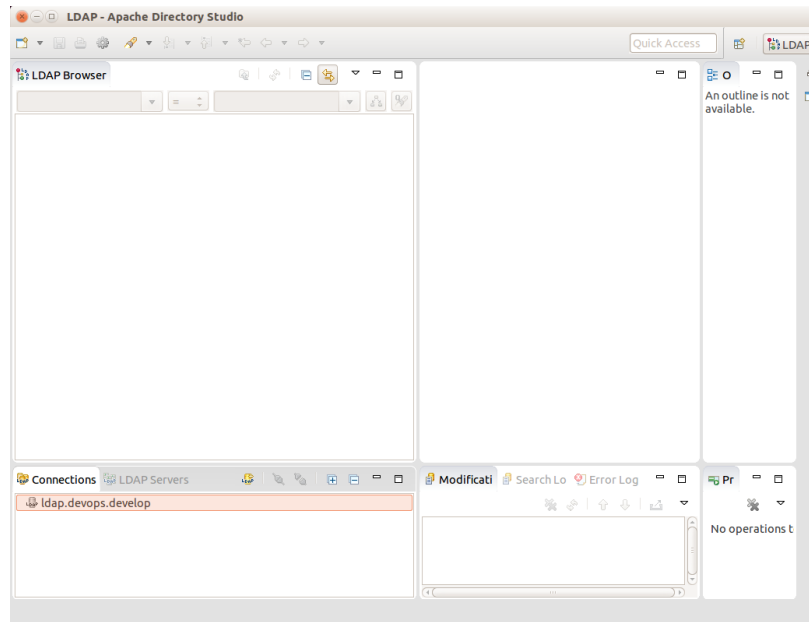


Fig. 3.1: Initial LDAP Browser Connection list

2. Click to open **DIT** in the tree.
3. Click to open **dc=prisem,dc=ashington,dc=edu** in the tree.
4. Click to open **ou=Users** in the tree. The current users will display.
5. Right-click **ou=Users** to open context menu and click **New -> New Entry**.
6. Select **Use existing entry as template**. Click **Browse** button to open the **ou** and select a member.
7. Click **Next**.
8. In the **Object Classes** dialog, do not add any more object classes. Just click **Next**.
9. In the **Distinguished Name** dialog, replace the template user's name you selected with the new user's name. The **DN** preview should then look like **cn=new\_user\_name,ou=Users,dc=prisem,dc=ashington,dc=edu**.
10. Click **Next**.
11. In the **Attribute Description** dialog (center panel), replace the template values with the values for your new user. Double click each **Valuefield** to edit.

---

**Note:** Tab to the next field or the value you entered might not be saved.

---

- **sn** - Enter the user's Last name
- **displayName** - Enter the user's First and Last name
- **mail** - Enter the user's Gmail address using for authenticating with OpenID Connect authentication.
- **ssoRoles** - These are used for testing right now (you can leave them as is.)
- **uid** - enter the uid in the form **firstname.lastname**
- **userPassword** - enter a password. It will be hashed.

12. Click **Finish**.

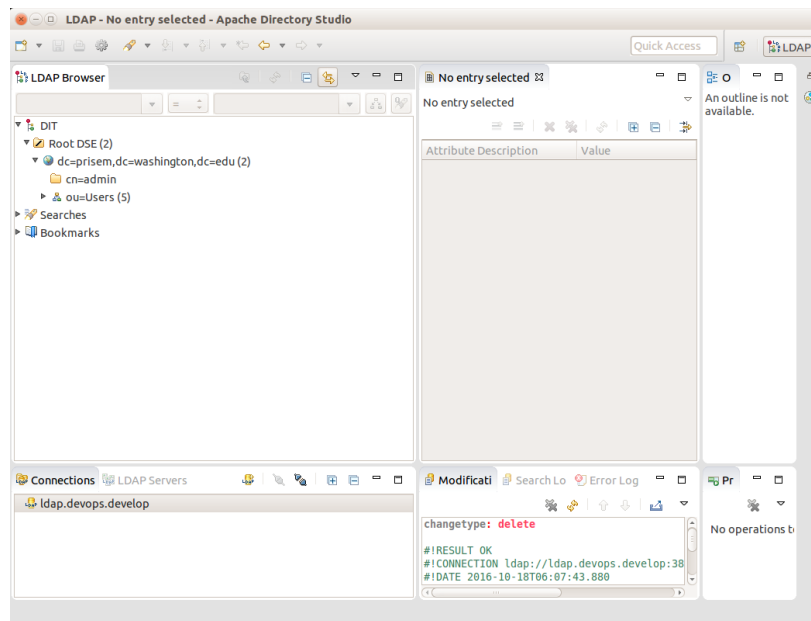


Fig. 3.2: DIT for connection ldap.devops.develop

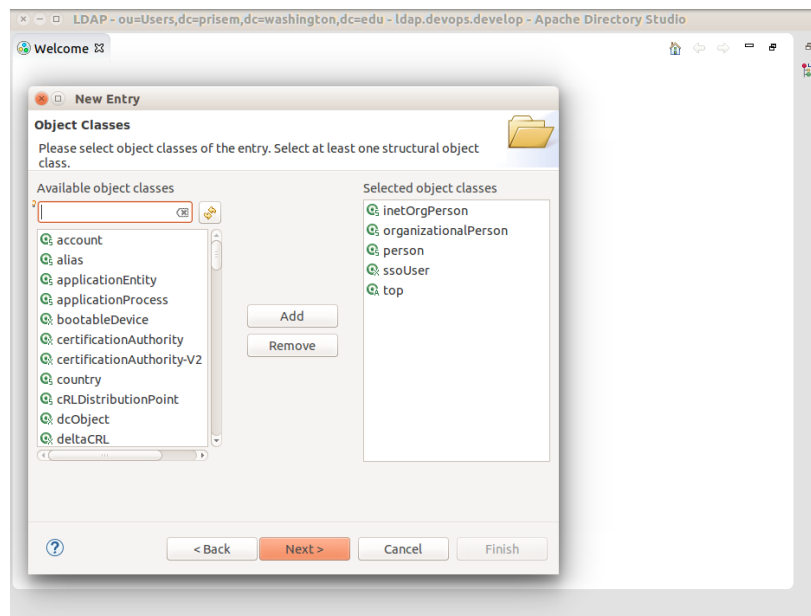


Fig. 3.3: Object Classes (skip)

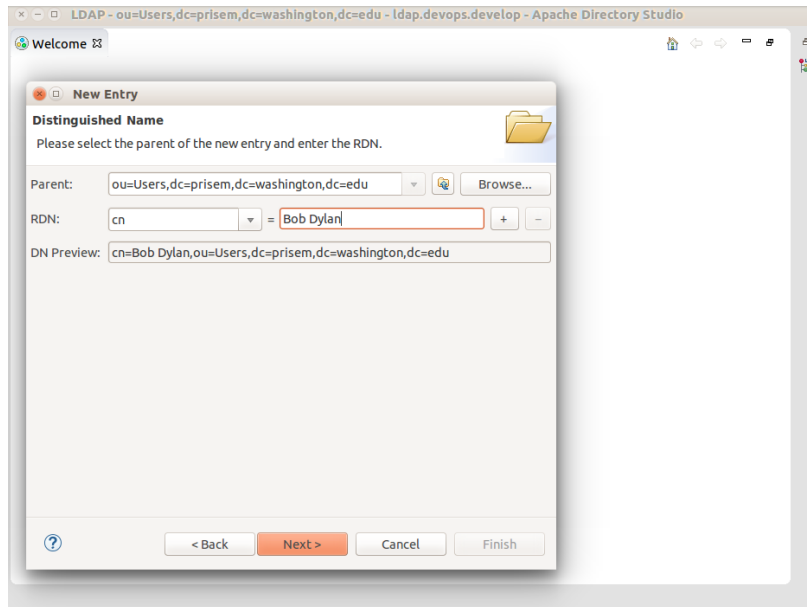


Fig. 3.4: Distinguished Name dialog

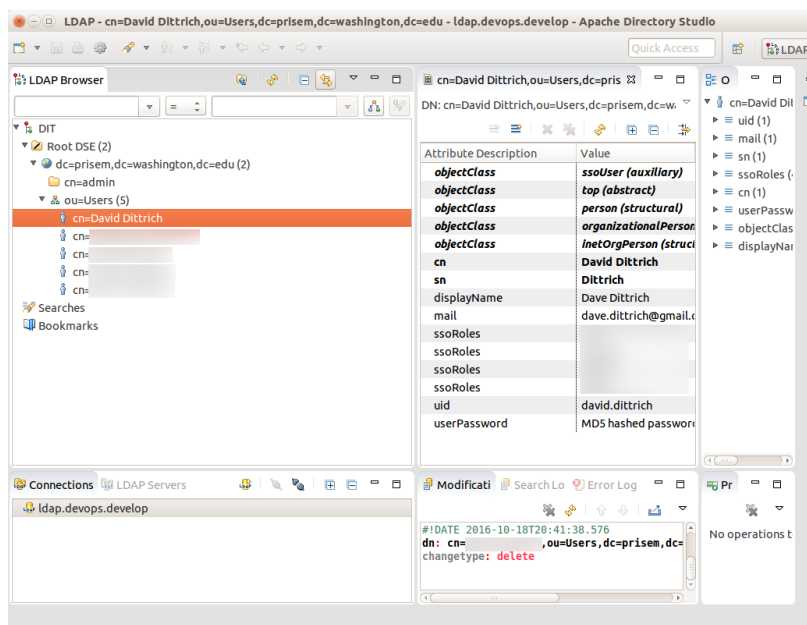


Fig. 3.5: Attribute Description dialog



13. Click on the new member and verify the fields. Edit any that were not entered correctly.

Exit the application when you are done and have the user test the authentication by going to <http://jira.prisem.washington.edu/> and select **Google** in the the **OpenID Login** dialog:

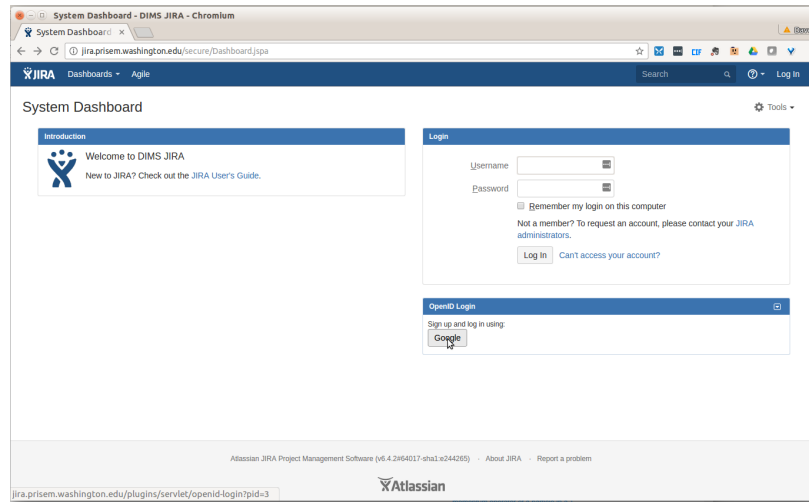


Fig. 3.6: JIRA Dashboard Login screen

**Note:** Google OpenID requires that the domain name of the system requesting authentication have a valid public DNS name. Even though you can connect to the system from within the VPN/VLAN via a non-public DNS name lookup, the authentication will not work. For this reason, the name `jira.prisem.washington.edu` is mapped in the split-horizon DNS mappings.

If the user has not recently authenticated to Google, they will be prompted for their password and/or second-factor authentication information. Once authenticated, the JIRA Dashboard will pop up.

### 3.6.2 Adding Users to JIRA Groups

After adding the user to JDAP, JIRA will show them as a valid user, but they will have no access once logged in.

To enable access to JIRA necessary to add and modify tickets, an administrator needs to grant access. Figure admin-panel shows the Administration panel where these changes will be made.

To grant a user “read-only” access, they need to be a member of the `jira-users` group. To grant “read/write” access, they need to also be a member of the `jira-developers` group. Only users with `jira-administrators` action can make these changes.

To change access, select **Groups** under the **Operations** column of the user table. The **Edit User Groups** dialog will pop up as shown in Figure adminpanel. Type into the search box to find options, then select the group from the list to add that group to the user’s permission.

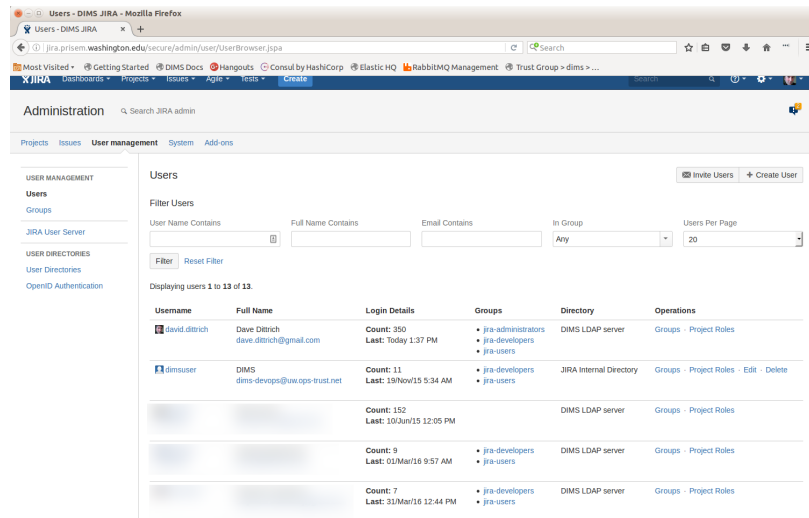


Fig. 3.7: JIRA Administration Panel

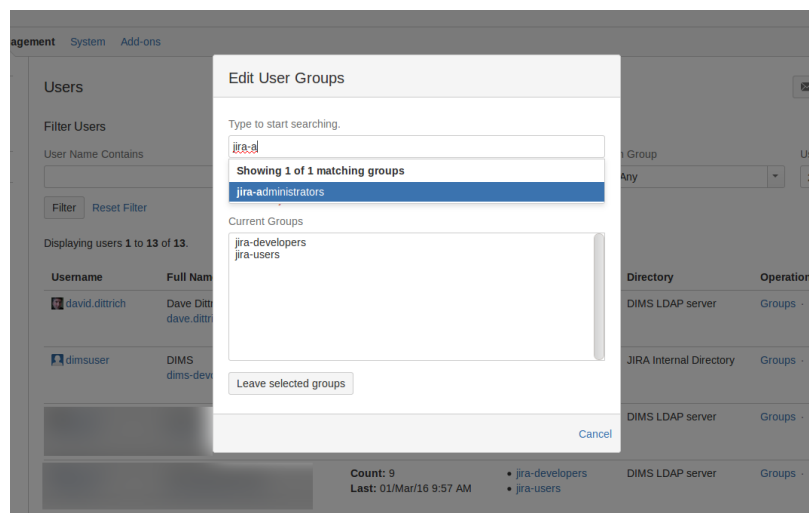


Fig. 3.8: JIRA Edit User Groups dialog

---

## Installation of DIMS Components on “Bare-metal”

---

This section describes installation of core Virtual Machine hypervisor servers, developer workstations, or collector devices on **physical hardware**. Installation of DIMS component systems in Virtual Machines is covered in Section *Installation of DIMS Components Using Virtual Machines*.

The initial operating system installation is handled using operating system installation media along with Kickstart auto-installation, followed by a second-stage pre-configuration step, and lastly by installation of required packages and configuration using Ansible.

A similar process is used to create Virtual Machines, though using Packer instead of stock OS installation ISO media plus Kickstart. This is covered in the `dimspacker:lifecycle` section of the `dimspacker:dimspacker` document.

### 4.1 Control and Target Prerequisites

For the **control machine**, the following must be true:

1. Must be able to run DIMS Ansible playbooks (i.e. be an existing developer workstation).
2. Must have the latest `dims-ci-utils` installed. That is, the latest `dims.remote.setupworkstation` script should be in `/opt/dims/bin`.
3. Must have the required DIMS VPN enabled (so it can retrieve DIMS Git repos and artifacts on Jenkins requested by playbooks.)

---

**Note:** We are assuming the control machine is an existing workstation that has been successfully used to run DIMS playbooks and has at a minimum followed the original instructions for setting environment variables and installing `dims-ci-utils`.

---

For the **target machine**, the following must be true:

1. The base operating system is installed.
2. An `ansible` account must be present, configured for `sudo` access for performing administrator tasks, with the matching public key allowing SSH access via the private key on the control machine.

3. Firewall rules must allow SSH access from the control machine.

## 4.2 Setting up a DIMS Developer Laptop

This section describes how to provision a new developer laptop using a custom bootable USB installation drive. Some of the steps are still manual ones, and these instructions will be updated as a more script-driven process is created. For now, this can serve to help guide the creation of the final process.

To achieve a repeatable and consistent process for installing a common base operating system (in this case, Ubuntu 14.04 LTS) that is ready to immediately be provisioned remotely from an Ansible control node, a customizable Ubuntu installation USB drive is used with all of the files necessary to go from a fresh computer system to a fully-functional networked host.

All of the steps for preparing an initial installation USB are given below, in the order they need to be performed. Once completed, you will have a bootable USB drive and a bit-copy of that drive that can be re-used.

---

**Note:** If you already have a bit-copy of one of these installation USB drives, skip to the *Cloning an installation USB* section.

If you already have a fresh (uncustomized) installation USB disk, skip forward to the *Customizing an installation USB* section.

---

---

**Note:** The DIMS project purchased a number of Dell Precision M4800 laptops for use for development and demonstration purposes. These laptops require the use of proprietary drivers for the Broadcom Wireless NIC and NVIDIA graphics controller. The specific models can be identified using `lspci`:

```
$ lspci -knn | grep -i Broadcom
03:00.0 Network controller [0280]: Broadcom Corporation BCM4352 802.11ac Wireless_
↳Network Adapter [14e4:43b1] (rev 03)
$ lspci | grep VGA
01:00.0 VGA compatible controller: NVIDIA Corporation GK107GLM [Quadro K1100M] (rev_
↳a1)
```

These drivers can be installed manually using the Ubuntu *Additional Drivers* app as seen in Figure *Additional Drivers from working laptop*.

There is prototype code in the Ubuntu post-install script designed to automate this task based on information from *How can I install Broadcom Wireless Adapter BCM4352 802.11ac PCID [14e4:43b1] (rev 03) on fresh install of Ubuntu 14.10 (Utopic Unicorn)?*, which is essentially:

```
$ sudo apt-get update
$ sudo apt-get install bcmwl-kernel-source
$ sudo modprobe wl
```

---

### 4.2.1 Preparation of Ubuntu installation USB drive

This section describes the manual steps used to create a two-partition 8GB Ubuntu installation USB drive. The following section describes the use of the program `dims.install.createusb` to bit-image copy this drive, store it for shared use by DIMS team members, and use this image copy to clone the original USB drive and then populate it with custom information to be used when auto-installing Ubuntu 14.04 on a development laptop using this customized USB drive.

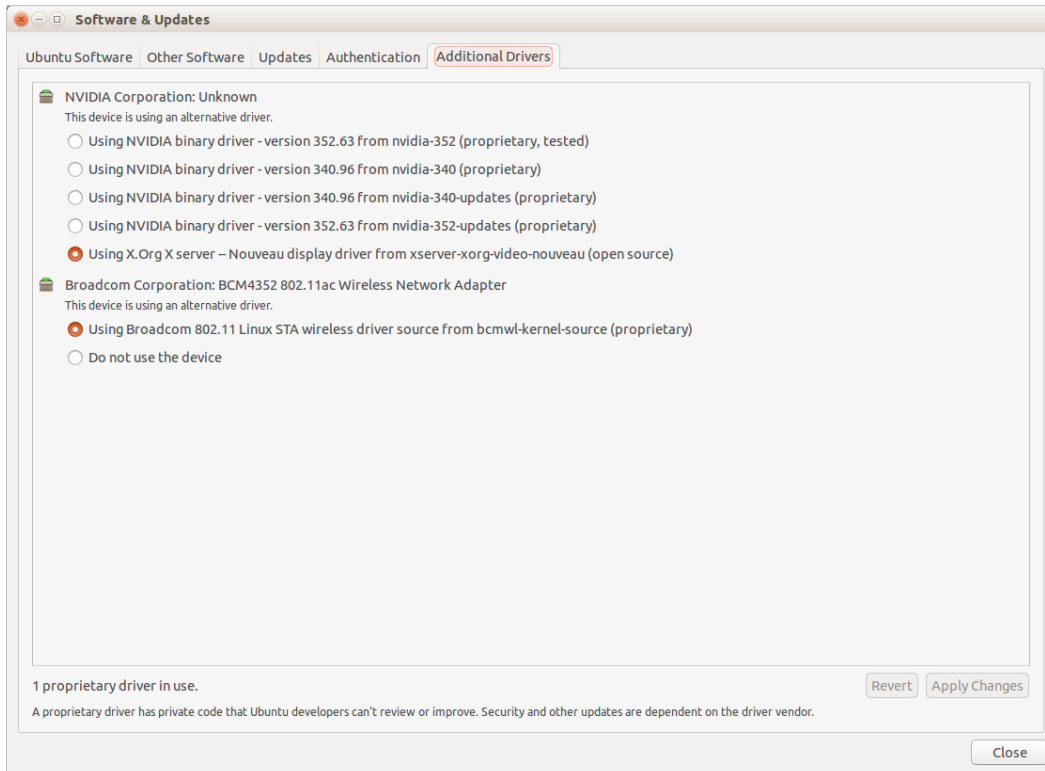


Fig. 4.1: Additional Drivers from working laptop

**Note:** Start out by studying the `--help` output of `dims.install.createusb` to understand the defaults it uses (shown by the highlighted lines in the following code block). These defaults are hard-coded into the program and should be updated when new Ubuntu install ISO images are used. Some of the command examples below make use of these defaults (rather than explicitly including all options on the command line.)

```
Usage: dims.install.createusb [options] [args]

Use "dims.install.createusb --help" to see help on command line options.

Options:
  -h, --help                show this help message and exit
  -d, --debug                Enable debugging.
  -D DEVICE, --device=DEVICE
                             Device file for mounting USB. [default: sdb]
  -H HOSTNAME, --hostname=HOSTNAME
                             Hostname of system to install. [default dimsdev3]
  -l USBLABEL, --usblabel=USBLABEL
                             USB device label. [default: DIMSINSTALL]
  --ubuntu-base=UBUNTUBASE
                             Ubuntu base version. [default: 14.04]
  --ubuntu-minor=UBUNTUMINOR
                             Ubuntu minor version. [default: 4]
  --base-configs-dir=BASE_CONFIGS_DIR
                             Base directory for configuration files. [default:
                             /opt/dims/nas/scd]
  -u, --usage                Print usage information.
  -v, --verbose              Be verbose (on stdout) about what is happening.
```

## Development Options:

Caution: use these options at your own risk.

```
--find-device      Attempt to find USB device actively mounted and exit.
--empty-casper     Empty out all contents (except lost+found) from
                   casper-rw and exit.
--ls-casper        Just list contents of casper-rw file system.
--label-casper     Put --usblabel into casper-rw and exit.
--mount-casper     Mount casper-rw in cwd and exit.
--umount-casper    Unmount casper-rw and exit.
--mount-usb        Mount DIMS install USB and exit. [default: sdb]
--unmount-usb      Unmount DIMS install USB and exit. [default: sdb]
--read-usb-into    Read USB drive into file. [default: False]
--write-usb-from   Write USB drive from file. [default: False]
-f IMAGEFILE, --imagefile=IMAGEFILE
                   File name to use for storing compressed USB image.
                   [default: ubuntu-14.04.4-install.dd.bz2]
--block-size=BLOCK_SIZE
                   Block size to use for 'dd' read/write. [default: 512]
```

---

## Partition USB drive

If you are starting out with a blank USB drive, you must first partition the drive and label it so it is recognizable by DIMS scripts. An easy program to use for this purpose on Ubuntu is the [Gnome Partition Editor](#) (a.k.a., **GParted**).

Figure *GParted formatting and labeling* shows an 8GB USB drive partitioned using GParted. Create two partitions with the primary partition (shown here as `/dev/sdb1`) marked as **bootable**, with a FAT32 file system, and labeled DIMSINSTALL. Make the second partition an ext3 file system and label it DIMSBACKUP.

The partitions can also be shown using `fdisk -l` (here assuming the disk is mounted as `/dev/sdb`).

```
[dittrich@dimsdev2 git]$ sudo fdisk -l /dev/sdb

Disk /dev/sdb: 8009 MB, 8009023488 bytes
247 heads, 62 sectors/track, 1021 cylinders, total 15642624 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cc03e

   Device Boot      Start         End      Blocks    Id  System
/dev/sdb1   *        2048      4196351      2097152    b   W95 FAT32
/dev/sdb2             4196352     15640575      5722112    83   Linux
```

---

**Note:** The `dims.install.createusb` script looks for a partition with the label DIMSINSTALL and will not manipulate drives that do not contain a partition with this label.

---

---

**Note:** The second partition can be used for backing up a user's directory contents prior to re-installation of the operating system on a system. Since the kickstart process automatically partitions the hard drive, existing contents would be lost.

---

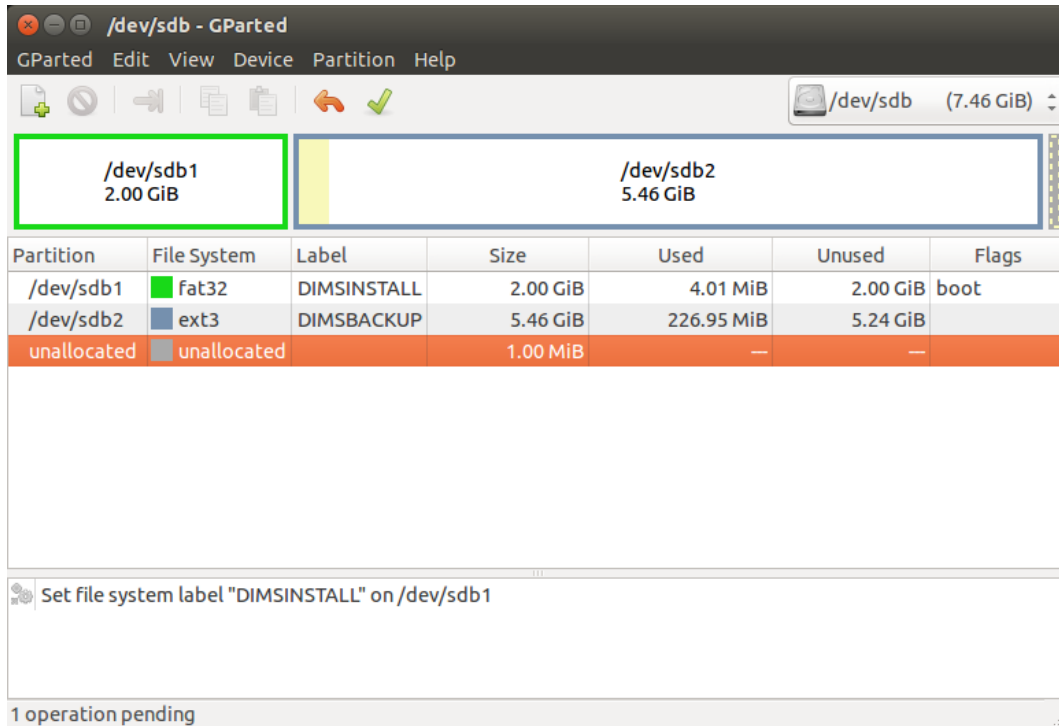


Fig. 4.2: GParted formatting and labeling

## Create Ubuntu installation USB

Installation of Ubuntu on a developer system is performed using the *Server* installation image (e.g., `ubuntu-14.04.4-server-amd64.iso`).

The program to use for this purpose is the Ubuntu **Startup Disk Creator**. Run it with `root` privileges (as they are needed to write the Master Boot Record on the USB drive).

```
$ sudo usb-creator-gtk &
```

After downloading the Ubuntu Server installation ISO and verifying its integrity using the signed SHA256 hash files, write the installation ISO to the partitioned USB.

The primary partition (i.e., `/dev/sdb1`) is where the Ubuntu installation ISO image (and `casper-rw` file system storage file, where DIMS customization files will be stored) will be written. Make sure that the option is checked to store files across boots, which will create a `casper-rw` partition image within the startup disk image.

**Note:** The second partition does not show up because it is not marked as bootable, though it may be mounted and visible using the File viewer.

Figure *Ubuntu Make Startup Disk* shows what the *Ubuntu Startup Disk Creator* GTK application will look like at this step.

**Note:** If you have to re-create the `DIMSINSTALL` partition with the Startup Disk Creator, it will erase the entire partition (which removes the label). To manually change the label, use GNU's GParted Partition Editor as described in the [Ubuntu RenameUSBDrive](#) page.

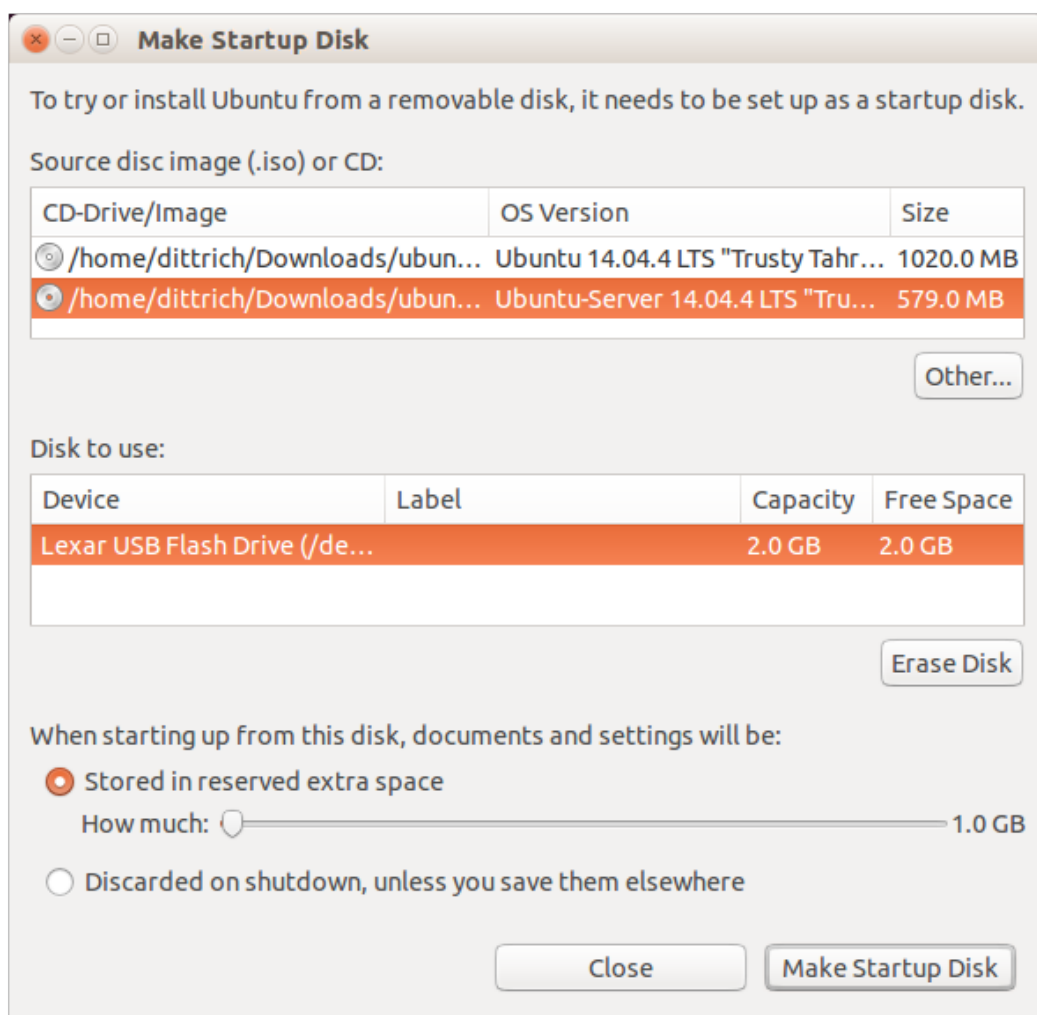


Fig. 4.3: Ubuntu Make Startup Disk



First verify the device name (so you don't accidentally harm another auto-mounted device), then use `mlabel` as seen here:

```
$ mount | grep '^/dev/sd'
/dev/sda1 on /boot type ext3 (rw)
/dev/sdb1 on /media/dittrich/917D-FA28 type vfat (rw,nosuid,nodev,uid=1004,gid=1004,
→shortname=mixed,dmask=0077,utf8=1,showexec,flush,uhelper=udisks2)
/dev/sdb2 on /media/dittrich/DIMSBACKUP type ext3 (rw,nosuid,nodev,uhelper=udisks2)
$ sudo mlabel -i /dev/sdb1 ::DIMSINSTALL
```

Now unmount and re-mount the device, and verify that the label did in fact get changed.

```
$ dims.install.createusb --unmount-usb
$ dims.install.createusb --mount-usb
$ mount | grep '^/dev/sd'
/dev/sda1 on /boot type ext3 (rw)
/dev/sdb1 on /media/dittrich/DIMSINSTALL type vfat (rw,nosuid,nodev,uid=1004,
→gid=1004,shortname=mixed,dmask=0077,utf8=1,showexec,flush,uhelper=udisks2)
/dev/sdb2 on /media/dittrich/DIMSBACKUP type ext3 (rw,nosuid,nodev,uhelper=udisks2)
```

## Bit-copy installation USB for cloning

After creating a bootable Ubuntu installation USB (which has not yet been customized for a specific host installation), a copy of the boot disk should be made. This allows for the vanilla installation USB to be cloned to as many USB drives as are needed, each then being uniquely customized. This customization includes host name, SSH keys, SSH `authorized_keys` and `known_hosts` files, OpenVPN certificates, and any other files used in the installation and setup process necessary to result in a remotely Ansible configurable host.

```
$ dims.install.createusb --verbose --read-usb-into
[+++] dims.install.createusb
[+++] Reading USB drive on sdb into ubuntu-14.04.4-install.dd.bz2
15642624+0 records in
15642624+0 records out
8009023488 bytes (8.0 GB) copied, 1171.45 s, 6.8 MB/s
2498225+1 records in
2498225+1 records out
1279091271 bytes (1.3 GB) copied, 1171.51 s, 1.1 MB/s
[+++] Finished writing ubuntu-14.04.4-install.dd.bz2 in 0:19:31.506338 seconds
$ ls -l *.bz2
-rw-r--r-- 1 dittrich dittrich 837948365 Jan 18 18:57 ubuntu-14.04.2-install.dd.bz2
-rw-rw-r-- 1 dittrich dittrich 1279091271 Mar 25 21:49 ubuntu-14.04.4-install.dd.bz2
```

## 4.2.2 Cloning an installation USB

The previous section walked through the process of creating a skeleton Ubuntu auto-installation USB drive and bit-copying it to a compressed image file. This section describes how to take that compressed bit-copy and clone it to USB drives that are then customized for installing Ubuntu on specific bare-metal hosts for subsequent Ansible configuration.

We will assume that the previous steps were followed, producing a clone of the Ubuntu 14.04.4 install ISO in a file named `ubuntu-14.04.4-install.dd.bz2`, and that the USB drive we will be cloning to is available as `/dev/sdb`.

**Caution:** Be sure that you confirm this is correct, since this script does direct writes using `dd`, which can destroy the file system if applied to the wrong drive! There was not enough time to make this script more robust against use by someone who is unfamiliar with bit copy operations in Unix/Linux.

```
$ dims.install.createusb --write-usb-from --verbose
[+++] dims.install.createusb
[+++] Partition /dev/sdb12 is not mounted
[+++] Partition /dev/sdb11 is not mounted
[+++] Writing ubuntu-14.04.4-install.dd.bz2 to USB drive on sdb
dd: error writing '/dev/sdb': No space left on device
15632385+0 records in
15632384+0 records out
8003780608 bytes (8.0 GB) copied, 2511.1 s, 3.2 MB/s

bzip2: I/O or other error, bailing out. Possible reason follows.
bzip2: Broken pipe
      Input file = ubuntu-14.04.4-install.dd.bz2, output file = (stdout)
[+++] Wrote sdb to USB drive on ubuntu-14.04.4-install.dd.bz2 in 0:41:51.110440
↪seconds
```

**Note:** The `dd` error “No space left on device” and the `bzip2` error “Broken pipe” are normal. This happens because the exact number of blocks read from the disk in the copy operation precisely matches the number of blocks coming from the compressed file, which triggers a “disk full” condition. A direct read/write operation on the device, rather than shelling out to `dd`, would be more robust (but would also consume more time in coding that was not available.)

### 4.2.3 Customizing an installation USB

The installation ISO is customized with SSH keys, OpenVPN certificates, etc., by inserting files from a common file share into the installation USB.

**Danger:** These files that are inserted into the USB are **not** encrypted, and **neither are** the installation USB’s file systems. This requires physical control of the USB disk. These files should either be encrypted with something like Ansible Vault, or the file system encrypted such that it is decrypted as part of the Ubuntu install process.

In order to make the necessary files available to any of the DIMS developers, an NFS file share is used. Alternatives remote file sharing protocols include SSHFS and SMB.

An environment variable `CFG` points to the path to the files used to customize the installation ISO. At present, these are in directories with the short name of the host to be installed (e.g., `dimsdev3`).

```
[dimsenv] dittrich@dimsdev3:/opt/dims/nas () $ echo $CFG
/opt/dims/nas/scd
[dimsenv] dittrich@dimsdev3:/opt/dims/nas () $ tree $CFG/dimsdev3
/opt/dims/nas/scd/dimsdev3
+- IP
+- openvpn-cert
|   +- 01_uwapl_dimsdev3.conf
|   +- 02_prsm_dimsdev3.conf
+- PRIVKEY
+- REMOTEUSER
+- ssh-host-keys
```

```
| +- key_fingerprints.txt
| +- known_hosts.add
| +- ssh_host_dsa_key
| +- ssh_host_dsa_key.pub
| +- ssh_host_ecdsa_key
| +- ssh_host_ecdsa_key.pub
| +- ssh_host_ed25519_key
| +- ssh_host_ed25519_key.pub
| +- ssh_host_rsa_key
| +- ssh_host_rsa_key.pub
+- ssh-user-keys
  +- ubuntu_install_rsa
  +- ubuntu_install_rsa.pub

3 directories, 17 files
```

**Note:** The OpenVPN certificates are created by hand. Two separate VPNs were originally used as hardware was split between two separate server rooms on two separate subnets, each with non-routable (RFC 1918) VLANs behind the VPNs. Hardware was moved into one data center and this will be reduced to one VPN as soon as VM consolidation and cabling changes can be made to use a single VLAN.

**Note:** The IP, PRIVKEY, and REMOTEUSER files hold the values used by some DIMS scripts for setting variables used for remotely provisioning the host using Ansible. We are migrating to using `group_vars` and/or `host_vars` files for holding these values so they can be shared by other scripts and used in Jinja templates.

New SSH host key sets can be generated using `keys.host.create`.

```
[dimsenv] dittrich@dimsdemo1:/opt/dims/nas () $ keys.host.create -d $CFG/dimsdev3/ssh-
↪host-keys/ -v -p dimsdev3
[+++] Storing files in /opt/dims/nas/scd/dimsdev3/ssh-host-keys/
[+++] Removing any previous keys and related files
[+++] Generating 1024 bit dimsdev3 ssh DSA key
[+++] Generating 2048 bit dimsdev3 ssh RSA key
[+++] Generating 521 bit dimsdev3 ssh ECDSA key
[+++] Generating 1024 bit dimsdev3 ssh ED25519 key
[+++] Key fingerprints
1024 70:0e:ee:8b:23:34:cf:34:aa:3b:a0:ca:fd:50:58:a9 'dimsdev3 ssh DSA host key' ↪
↪ (DSA)
2048 7f:89:da:e7:4d:92:fd:c1:3f:96:4f:05:f5:72:63:65 'dimsdev3 ssh RSA host key' ↪
↪ (RSA)
521 0a:af:c7:c4:a8:35:47:48:22:b3:7e:5b:bf:39:76:69 'dimsdev3 ssh ECDSA host key' ↪
↪ (ECDSA)
256 b2:dd:be:36:4d:03:a4:57:17:fb:a9:a9:97:e5:58:51 'dimsdev3 ssh ED25519 host key' ↪
↪ (ED25519)
[dimsenv] dittrich@dimsdemo1:/opt/dims/nas () $ ls -l $CFG/dimsdev3/ssh-host-keys
total 18
-rw-rw-r-- 1 nobody nogroup 362 Apr 4 11:24 key_fingerprints.txt
-rw-rw-r-- 1 nobody nogroup 1304 Apr 4 11:24 known_hosts.add
-rw----- 1 nobody nogroup 668 Apr 4 11:24 ssh_host_dsa_key
-rw-r--r-- 1 nobody nogroup 617 Apr 4 11:24 ssh_host_dsa_key.pub
-rw----- 1 nobody nogroup 361 Apr 4 11:24 ssh_host_ecdsa_key
-rw-r--r-- 1 nobody nogroup 283 Apr 4 11:24 ssh_host_ecdsa_key.pub
-rw----- 1 nobody nogroup 432 Apr 4 11:24 ssh_host_ed25519_key
-rw-r--r-- 1 nobody nogroup 113 Apr 4 11:24 ssh_host_ed25519_key.pub
```

```
-rw----- 1 nobody nogroup 1679 Apr  4 11:24 ssh_host_rsa_key
-rw-r--r-- 1 nobody nogroup  409 Apr  4 11:24 ssh_host_rsa_key.pub
```

**Note:** The equivalent script to generate SSH user keys has not yet been written, but an early helper Makefile is available to perform these steps in a consistent manner. The highest level of security is achieved by having unique SSH keys for each account, however this would significantly complicate use of Ansible, which is designed to control a large number of hosts in a single run. Each DIMS instance being controlled by Ansible will thus have a shared key for the Ansible account that, at most, is unique to a deployment and/or category.

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/dims-keys/ssh-pub (develop*) $ _
↪DIMSUSER=ansible make genkey
ssh-keygen -t rsa \
            -C "DIMS key for ansible" \
            -f dims_ansible_rsa
Generating public/private rsa key pair.
dims_ansible_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in dims_ansible_rsa.
Your public key has been saved in dims_ansible_rsa.pub.
The key fingerprint is:
06:52:35:82:93:73:8b:e8:0f:7a:15:f4:44:29:a2:b8 DIMS key for ansible
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      ++oo          |
|    . B.+ .         |
|   . -.O.           |
|  o. o.o.           |
| o   . S            |
| Eo . .             |
| . +                |
| . . .              |
| .                  |
+-----+
ssh-keygen -l \
            -f dims_ansible_rsa.pub > dims_ansible_rsa.sig
[dimsenv] dittrich@dimsdemo1:~/dims/git/dims-keys/ssh-pub (develop*) $ ls -lat | head
total 128
-rw-rw-r-- 1 dittrich dittrich  81 Nov 15 14:58 dims_ansible_rsa.sig
-rw----- 1 dittrich dittrich 1675 Nov 15 14:58 dims_ansible_rsa
-rw-rw-r-- 1 dittrich dittrich  402 Nov 15 14:58 dims_ansible_rsa.pub
. . .
[dimsenv] dittrich@dimsdemo1:~/dims/git/dims-keys/ssh-pub (develop*) $ mv dims_
↪ansible_rsa $CFG/zion/ssh-user-keys/
```

After all keys, certificates, etc., are installed in the new host's directory in \$CFG, you can write the contents to the installation USB disk partition.

```
[dimsenv] dittrich@dimsdemo1:/git/dims-ci-utils/usb-install (develop*) $ dims.install.
↪createusb --help
Usage: ./dims.install.createusb [options] [args]

Use "./dims.install.createusb --help" to see help on command line options.
```

```
Options:
-h, --help                show this help message and exit
-d, --debug               Enable debugging.
-D DEVICE, --device=DEVICE
                        Device file for mounting USB. [default: sdb]
-H HOSTNAME, --hostname=HOSTNAME
                        Hostname of system to install. [default: dimsdemo1]
-l USBLABEL, --usblabel=USBLABEL
                        USB device label. [default: DIMSINSTALL]
--distro-version=DISTROVERSION
                        Distribution version. [default: 14.04.5]
--base-configs-dir=BASE_CONFIGS_DIR
                        Base directory for configuration files. [default:
/opt/dims/nas/scd]
-u, --usage               Print usage information.
-v, --verbose             Be verbose (on stdout) about what is happening.
-V, --version             Print version and exit.

Development Options:
  Caution: use these options at your own risk.

--find-device             Attempt to find USB device actively mounted and exit.
--empty-casper            Empty out all contents (except lost+found) from
                        casper-rw and exit.
--ls-casper               Just list contents of casper-rw file system.
--label-casper            Put --usblabel into casper-rw and exit.
--mount-casper            Mount casper-rw in cwd and exit.
--unmount-casper          Unmount casper-rw and exit.
--mount-usb               Mount DIMS install USB (sdb) and exit. [default:
False]
--unmount-usb             Unmount DIMS install USB (sdb) and exit. [default:
False]
--read-usb-into           Read USB drive into file. [default: False]
--write-usb-from          Write USB drive from file. [default: False]
-f IMAGEFILE, --imagefile=IMAGEFILE
                        File name to use for storing compressed USB image.
                        [default: ubuntu-14.04.5-install.dd.bz2]
--block-size=BLOCK_SIZE
                        Block size to use for 'dd' read/write. [default: 512]
[dimsenv] dittrich@dimsdemo1:/git/dims-ci-utils/usb-install (develop*) $ dims.install.
↪ createusb --hostname zion
```

After installing the operating system using the Kickstart customized USB drive, the system should be able to access the network. Test using `ping 8.8.8.8` to verify network connectivity and a default route.

Install an initial `clouds.yml` file to configure `dimsscli`:

```
[dimsenv] ansible@zion:~ () $ cat ~/.config/openstack/clouds.yml
clouds:
  ectf:
    profile: ectf
    prefer_ipv6: False
    force_ipv4: True
    consul_peers: ['node01.ops.ectf', 'node02.ops.ectf', 'node03.ops.ectf']
    region_name: ectf
    debug: True
```



---

## Installation of DIMS Components Using Virtual Machines

---

This section describes installation of servers, developer workstations, or collector devices using **virtual machines**. Installation of DIMS component systems on “bare-metal” is covered in Section *Installation of DIMS Components on “Bare-metal”*.

### 5.1 DIMS on Virtual Machines

A local deployment of the DIMS system installed on virtual machines includes the following systems:

- red.devops.local (Ubuntu Trusty)
- yellow.devops.local (Debian Jessie)
- blue16.devops.local (Ubuntu Xenial)
- core-01.devops.local (CoreOS 1164.1.0)
- core-02.devops.local (CoreOS 1164.1.0)
- core-03.devops.local (CoreOS 1164.1.0)

This list will be updated as the group changes.

The following services and configurations are currently installed on some or all of the machines:

- Basic DIMS configurations (environment variables, directories, etc)
- Basic DIMS utilities
- A DIMS-specific python virtual environment
- DNS
- Postfix
- Docker
- Consul

- Swarm
- Postgres
- Nginx
- Trident
- Vagrant
- Pycharm
- Byobu

This list will be updated as more services and configurations are added.

## 5.2 Prerequisites for Instantiating Virtual Machines

You must have a centralized place to organize all the VMs. Scripts used in the build process depend on this place being rooted at `/vm`. To most easily structure this, and run into the least trouble with the build scripts, run the `Vagrant` role against the machine you will be instantiating the VMs on.

Once you've done that, you should end up with a structure that looks like the following:

```
[dimsenv] mboggess@dimsdev2:ims/nas/private/files/vagrants () $ tree -L 2 /vm
/vm
+- box
|   +- coreos
|   +- red
+- cache
|   +- apt
|   +- coreos_production_vagrant.box
|   +- debian-7.11.0-amd64-netinst.iso
|   +- debian-8.5.0-amd64-netinst.iso
|   +- sources
|   +- ubuntu-14.04.4-desktop-amd64.iso
|   +- ubuntu-14.04.4-server-amd64.iso
|   +- ubuntu-16.04.1-server-amd64.iso
+- ovf
|   +- red
+- run
|   +- core-01
|   +- core-02
|   +- core-03
|   +- red
+- sources
+- vbox
```

As artifacts are made for the VMs (.box files, .ovf files, etc) they get placed into the appropriate folder. Some other files though you need to make sure you have before starting the build workflow. This includes any iso files for building the beefier Debian OSes or the CoreOS box files. We have gathered the isos on the \$NAS, so you need access to it in order to retrieve these files.

- Ubuntu 14.04.4 server iso download: `$NAS/share/isos/ubuntu-14.04.4-server-amd64.iso`
- Ubuntu 14.04.4 desktop iso download: `$NAS/share/isos/ubuntu-14.04.4-desktop-amd64.iso`
- Ubuntu 16.04.1 server iso download: `$NAS/share/isos/ubuntu-16.04.1-server-amd64.iso`
- Debian Jessie 8.6.0 iso download: `$NAS/share/isos/debian-8.5.0-amd64-netinst.iso`



- CoreOS 1164.1.0 box file download: \$NAS/share/boxes/coreos\_production\_vagrant.box

You can download most of these files from the web, but we did make some changes to the Ubuntu 16.04.1 server iso itself, so you really need the iso from the NAS.

Then you need to set up your `/vm/cache/sources` directory. Since this is for a local deployment, the `/vm/cache/sources` directory acts as the central artifacts server location.

These are the files you need:

```
[dimsenv] mboggess@dimsdev2:/vm/cache/sources () $ tree
.
+- dims-ci-utils-develop.tgz
+- prisem-rpc-0.5.10.tar.gz
+- Python-2.7.12.tgz
+- python-dimsccli-0.8.0.tar.gz
+- trident-cli_1.3.8_amd64.deb
+- trident-server_1.3.8_amd64.deb

0 directories, 11 files
```

To get these files you must download them from the artifacts server at `jenkins.devops.develop` in the `/data/src` directory. You can run `wget` or `curl` or `scp` to retrieve those files. Ensure they are stored at `/vm/cache/sources`.

Finally, you need access to the \$NAS so you have access to the SSH keys used to access the VMs. Just make sure the \$NAS is up before starting the process (run `dims.nas.mount`).

## 5.3 VM Build Workflow

Once all of the prerequisite structure and artifacts are in place, you can begin to build the VMs. You need to have access to the `dims-packer` and `ansible-playbooks` repos.

**Note:** Soon there should be a way to build these things using the `develop` branch on both of those repos. Currently, however, the major updates to the build workflow have been made on the `dims-packer` branch called `feature/dims-760`. Once that branch is merged, only specific feature updates will be on any branch; stable code for building the VMs will be available on the `develop` branch.

These instructions do *not* indicate branches as work *should* be done from the `develop` branch and *will* be able to be done from the `develop` branch soon.

Follow these steps to build the 3 CoreOS VMs and the 3 Debian VMs.

1. If you have the `byobu` program, get a new window (F2) and change directories to `$GIT/dims-packer`.
2. Make sure you have an updated repo (`git hf update && git hf pull`).
3. Build the artifacts for the VMs by running

```
for node in core-01 core-02 core-03 red yellow blue16;
do test.vagrant.factory build $node.devops.local;
done
```

This will build the CoreOS nodes first, which is nice because they build really fast, so you can move on to getting those machines booted and provisioned, while you're waiting for the beefier VM artifacts to build.

4. Once you've made it through the CoreOS VM builds, but are still waiting on red, yellow, and blue16, you can start to provision the CoreOS nodes. Get a new byobu window and split it into thirds, vertically (Ctrl-Shift-F2)
5. In each of the splits, you'll change directories to one of the CoreOS VM's run directories. So `cd /vm/run/core-01` in the left split, `cd /vm/run/core-02` in the middle split, `cd /vm/run/core-03` in the right split. You should have something that looks like this:

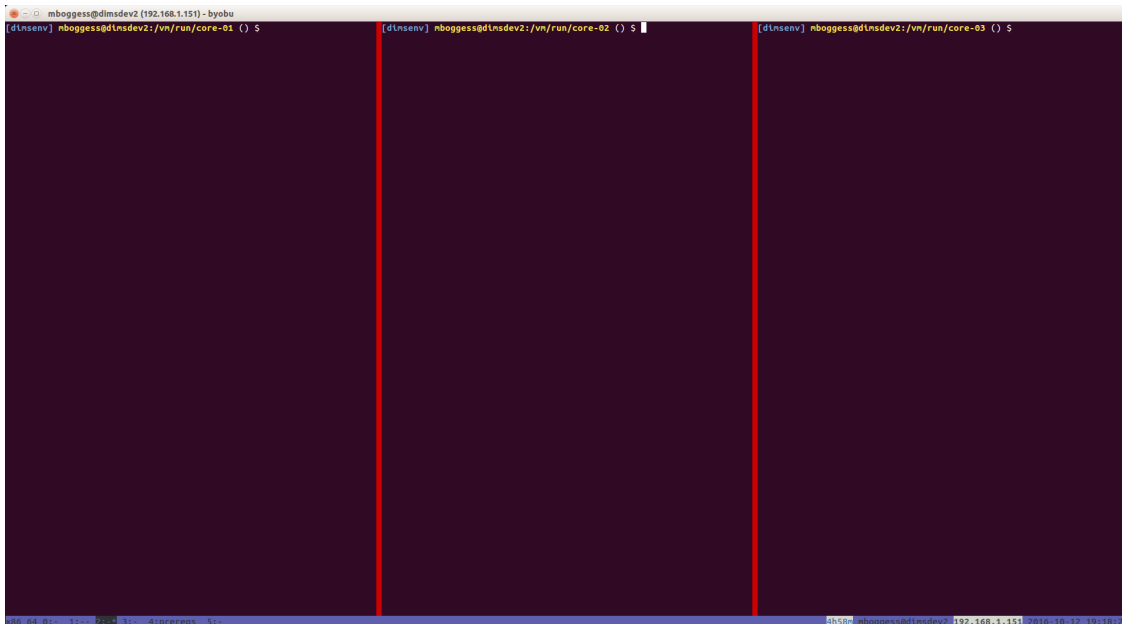


Fig. 5.1: Byobu window with 3 splits for working in CoreOS VM run directories

6. Now, you can use the byobu's "spray" functionality to send the same commands to all three splits. First, hit Alt-F9 to turn the spray functionality on. Then, we want to "boot" the machines and provision them, so we will run `make up && make provision`. This will run `vagrant up`, trigger some post-up configurations, and then use Ansible to provision the machines.

At the end, once everything has provisioned, you should get output from tests that are run. The more successes, the better. The current test output looks like the following:

7. When the red, yellow, and blue16 artifacts have all been built, you can do the same thing to boot and provision those machines. Get a new byobu window, make three vertical splits, and change directories to the appropriate run directories (`/vm/run/red`, `/vm/run/yellow`, `/vm/run/blue16`). You should have something that looks like the following

Turn on the byobu spray functionality and run `make up && make provision`.

Again, at the end, you should get output from the tests that are run. The very end of the current test output look like the following:

## 5.4 Run Directory Helper Makefile Targets

Beyond the steps outlined in the section above, there are many other `make` helpers in the VM run directory.

```
[dmsenv] mbogges@dmsdev2:/vm/run/red () $ make help
/vm/run/red
[Using Makefile.dims.global v1.7.1 rev ]
```

```

sys 0n5.0046
[+] New Virtualbox VMs:
[+] Output saved to make-provision-201610121751.txt
[+] Running 'test.runner --level system --exclude vpn --terse' on vagrant:
[+] Running test system/dns-accounts
1 test, 0 failures

[+] Running test system/dns
X [S][EV] /usr/sbin/dnsmasq exists and is executable
  (in test file system/dns-bats, line 12)
  [-X /usr/sbin/dnsmasq ] failed
X [S][EV] /etc/resolv.conf is a link to /run/resolvconf/resolv.conf
  (in test file system/dns-bats, line 28)
  result: "S[file /etc/resolv.conf]" failed with status 127
  (tmp/bats:340:src: line 20: File: command not found)
X [S][EV] dnsmasq is running
  (in test file system/dns-bats, line 27)
  [-Result: on 0 ] failed
X [S][EV] Can dig 127.0.0.1 www.google.com
  (from function: dig_returns_A_records' in file system/helpers.bash, line 94)
  (in test file system/dns-bats, line 66)
  dig: returns A_records: 127.0.0.1 www.google.com failed
X [S][EV] Short hostname resolves to 127.0.0.1 (127.0.0.1)
  (in test file system/dns-bats, line 83)
  result: "dig 127.0.0.1 (name) short" failed with status 0
X [S][EV] Short hostname resolves to 127.0.0.1 (no server specified)
  (in test file system/dns-bats, line 99)
  [-Result: on 127.0.0.1 ] failed
  [-dig core-01 -short ] returned
17 tests, 6 failures

[+] Running test system/proxy
1 test, 0 failures

[+] Running test system/deprecated
1 test, 0 failures

[+] Running test system/dns-base
X [S][EV] File bashrc.dns is referenced in /etc/bash.bashrc
  (in test file system/dns-base-bats, line 107)
  [-Exitcode: on 0 ] failed
36 tests, 1 failure

Connection to 127.0.0.1 closed.
make[1]: *** [provision] Error 2
make[1]: Leaving directory '/vm/run/core-01'
make: *** [reprovision-local] Error 2
[dmsenv] mbogges@dmsdev2:/vm/run/core-01 () $

[+] New Virtualbox VMs:
[+] Output saved to make-provision-201610121751.txt
[+] Running 'test.runner --level system --exclude vpn --terse' on vagrant:
[+] Running test system/dns-accounts
1 test, 0 failures

[+] Running test system/dns
X [S][EV] /usr/sbin/dnsmasq exists and is executable
  (in test file system/dns-bats, line 12)
  [-X /usr/sbin/dnsmasq ] failed
X [S][EV] /etc/resolv.conf is a link to /run/resolvconf/resolv.conf
  (in test file system/dns-bats, line 28)
  result: "S[file /etc/resolv.conf]" failed with status 127
  (tmp/bats:340:src: line 20: File: command not found)
X [S][EV] dnsmasq is running
  (in test file system/dns-bats, line 27)
  [-Result: on 0 ] failed
X [S][EV] Can dig 127.0.0.1 www.google.com
  (from function: dig_returns_A_records' in file system/helpers.bash, line 94)
  (in test file system/dns-bats, line 66)
  dig: returns A_records: 127.0.0.1 www.google.com failed
X [S][EV] Short hostname resolves to 127.0.0.1 (127.0.0.1)
  (in test file system/dns-bats, line 83)
  result: "dig 127.0.0.1 (name) short" failed with status 0
X [S][EV] Short hostname resolves to 127.0.0.1 (no server specified)
  (in test file system/dns-bats, line 99)
  [-Result: on 127.0.0.1 ] failed
  [-dig core-02 -short ] returned
17 tests, 6 failures

[+] Running test system/proxy
1 test, 0 failures

[+] Running test system/deprecated
1 test, 0 failures

[+] Running test system/dns-base
X [S][EV] File bashrc.dns is referenced in /etc/bash.bashrc
  (in test file system/dns-base-bats, line 107)
  [-Exitcode: on 0 ] failed
36 tests, 1 failure

Connection to 127.0.0.1 closed.
make[1]: *** [provision] Error 2
make[1]: Leaving directory '/vm/run/core-02'
make: *** [reprovision-local] Error 2
[dmsenv] mbogges@dmsdev2:/vm/run/core-02 () $

[+] New Virtualbox VMs:
[+] Output saved to make-provision-201610121751.txt
[+] Running 'test.runner --level system --exclude vpn --terse' on vagrant:
[+] Running test system/dns-accounts
1 test, 0 failures

[+] Running test system/dns
X [S][EV] /usr/sbin/dnsmasq exists and is executable
  (in test file system/dns-bats, line 12)
  [-X /usr/sbin/dnsmasq ] failed
X [S][EV] /etc/resolv.conf is a link to /run/resolvconf/resolv.conf
  (in test file system/dns-bats, line 28)
  result: "S[file /etc/resolv.conf]" failed with status 127
  (tmp/bats:340:src: line 20: File: command not found)
X [S][EV] dnsmasq is running
  (in test file system/dns-bats, line 27)
  [-Result: on 0 ] failed
X [S][EV] Can dig 127.0.0.1 www.google.com
  (from function: dig_returns_A_records' in file system/helpers.bash, line 94)
  (in test file system/dns-bats, line 66)
  dig: returns A_records: 127.0.0.1 www.google.com failed
X [S][EV] Short hostname resolves to 127.0.0.1 (127.0.0.1)
  (in test file system/dns-bats, line 83)
  result: "dig 127.0.0.1 (name) short" failed with status 0
X [S][EV] Short hostname resolves to 127.0.0.1 (no server specified)
  (in test file system/dns-bats, line 99)
  [-Result: on 127.0.0.1 ] failed
  [-dig core-03 -short ] returned
17 tests, 6 failures

[+] Running test system/proxy
1 test, 0 failures

[+] Running test system/deprecated
1 test, 0 failures

[+] Running test system/dns-base
X [S][EV] File bashrc.dns is referenced in /etc/bash.bashrc
  (in test file system/dns-base-bats, line 107)
  [-Exitcode: on 0 ] failed
36 tests, 1 failure

Connection to 127.0.0.1 closed.
make[1]: *** [provision] Error 2
make[1]: Leaving directory '/vm/run/core-03'
make: *** [reprovision-local] Error 2
[dmsenv] mbogges@dmsdev2:/vm/run/core-03 () $

```

Fig. 5.2: CoreOS VMs provisioned and test output

```

[dmsenv] mbogges@dmsdev2:/vm/run/red () $
make[1]: *** [provision] Error 2
make[1]: Leaving directory '/vm/run/core-01'
make: *** [reprovision-local] Error 2
[dmsenv] mbogges@dmsdev2:/vm/run/core-01 () $

[dmsenv] mbogges@dmsdev2:/vm/run/yellow () $
make[1]: *** [provision] Error 2
make[1]: Leaving directory '/vm/run/core-02'
make: *** [reprovision-local] Error 2
[dmsenv] mbogges@dmsdev2:/vm/run/core-02 () $

[dmsenv] mbogges@dmsdev2:/vm/run/blue16 () $
make[1]: *** [provision] Error 2
make[1]: Leaving directory '/vm/run/core-03'
make: *** [reprovision-local] Error 2
[dmsenv] mbogges@dmsdev2:/vm/run/core-03 () $

```

Fig. 5.3: Byobu window with 3 splits for working in non-CoreOS VM run directories

Fig. 5.4: Non-CoreOS VMs provisioned and test output

```
-----
Usage: make [something]
```

Where "something" is one of the targets listed in the sections below.

```
----- Targets from Makefile -----
```

```
show - show all variables used with this Makefile
```

```
NOTE: all of the following are done with timing and with
      output saved to a file named 'make-DATESTRING.txt'
```

```
up - Do 'vagrant up --no-provision'
```

```
reboot - Do 'vagrant halt && vagrant up --no-provision'
```

```
halt - halt vagrant cluster
```

```
update-box - update the CoreOS Vagrant box file
```

```
provision - Time and record 'vagrant provision'
```

```
reprovision-remote - Update ansible-playbooks from remote (w/current checked out,
↳branch)
```

```
reprovision-local - Reprovision host via locally rsync-ed ansible-playbooks
```

```
sync-playbooks - Update ansible-playbooks by rsync from current checked out working,
↳directory
```

```
rebuild - use test.vagrant.factory from packer repo to do 'destroy' and 'build' in,
↳one step
```

```
destroy - Do 'vagrant destroy'
```

```
clean - Remove unnecessary files
```

```
spotless - Remove all temporary files for this VM.
```

```
listvms - lists all configured virtual machines (using 'vboxmanage')
```

```
list - list all running VMs
```

```
vminfo - See some info about VMs
```

```
test - Run 'test.sh' with bash -x and redirect output to 'test.out'
```

```
This is a helper that can be run from the /vagrant
directory in the VM. Have it write output to a file
```

```
        that you follow with "tail -F" and you can observe
        results from the host
run-tests: Run test.runner for system level tests
           This will be like at the end of running
           the Ansible provisioner, but at will.
@echo
----- Targets from /opt/dims/etc/Makefile.dims.global -----

help - Show this help information (usually the default rule)

dimsdefaults - show default variables included from Makefile.dims.global
print-SOMETHING - prints the value of variable "SOMETHING"
version - show the Git revision for this repo
envcheck - perform checks of requirements for DIMS development

-----
```



---

# Installation of a Complete DIMS Instance

---

The Distributed Incident Management System (DIMS) is a system comprised of many sub-systems. That is to say, there are many inter-related and inter-dependent services that work together to provide a coherent whole which is called a *DIMS instance*. These subsystems may be provided by daemons running in a normal Linux system running on bare-metal (i.e., an operating system installed onto a standard computer hardware server), in a virtual machine running on a bare-metal host, or in Docker containers. Conceptually, it does not matter what underlying operating system is used, whether it is physical or virtual, or whether it is a Docker container: DIMS is comprised of micro-services that communicate using standard TCP/IP connections, regardless of where those services are running.

This chapter covers the steps necessary to install and configure a DIMS instance using (a) a single server running a cluster comprised of three virtual machines, and (b) a three-node bare-metal cluster.

## 6.1 Cluster Foundation Setup

To bootstrap a DIMS instance, it is necessary to first install the required base operating system, pre-requisite packages, and software components that serve as the foundation for running the DIMS micro-services. This includes the DIMS software and configuration files that differentiate one DIMS instance from another on the network.

Each DIMS instance has a routable Internet connection from at least one node and an internal local area network on which the DIMS system components are connected on the back end. This means there is at least one IP address block that is shared on the back, regardless of whether the primary node has its own DNS domain and Internet accessible IP address (as would be the case for a production service deployment) or uses dynamic addressing on WiFi or wired interface for a local development deployment.

A DIMS deployment that is to be used for public facing services on the Internet requires a real DNS domain and routable IP address(es), with SSL certificates to secure the web application front end. To remotely administer the system requires setting up SSH keys for secure remote access and/or remote administration using Ansible.

Accounts in the Trident user portal can be set up from the command line using the `tcli` user interface, or by using the Trident web application front end.

### **6.1.1 Single-host Virtual Machine Deployment**

## **6.2 Bootstrapping User Base**



This chapter introduces **Trident**, a “Trusted Information Exchange Toolkit” that facilitates the formation of trust groups, communication between members of trust groups, among other things. This chapter will walk through the installation and configuration of Trident and its prerequisites. How to use Trident and its various features will be covered in a different section.

## 7.1 Installing Trident manually

This section walks through the steps to use the `tcli` command line interface to manually configure a Trident deployment with an initial trust group, trust group administrator accounts and default mailing lists. These would be the steps necessary to bootstrap a Trident system for use by a trusted information sharing organization before starting to add regular trust group members and moving into the standard vetting process for growing the trust group.

Before logging in, you can get help on the top level command options using `tcli help`:

```
$ tcli help
-- Trident Help --

Welcome to the Trident menu system which is CLI command based.
If a given command is not in help menu the selected user does not have permissions_
↳for it.

Each section, items marked [SUB], has its own 'help' command.

The following commands are available on the root level:
user          [SUB]          User commands
system        [SUB]          System commands
```

Logging in is done using the `system` subcommand block. To get help on that subcommand block, add the subsection to the command:

```
$ tcli system help
Help for system
```

login	<username> <password> <twofactor>	Login
logout		Logout
whoami		Who Am I?
get	[SUB]	Get values from the system

The standard Trident administrator account is `trident`. Log in to it with the secret password configured at the time the Trident packages were installed and the initial `tsetup` command was used to bootstrap the database.

```
$ tcli system login trident THE_ACTUAL_SECRET_PASSWORD
Login successful
```

Now that you are logged in, further subcommand blocks become available. Use `help` (or just add the subcommand without any options, in some cases) to see what new options are available:

```
$ tcli system help
Help for system
login          <username> <password> <twofactor> Login
logout         Logout
whoami        Who Am I?
swapadmin     Swap from regular to sysadmin user
get           [SUB] Get values from the system
```

To perform system administration actions, you must use `swapadmin` to change the logged in user to be an administrator:

```
$ tcli system swapadmin
Now a SysAdmin user
```

Again, this opens up further options and/or subcommands. Look to see what those are:

```
$ tcli system help
Help for system
report         Report system statistics
login          <username> <password> <twofactor> Login
logout         Logout
whoami        Who Am I?
swapadmin     Swap from regular to sysadmin user
set           [SUB] Configure the system
get           [SUB] Get values from the system
```

To get the current setting of system attributes, use `tcli system get` followed by the attribute you want to get. Again, you can either add `help` to see the list, or just use the command `tcli system get` to see the attributes:

```
$ tcli system get help
Help for system get
name           System Name - Name of the System
welcome_text   Welcome Text - Welcome message shown on_
↪login page
adminname      Name of the Administrator(s) - Name of the_
↪Administrator, shown at bottom of the page
adminemail     Administrator email address - Email_
↪address of the Administrator, linked at the bottom of the page
copyyears      Copyright Years - Years that copyright_
↪ownership is claimed
email_domain   Email Domain - The domain where emails are_
↪sourced from
url_public     Public URL - The full URL where Trident is_
↪exposed to the public, used for redirects and OAuth2 (Example: https://trident.
↪example.net)
```

people_domain	People Domain - Domain used for people's
→email addresses and identifiers (Example: people.trident.example.net)	
cli_enabled	CLI Enabled - Enable the Web CLI (/cli/)
api_enabled	API Enabled - Enable the API URL (/api/)
→thus allowing external tools to access the details provided they have authenticated	
oauth_enabled	OAuth/OpenID Enabled - Enable OAuth 2.0
→and OpenID Connect support (/oauth2/ + /.wellknown/webfinger)	
no_index	No Web Indexing - Disallow Web crawlers/
→robots from indexing and following links	
email_sig	Email Signature - Signature appended to
→mailinglist messages	
require2fa	Require 2FA - Require Two Factor
→Authentication (2FA) for every Login	
pw_enforce	Enforce Rules - When enabled the rules
→below are enforced on new passwords	
pw_length	Minimal Password Length (suggested: 12)
pw_letters	Minimum amount of Letters
pw_uppers	Minimum amount of Uppercase characters
pw_lowers	Minimum amount of Lowercase characters
pw_numbers	Minimum amount of Numbers
pw_specials	Minimum amount of Special characters
sysadmin_restrict	IP Restrict SysAdmin - When provided the
→given CIDR prefixes, space separated, are the only ones that allow the SysAdmin bit	
→to be enabled. The SysAdmin b	
it is dropped for SysAdmins coming from different prefixes. Note that 127.0.0.1 and	
:::1 are always included in the set, thus CLI access remains working.	
header_image	Header Image - Image shown on the Welcome
→page	
logo_image	Logo Image - Logo shown in the menu bar
unknown_image	Unknown Person Image - Logo shown for
→users who do not have an image set	
showversion	Show Trident Version in UI - Show the
→Trident version in the UI, default enabled so that users can report issues to the	
→Trident Project	
adminemailpublic	Show Sysadmin E-mail to non-members - Show
→sysadmin e-mail in the public footer	
\$ tcli system get	
Help for system get	
name	System Name - Name of the System
welcome_text	Welcome Text - Welcome message shown on
→login page	
adminname	Name of the Administrator(s) - Name of the
→Administrator, shown at bottom of the page	
adminemail	Administrator email address - Email
→address of the Administrator, linked at the bottom of the page	
. . .	
showversion	Show Trident Version in UI - Show the
→Trident version in the UI, default enabled so that users can report issues to the	
→Trident Project	
adminemailpublic	Show Sysadmin E-mail to non-members - Show
→sysadmin e-mail in the public footer	

On first installation, the database exists for Trident configuration, but many attributes are not yet configured. For example, if you try to see the administrator's name and email address (which are shown in the main page of the web UI), do:

```
$ tcli system get adminname
unknown
$ tcli system get adminemail
unknown
```

There is a setting for the email domain, but it is just an example that will not actually work:

```
$ tcli system get email_domain
trident.example.net
```

You will need to set it to something that matches the SMTP Mail Transfer Agent (MTA), which is Postfix in this case:

```
$ tcli system set email_domain prisem.washington.edu
Updated email_domain
```

If you will be giving members a unique email address that is related to the trust group, rather than their personal or work email address, set the `people_domain` (which also initially comes with an example default):

```
$ tcli system get people_domain
people.trident.example.net
$ tcli system set people_domain people.prisem.washington.edu
Updated people_domain
```

As with the email addresses, the public URL is configured with a non-working example:

```
$ tcli system get url_public
https://trident.example.net
```

Set it to match the routable public URL that people will use to get to the Trident portal from the Internet:

```
$ tcli system set url_public https://zion.prisem.washington.edu
Updated url_public
```

You may toggle whether the web UI shows the address of the administrator to anyone who is not logged in (i.e., the general public) or does not. The default setting is `yes`:

```
$ tcli system get adminemailpublic
yes
```

There is no initial welcome text shown on the web UI. Set it as appropriate:

```
$ tcli system get welcome_text
Not Configured
$ tcli system set welcome_text "DIMS"
Updated welcome_text
```

Set the descriptive name of the administrator and the email address used to communicate with them:

```
$ tcli system set adminname "DIMS Administrator"
Updated adminname
$ tcli system set adminemail trident@prisem.washington.edu
Updated adminemail
```

You must set the name of the deployed portal that will be presented in the web UI:

```
$ tcli system get name
Not Configured
```

```
$ tcli system set name "DIMS Trident"
Updated name
```

A trailer is placed on all outgoing email messages. This allows including reminders about information sharing policies or other disclaimers. By default, it reads as follows:

```
$ tcli system get email_sig
All message content remains the property of the author
and must not be forwarded or redistributed without explicit permission.
```

The main web page includes a “header” graphic image that spans the browser window, allowing you to brand the portal. The file must be loaded under the `web_root` directory for the Trident web app to access it. By default, it is located in a subdirectory named `gfx/` with the name `gm.jpg`:

```
$ tcli system get header_image
/gfx/gm.jpg
$ sudo find / -type d -name gfx
/usr/share/trident/webroot/gfx
```

There is also a logo that is displayed by the web app:

```
$ tcli system get logo_image
/gfx/logo.png
```

You can either replace these files with content of your choosing, or you can add new files with different names and change the configuration settings. The directory with these files may contain other files, so check first:

```
$ ls /usr/share/trident/webroot/gfx
gm.jpg  info.png  invalid.png  logo.png  logo.svg  red_asterisk.png  search.png  ↵
↵unknown_person.jpg  valid.png  warning.png  xkcd_password_strength.png
```

If you wish to use your organization’s logo, you must first copy the file onto the system.

```
$ wget https://www.example.com/images/logo_24.png
--2017-01-13 12:41:27-- https://www.example.com/images/logo_24.png
Resolving www.example.com (www.example.com)... 93.184.216.34
Connecting to www.example.com (www.example.com)|93.184.216.34|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6220 (6.1K) [image/png]
Saving to: 'logo_24.png'
```

```
logo_24.png                               100
↵%[=====
↵]    6.07K  --.-KB/s    in 0s
```

```
2017-01-13 12:41:28 (125 MB/s) - 'logo_24.png' saved [6220/6220]
```

For this example, we will over-write the original logo with this new file:

```
$ sudo mv logo_24.png /usr/share/trident/webroot/gfx/logo.png
```

For the next example, we will add a new file for the header image, and change the variable to point to it:

```
$ sudo mv vagrant/our_org_header.png /usr/share/trident/webroot/gfx/
$ tcli system set header_image our_org_header.png
Updated header_image
```

```
$ ls -l /usr/share/trident/webroot/gfx/
total 580
-rwxr-xr-x 1 root root 83078 Sep 12 07:37 gm.jpg
-rwxr-xr-x 1 root root 580 Sep 12 07:37 info.png
-rwxr-xr-x 1 root root 424 Sep 12 07:37 invalid.png
-rw-r--r-- 1 ansible ansible 6220 Dec 9 2015 logo.png
-rwxr-xr-x 1 root root 2541 Sep 12 07:37 logo.svg
-rwxr-xr-x 1 root root 223 Sep 12 07:37 red_asterisk.png
-rwxr-xr-x 1 root root 3287 Sep 12 07:37 search.png
-rwxr-xr-x 1 root root 2994 Sep 12 07:37 unknown_person.jpg
-rw-r--r-- 1 root root 59250 Jan 13 12:53 usss-1.jpg
-rw-rw-r-- 1 ansible dims 309901 Jan 13 12:50 our_org_header.png
-rwxr-xr-x 1 root root 389 Sep 12 07:37 valid.png
-rwxr-xr-x 1 root root 616 Sep 12 07:37 warning.png
-rwxr-xr-x 1 root root 93029 Sep 12 07:37 xkcd_password_strength.png
```

```
$ sudo chown root:root /usr/share/trident/webroot/gfx/*
```

```
$ sudo chmod 755 /usr/share/trident/webroot/gfx/*
```

```
$ ls -l /usr/share/trident/webroot/gfx/
total 580
-rwxr-xr-x 1 root root 83078 Sep 12 07:37 gm.jpg
-rwxr-xr-x 1 root root 580 Sep 12 07:37 info.png
-rwxr-xr-x 1 root root 424 Sep 12 07:37 invalid.png
-rwxr-xr-x 1 root root 6220 Dec 9 2015 logo.png
-rwxr-xr-x 1 root root 2541 Sep 12 07:37 logo.svg
-rwxr-xr-x 1 root root 223 Sep 12 07:37 red_asterisk.png
-rwxr-xr-x 1 root root 3287 Sep 12 07:37 search.png
-rwxr-xr-x 1 root root 2994 Sep 12 07:37 unknown_person.jpg
-rwxr-xr-x 1 root root 309901 Jan 13 12:50 our_org_header.png
-rwxr-xr-x 1 root root 389 Sep 12 07:37 valid.png
-rwxr-xr-x 1 root root 616 Sep 12 07:37 warning.png
-rwxr-xr-x 1 root root 93029 Sep 12 07:37 xkcd_password_strength.png
```

```
$ tcli system get header_image
/gfx/gm.jpg
```

```
$ tcli system set header_image /gfx/gm.jpg
Updated header_image
```

## 7.2 Installing Trident with Ansible

### 7.2.1 Prerequisites

The following items are necessary before installing Trident via Ansible:

- Access to and knowledge of how to use Ansible roles foundational to provisioning DIMS systems. More information about these roles can be found at [tbd:tbd](#).
- Host(s) provisioned by Ansible roles foundational to DIMS systems. If using multiple hosts for a Trident instance, they must all be provisioned with these roles.

- Access to and knowledge of how to use Ansible roles specific to standing up a working Trident instance. More information about these roles can be found below, and information about how to provision a host with them can be found at tbd:tbd.
- Latest Trident package OR
- Access to the github.com Trident repo

## 7.2.2 Trident Artifact Build Process

**Note:** You must have access to the Trident github repo in order to build Debian packages. You must be able to clone the repo.

The following section outlines the steps needed to obtain/update the Trident source code and build a Debian package from it so that artifact is available for use by the Ansible role.

1. Prerequisite environment, per Trident documentation on their DEV-1.3 branch:
  - Debian Jessie
2. Prerequisite packages, per Trident documentation on their DEV-1.3 branch:
  - build-essential
  - git
  - pbuilder
3. Additional packages required, not listed in Trident's documentation:
  - dh-systemd
  - golang-go
4. Also, not listed in Trident's "build" requirements list, you must have Go installed. In Trident's "runtime" requirements list, it says version 1.5.1+, so I have downloaded and installed version 1.5.1.

```
$ cd /usr/local
$ wget https://storage.googleapis.com/golang/go1.5.1.linux-amd64.tar.gz
$ sudo tar -xzf go1.5.1.linux-amd64.tar.gz
$ export PATH=$PATH:/usr/local/go/bin
```

5. If you have a copy of the Trident source code, determine which version it is by running

```
$ /usr/sbin/tridentd --version
```

6. Compare this with the latest version of Trident source code on GitHub. This is a little tricky because there is a mismatch of version numbers between the `debian/changelog` file in the repo and the tags and branch names.

As of 13 Jul 2016, the official latest version is 1.2.0.

Go to the *Trident repo* on the master branch and go to the `debian/changelog` file. Here you will see the latest version.

7. Update or retrieve source code from GitHub. This may be a `git clone` or a `git pull` depending on how you are utilizing the Trident source (whether you need it once or if you are forking the repo).
8. In root directory of Trident git source, build the package:

```
$ dpkg-buildpackage -b -uc -us
```

This will build the binaries one level up from the trident root dir.

---

**Note:** The `dpkg-buildpackage` command will prompt you for your github username and password.

---

---

**Note:** The `dpkg-buildpackage` command runs a script called `doc/deps.sh` which has a plethora for “cannot find package X” errors. This is a known issue, see <https://github.com/bapril/trident/issues/371>. It still seems to build a usable artifact...

---

9. Place debian package wherever your Ansible role retrieves the package from for installation.

### 7.2.3 Provisioning Process

The following section outlines the steps needed to provision a host to stand up a working Trident instance.

1. Ensure all variables for your deployment are set to the correct values. In particular, ensure any Trident-Postgres-Nginx-Postfix networking variables are set correctly.
2. Apply the `postgres` Ansible role.
3. Apply the `nginx` Ansible role.
4. Apply the `postfix` Ansible role.
5. Apply the `trident` Ansible role.

Once all the roles have been applied, on the `nginx` host, you should be able to browse to the proxy address and see the Trident homepage. Instructions about how to actually use Trident and set up trust groups, etc. can be found at `tbd:tbd`.

## 7.3 Trident Prerequisites

The following are prerequisites that must be installed and configured before installing and configuring Trident:

- PostgreSQL 9.1+ database
- Postfix
- Nginx

### 7.3.1 PostgreSQL Database

The [Trident documentation](#) gives instructions on how to set up both a local postgres server and Trident database, as well as a remote server and database. In this section, we will cover and expand the instructions for installing and configuring a remote postgres server and Trident database. See Trident’s documentation page for a local installation and configuration.

For remote postgres servers, the [Trident documentation](#) recommends temporarily installing Trident on the remote target on which the postgres server will reside, use Trident’s `tsetup` command to create and setup the Trident database, then remove the Trident package.



**Note:** The “In a nutshell” steps in the “Remote Database” section of the [Trident documentation](#) seem to conflict with each other and the steps outlined in the “Local Database” section, which the location should really be the only thing that differentiates the two, I believe.

The following is my best interpretation, though it is just that, my interpretation. Notes and todo blocks follow at steps where I’m interpreting.

Essentially, the following steps would need to occur on the remote target:

1. Install PostgreSQL 9.1+
2. Create the `system trident` user
3. Temporarily install the Trident package(s).

**Note:** Here is a confusing bit from the “nutshell” steps in the “Remote Database” section of the [Trident documentation](#). The first two steps are to “Create the `trident` user” and “Create the `trident` database”, and the last step is “Run `tsetup` from the remote server as normal”. However, `tsetup` *does* those two things (user and database creation).

The third step says “Provide permissions for *the user* to access the database”. I’m not sure *which* user this means—the PostgreSQL `trident` user, I’m assuming. I’m also assuming that since `tsetup` creates a `trident` user for PostgreSQL, it will also give it the appropriate permissions. (I’m assuming this because the “Local Database” section said nothing about giving anyone appropriate permissions.)

Perhaps I’m confused, and this step means give the `system trident` user appropriate permissions, but...I don’t think the system user would be accessing the database.

Either way, for now, until this is clarified, I’m “skipping” this step because it seems to be taken care of by another “step”.

4. Properly configure the Trident daemon at `/etc/trident/trident.conf`

The following is a template of `trident.conf`:

```
#####
# Trident Configuration
#####
# Except for comment lines (anything starting with '#')
# this file is in the JSON format, thus mind the commas
# and quotes otherwise Trident can't properly use it.
#
# This file should only be readable by the Trident user
#####

{
  # Where the dbschemas, webroot and templates are located
  "file_root": "/usr/share/trident/",

  # Where variable files are stored
  "var_root": "/var/lib/trident/",

  # Crypto Keys for JWT (in directory relative to config dir)
  "jwt_key_prv": "jwt.prv",
  "jwt_key_pub": "jwt.pub",
}
```

```
#####
# PostgreSQL Database details
#####
# PSQL local unix socket
# Uses PSQL peer authentication
# This works out of the box on Debian
#####
"db_host": "/var/run/postgresql/",
#"db_port": "5432",
#"db_name": "trident",
#"db_user": "trident",
#"db_pass": "trident",

"db_port": "{{ tridentDBPort }}",
"db_name": "{{ tridentDBName }}",
"db_user": "{{ tridentDBUser }}",
"db_pass": "{{ tridentDBPass }}",

# The Nodename is used to identify this instance
# in a cluster of hosts. The name must be unique.
#
# The name is also used as a hostname for SMTP EHLO/HELO
# messages and thus must be a FQDN.
#
# empty => system configured (typically /etc/hostname)
"nodename": "{{ tridentFQDN }}",

# On which HTTP port to run our Trident Daemon
"http_port": "{{ tridentHTTPPort }}"
}
```

## 5. Properly configure the postgres pg\_hba.conf file (location variable)

The following is a template of pg\_hba.conf:

```
# PostgreSQL Client Authentication Configuration File
# =====
#
# Refer to the "Client Authentication" section in the PostgreSQL
# documentation for a complete description of this file. A short
# synopsis follows.
#
# This file controls: which hosts are allowed to connect, how clients
# are authenticated, which PostgreSQL user names they can use, which
# databases they can access. Records take one of these forms:
#
# local      DATABASE  USER  METHOD  [OPTIONS]
# host       DATABASE  USER  ADDRESS METHOD  [OPTIONS]
# hostssl    DATABASE  USER  ADDRESS METHOD  [OPTIONS]
# hostnossl  DATABASE  USER  ADDRESS METHOD  [OPTIONS]
#
# (The uppercase items must be replaced by actual values.)
#
# The first field is the connection type: "local" is a Unix-domain
# socket, "host" is either a plain or SSL-encrypted TCP/IP socket,
# "hostssl" is an SSL-encrypted TCP/IP socket, and "hostnossl" is a
# plain TCP/IP socket.
#
```

```
# DATABASE can be "all", "sameuser", "samerole", "replication", a
# database name, or a comma-separated list thereof. The "all"
# keyword does not match "replication". Access to replication
# must be enabled in a separate record (see example below).
#
# USER can be "all", a user name, a group name prefixed with "+", or a
# comma-separated list thereof. In both the DATABASE and USER fields
# you can also write a file name prefixed with "@" to include names
# from a separate file.
#
# ADDRESS specifies the set of hosts the record matches. It can be a
# host name, or it is made up of an IP address and a CIDR mask that is
# an integer (between 0 and 32 (IPv4) or 128 (IPv6) inclusive) that
# specifies the number of significant bits in the mask. A host name
# that starts with a dot (.) matches a suffix of the actual host name.
# Alternatively, you can write an IP address and netmask in separate
# columns to specify the set of hosts. Instead of a CIDR-address, you
# can write "samehost" to match any of the server's own IP addresses,
# or "samenet" to match any address in any subnet that the server is
# directly connected to.
#
# METHOD can be "trust", "reject", "md5", "password", "gss", "sspi",
# "krb5", "ident", "peer", "pam", "ldap", "radius" or "cert". Note that
# "password" sends passwords in clear text; "md5" is preferred since
# it sends encrypted passwords.
#
# OPTIONS are a set of options for the authentication in the format
# NAME=VALUE. The available options depend on the different
# authentication methods -- refer to the "Client Authentication"
# section in the documentation for a list of which options are
# available for which authentication methods.
#
# Database and user names containing spaces, commas, quotes and other
# special characters must be quoted. Quoting one of the keywords
# "all", "sameuser", "samerole" or "replication" makes the name lose
# its special character, and just match a database or username with
# that name.
#
# This file is read on server startup and when the postmaster receives
# a SIGHUP signal. If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect. You can
# use "pg_ctl reload" to do that.

# Put your actual configuration here
# -----
#
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.

# CAUTION: Configuring the system for local "trust" authentication
# allows any local user to connect as any PostgreSQL user, including
# the database superuser. If you do not trust all your local users,
# use another authentication method.

# TYPE    DATABASE          USER                ADDRESS             METHOD
```

```
# "local" is for Unix domain socket connections only
local    all                all                                trust
# IPv4 local connections:
host     all                all                                127.0.0.1/32        trust
# IPv6 local connections:
host     all                all                                ::1/128             trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local   replication        postgres               trust
#host    replication        postgres               127.0.0.1/32        trust
#host    replication        postgres               ::1/128             trust

# Allow connections to trident db from remote user via md5
host     {{ tridentDBName }} {{ tridentDBUser }}             0.0.0.0/0
→      md5
```

6. Ensure reachability of the database port defined in `/etc/trident/trident.conf`
7. Create the Trident database using the following command: `su - postgres -c "/usr/sbin/tsetup setup_db"`
8. Remove the Trident packages

### 7.3.2 Nginx Webserver

1. Install Nginx
2. Properly configure `/etc/nginx/conf.d/trident.conf`

The following is a template of the `nginx trident.conf` for a *production* system:

```
# The Trident Daemon Upstream
include /etc/trident/nginx/trident-upstream.inc;

# Redirect all HTTP (80) traffic to HTTPS (443)
# Trident should only be exposed over HTTPS
server {
    listen {{ nginxTridentHTTPPort }} default_server;
    listen [::]:{{ nginxTridentHTTPPort }} default_server;

    server_name _default_;

    rewrite ^ https://$host$request_uri permanent;
}

# The HTTPS server that exposed Trident
server {
    listen {{ nginxTridentHTTPSPort }} ssl;
    listen [::]:{{ nginxTridentHTTPSPort }} ssl;

    server_name {{ tridentFQDN }};

    # May need to variablize these...
    ssl_certificate trident.crt;
    ssl_certificate_key trident.key;
    ssl_prefer_server_ciphers on;
```

```

# And other SSL options, recommended:
# - ssl_dhparam
# - ssl_protocols
# - ssl_ciphers
# See https://cipherli.st/ for details

# STS header
add_header Strict-Transport-Security "max-age=31536001";

# HTTP Key Pinning
add_header Public-Key-Pins "Public-Key-Pins: max-age=5184000; pin-sha256=\"...\"
↪"

access_log /var/log/nginx/trident-access.log;

# Include the config for making Trident work
include /etc/trident/nginx/trident-server.inc;
}

```

The following is a template of the `nginx trident.conf` for a *development* system:

```

# The Trident Daemon Upstream
include /etc/trident/nginx/trident-upstream.inc;

# The HTTP server that exposed Trident - development only

server {
    listen {{ nginxTridentHTTPPort }} default_server;
    listen [::]:{{ nginxTridentHTTPPort }} default_server;

    server_name _default_;

    access_log /var/log/nginx/trident-access.log;

    # Include the config for making Trident work
    include /etc/trident/nginx/trident-server.inc;
}

```

**Note:** With this config, Nginx will only listen for the Trident daemon on an HTTP port (no HTTPS).

3. Properly configure Trident Daemon Upstream at `/etc/trident/nginx/trident-upstream.inc`

The following is a template of `trident-upstream.inc`:

```

upstream trident-daemon {
    server {{ tridentDBIP }}:{{ tridentDBPort }};
}

```

4. Properly configure the Trident server at `/etc/trident/nginx/trident-server.inc`

The following is an example of `trident-server.inc`:

```

# Our webroot (contains static, non-sensitive files, source if public ;)
root /usr/share/trident/webroot/;

```

```
#####
# Static files
#####
location /css/ {
}

location /favicon.ico {
}

location /gfx/ {
}

location /js/ {
}

#####
# Forward all requests to the Trident Daemon
#####
location / {
    client_max_body_size    0;
    proxy_set_header    Host $host;
    proxy_http_version    1.1;
    proxy_pass            http://trident-daemon;
}
```

### 7.3.3 Postfix

1. Install Postfix
2. Know the answers to the following:
  - What type of mail configuration
  - The Fully Qualified Domain Name (FQDN) of your server
3. Properly configure Postfix's main config file at `/etc/postfix/main.cf`

The following is a template of `main.cf`:

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version

# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTPE $mail_name (Ubuntu)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h
```

```

readme_directory = no

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_
↪unauth_destination
#myhostname = dimsdev2.prisem.washington.edu
myhostname = {{ postfixHostname }}
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
#mydestination = dimsdev2.prisem.washington.edu, localhost.prisem.washington.edu, ↪
↪, localhost
mydestination = {{ postfixDestinations }}
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

```

#### 4. Properly configure /etc/aliases

The following is a template of aliases:

```

# See man 5 aliases for format
postmaster:    root
{{ tridentHandlerName }}: "|/usr/sbin/trident-wrapper"

```

#### 5. Might have to configure Postfix's master config file at /etc/postfix/master.cf

The following is an example of master.cf:

```

#
# Postfix master process configuration file.  For details on the format
# of the file, see the master(5) manual page (command: "man 5 master" or
# on-line: http://www.postfix.org/master.5.html).
#
# Do not forget to execute "postfix reload" after editing this file.
#
# =====
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# =====
smtp            inet      n        -       -       -       -       smtpd
#smtp          inet      n        -       -       -       1       postscreen
#smtpd         pass      -        -       -       -       -       smtpd
#dnsblog       unix      -        -       -       -       0       dnsblog
#tlsproxy      unix      -        -       -       -       0       tlsproxy
#submission    inet      n        -       -       -       -       smtpd
#  -o syslog_name=postfix/submission

```

```

# -o smtpd_tls_security_level=encrypt
# -o smtpd_sasl_auth_enable=yes
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=
# -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
# -o milter_macro_daemon_name=ORIGINATING
#smtps      inet  n      -      -      -      -      smtpd
# -o syslog_name=postfix/smtps
# -o smtpd_tls_wrappermode=yes
# -o smtpd_sasl_auth_enable=yes
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=
# -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
# -o milter_macro_daemon_name=ORIGINATING
#628       inet  n      -      -      -      -      qmqpd
pickup     unix  n      -      -      60      1      pickup
cleanup    unix  n      -      -      -      0      cleanup
qmgr       unix  n      -      n      300     1      qmgr
#qmgr      unix  n      -      n      300     1      oqmgr
tlsmgr     unix  -      -      -      1000?   1      tlsmgr
rewrite    unix  -      -      -      -      -      trivial-rewrite
bounce     unix  -      -      -      -      0      bounce
defer      unix  -      -      -      -      0      bounce
trace      unix  -      -      -      -      0      bounce
verify     unix  -      -      -      -      1      verify
flush      unix  n      -      -      1000?   0      flush
proxymap   unix  -      -      n      -      -      proxymap
proxywrite unix  -      -      n      -      1      proxymap
smtp       unix  -      -      -      -      -      smtp
relay      unix  -      -      -      -      -      smtp
#
# -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq      unix  n      -      -      -      -      showq
error      unix  -      -      -      -      -      error
retry      unix  -      -      -      -      -      error
discard    unix  -      -      -      -      -      discard
local      unix  -      n      n      -      -      local
virtual    unix  -      n      n      -      -      virtual
lmtpl      unix  -      -      -      -      -      lmtpl
anvil      unix  -      -      -      -      1      anvil
scache     unix  -      -      -      -      1      scache
#
# =====
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
#
# Many of the following services use the Postfix pipe(8) delivery
# agent. See the pipe(8) man page for information about ${recipient}
# and other message envelope options.
# =====
#
# maildrop. See the Postfix MAILDROP_README file for details.
# Also specify in main.cf: maildrop_destination_recipient_limit=1

```



```
#
maildrop unix - n n - - pipe
 flags=DRhu user=vmail argv=/usr/bin/maildrop -d ${recipient}
#
# =====
#
# Recent Cyrus versions can use the existing "lmtp" master.cf entry.
#
# Specify in cyrus.conf:
#   lmtp    cmd="lmtpd -a" listen="localhost:lmtp" proto=tcp4
#
# Specify in main.cf one or more of the following:
#   mailbox_transport = lmtp:inet:localhost
#   virtual_transport = lmtp:inet:localhost
#
# =====
#
# Cyrus 2.1.5 (Amos Gouaux)
# Also specify in main.cf: cyrus_destination_recipient_limit=1
#
#cyrus      unix - n n - - pipe
#   user=cyrus argv=/cyrus/bin/deliver -e -r ${sender} -m ${extension} ${user}
#
# =====
#
# Old example of delivery via Cyrus.
#
#old-cyrus unix - n n - - pipe
#   flags=R user=cyrus argv=/cyrus/bin/deliver -e -m ${extension} ${user}
#
# =====
#
# See the Postfix UUCP_README file for configuration details.
#
uucp      unix - n n - - pipe
 flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
#
# Other external delivery methods.
#
ifmail    unix - n n - - pipe
 flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp     unix - n n - - pipe
 flags=Fq. user=bsmtp argv=/usr/lib/bsmtp/bsmtp -t$nexthop -f$sender $recipient
scalemail-backend unix - n n - 2 pipe
 flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store ${nexthop} $
→{user} ${extension}
mailman    unix - n n - - pipe
 flags=FR user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.py
 ${nexthop} ${user}
```

#### 6. Might have to configure additional email addresses at /etc/postfix/virtual

The following is a template of virtual:

```
mail-handler@example.net {{ tridentHandlerName }}
@example.net             {{ tridentHandlerName }}
```

**Note:** The [Trident documentation](#) gave the information used to configure the `/etc/aliases` file and the `/etc/postfix/virtual` file, but then just said “Of course do configure the rest of Postfix properly.” I don’t really know what that means, so that’s why I included the `master.cf` file, since that was included in the `/etc/postfix` dir. There are a couple other files there, `/etc/postfix/dynamicmaps.cf` and `/etc/postfix/postfix-files`, along with a `sasl/` dir and a couple scripts.

---

## 7.4 Install Trident

Now we can install the Trident server and the Trident CLI.

1. Retrieve the Trident debian packages from `source.prisem.washington.edu`

```
$ wget http://source.prisem.washington.edu:8442/trident-server_1.0.3_amd64.deb
$ wget http://source.prisem.washington.edu:8442/trident-cli_1.0.3_amd64.deb
```

---

**Note:** The version may change...the above commands need to be kept in sync.

---

2. Properly configure the Trident daemon at `/etc/trident/trident.conf`

This template can be seen in the [PostgreSQL Database](#) section.

3. Properly configure Trident daemon defaults at `/etc/default/trident`

The following is an example of `/etc/default/trident`:

```
# This is a configuration file for /etc/init.d/trident; it allows you to
# perform common modifications to the behavior of the Trident daemon
# startup without editing the init script (and thus getting prompted
# by dpkg on upgrades).

# Start Trident at startup ? (ignored by systemd)
TRIDENT_ENABLED=No

# The username as who to run Trident
DAEMON_USER=trident

# Extra options to pass to the Trident daemon
DAEMON_OPTS="-username trident -insecurecookies -disabletwofactor -debug -config /
↳etc/trident"
```

## 7.5 Running Trident

There are several ways of running the Trident daemon, but we have divided them into a “secure, non-debug” way and a “non-secure, debug” way.

- Insecure, debug:

```
DAEMON_USER=trident /usr/sbin/tridentd \
  -insecurecookies \
  -disabletwofactor \
  -debug \
```

```
-config /etc/trident/ \
-daemonize \
-syslog \
-verbosedb
```

- Secure, non-debug:

```
DAEMON_USER=trident /usr/sbin/tridentd \
-config /etc/trident/ \
-daemonize \
```

---

**Note:**

- The above code is from a start script used by the Dockerfile created by Linda Parsons (\$GIT/dims-dockerfiles/dockerfiles/trident/conf/start.sh). I just grabbed it to show how to run the daemon. We should probably always have syslog enabled...
  - There's a note in that start script that says using the daemonize flag doesn't appear to be daemonizing the Trident daemon. Should keep that in mind.
- 

## 7.6 Using tcli on the command line

The following output shows some of the commands available to tcli command line users, and how to log in as a sysadmin user to gain access to more commands.

```
[dimsenv] ansible@yellow:~ () $ tcli help
--- Trident Help ---

Welcome to the Trident menu system which is CLI command based.
If a given command is not in help menu the selected user does not have permissions_
↳for it.

Each section, items marked [SUB], has its own 'help' command.

The following commands are available on the root level:
user [SUB] User commands
system [SUB] System commands
[dimsenv] ansible@yellow:~ () $ tcli user help
Help for user
password [SUB] Password commands
[dimsenv] ansible@yellow:~ () $ tcli system help
Help for system
login <username> <password> <twofactor> Login
logout Logout
whoami Who Am I?
get [SUB] Get values from the system
[dimsenv] ansible@yellow:~ () $ tcli system login trident trident123
Login successful
[dimsenv] ansible@yellow:~ () $ tcli system whoami
Username: trident
Fullname:
[dimsenv] ansible@yellow:~ () $ tcli system swapadmin
Now a SysAdmin user
[dimsenv] ansible@yellow:~ () $ tcli system help
```

```

Help for system
report                                Report system statistics
login                                <username> <password> <twofactor> Login
logout                                Logout
whoami                                Who Am I?
swapadmin                            Swap from regular to sysadmin user
set                                  [SUB]          Configure the system
get                                  [SUB]          Get values from the system
[dimsenv] ansible@yellow:~ () $ tcli user help
Help for user
new                                  <username> <email>      Create a new user
nominate                            <username> <email> <bio_info> <affiliation> <descr> Nominate_
↪New User
set                                  [SUB]          Set properties of a user
get                                  [SUB]          Get properties of a user
list                                <match>          List all users
merge                                <into> <from>      Merge a user
delete                              <username>        Delete a new user
2fa                                  [SUB]          2FA Token Management
email                                [SUB]          Email commands
password                            [SUB]          Password commands
detail                              [SUB]          Manage Contact Details
language                            [SUB]          Manage Language Skills
[dimsenv] ansible@yellow:~ () $

```

There are certain things with which a DIMS system is automatically configured. These attributes are set via tasks in the Trident Ansible role:

```

---
# file: v2/roles/trident/tasks/main.yml

<snip>

- name: Ensure trident administrator is logged in
  shell: "tcli system login {{ trident.initial_sysadmin.name }} {{ trident.initial_
↪sysadmin.password }}"
  register: tcli_login
  no_log: true
  when: ansible_lsb.codename == "jessie"
  become: yes
  tags: [ trident ]

- name: Require successful login to trident
  fail: "Failed to log in via trident: {{ tcli_login.stdout }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout != "Login successful"
  tags: [ trident ]

- name: Ensure system configuration is present
  shell: "{{ item }}"
  with_items:
    - "tcli system swapadmin"
    - "tcli system set name '{{ trident.name }}'"
    - "tcli system set welcome_text '{{ trident.welcome_text }}'"
    - "tcli system set url_public {{ trident.url_public }}"
    - "tcli system set adminname '{{ trident.adminname }}'"
    - "tcli system set adminemail '{{ trident.adminemail }}'"
    - "tcli system set email_domain '{{ trident.email_domain }}'"

```

```

- "tcli system set people_domain '{{ trident.people_domain }}'"
- "tcli system set logo_image '{{ trident.logo_image }}'"
- "tcli system set header_image '{{ trident.header_image }}'"
when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
become: yes
tags: [ trident ]

<snip>

#EOF

```

Once the role is run against the host machine which is to run the Trident application, not only is Trident running, and you have access to the web application, but the web app shows that the customization has taken place.

Additionally, we bootstrap global initial admin accounts and a initial trust group with its mailing lists:

```

---

# file: v2/roles/trident/tasks/main.yml

<snip>

- name: Ensure trident administrator is logged in
  shell: "tcli system login {{ trident.initial_sysadmin.name }} {{ trident.initial_
↪sysadmin.password }}"
  register: tcli_login
  no_log: true
  when: ansible_lsb.codename == "jessie"
  become: yes
  tags: [ trident ]

- name: Require successful login to trident
  fail: "Failed to log in via trident: {{ tcli_login.stdout }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout != "Login successful"
  tags: [ trident ]

<snip>

- name: Ensure initial sysadmin user example email is not present
  shell: "tcli user email remove trident@trident.example.net"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Ensure initial sysadmin user email is present
  shell: "tcli user email add {{ trident.initial_sysadmin.name }} {{ trident.initial_
↪sysadmin.email }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Force initial sysadmin email address to be confirmed
  shell: "tcli user email confirm_force {{ trident.initial_sysadmin.name }} {{ _
↪trident.initial_sysadmin.email }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

```

```
- name: Ensure initial TG is present
  shell: "tcli tg add {{ trident.initial_tg.id }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Ensure initial TG description is present
  shell: "tcli tg set descr {{ trident.initial_tg.descr }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Ensure initial ML is present
  shell: "tcli ml new {{ trident.initial_tg.id }} {{ trident.initial_ml }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Ensure global admin accounts are present
  shell: "tcli user new {{ item.key }} {{ item.value.email }}"
  with_dict: "{{ trident_admins }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Ensure global admin accounts have passwords
  shell: "tcli user password set portal {{ item.key }} {{ tridentSysAdminPass }}"
  with_dict: "{{ trident_admins }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Force global admin emails to be confirmed
  shell: "tcli user email confirm_force {{ item.key }} {{ item.value.email }}"
  with_dict: "{{ trident_admins }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Ensure global admin users have global sysadmin rights
  shell: "tcli user set sysadmin {{ item.key }} true"
  with_dict: "{{ trident_admins }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Nominate global admin users to initial TG
  shell: "tcli tg member nominate {{ trident.initial_tg.id }} {{ item.key }}"
  with_dict: "{{ trident_admins }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

- name: Approve global admin users to initial TG
  shell: "tcli tg member approve {{ trident.initial_tg.id }} {{ item.key }}"
  with_dict: "{{ trident_admins }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
```

```

tags: [ trident ]

- name: Ensure global admin users have initial TG sysadmin rights
  shell: "tcli tg member promote {{ trident.initial_tg.ident }} {{ item.key }}"
  with_dict: "{{ trident_admins }}"
  when: ansible_lsb.codename == "jessie" and tcli_login.stdout == "Login successful"
  become: yes
  tags: [ trident ]

<snip>

#EOF

```

At the end of the role, there are now admin accounts that can be immediately used to set up other trust groups and other mailing lists, as well begin and continue the process of curating memberships of these trust groups.

To set these things up yourself, follow these commands:

Now you should have a pretty good understanding of how tcli works. Always remember to login and then “swapadmin” when you need to change and customize things.

## 7.7 Configuring Trident via web app

Once Trident is running and DNS is working properly, to get to the web GUI, you will navigate to `trident.$category.$deployment` in your web browser, given what development category and DIMS deployment you are in.

This will open the following home page:

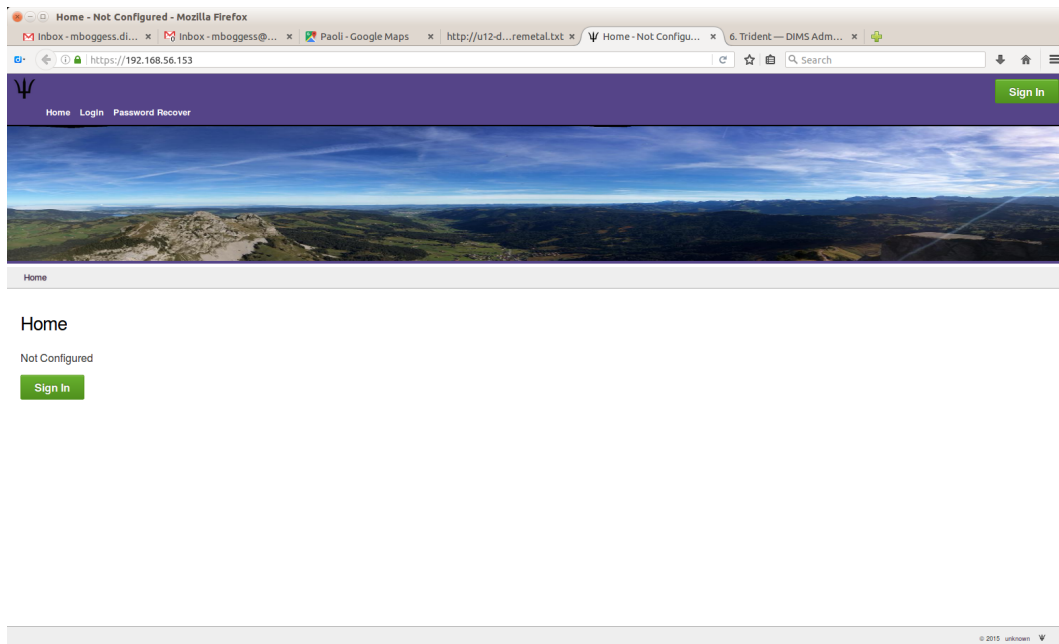


Fig. 7.1: Trident home page

To login, click the sign-in button, which will take you to the following page where you can enter your login information:

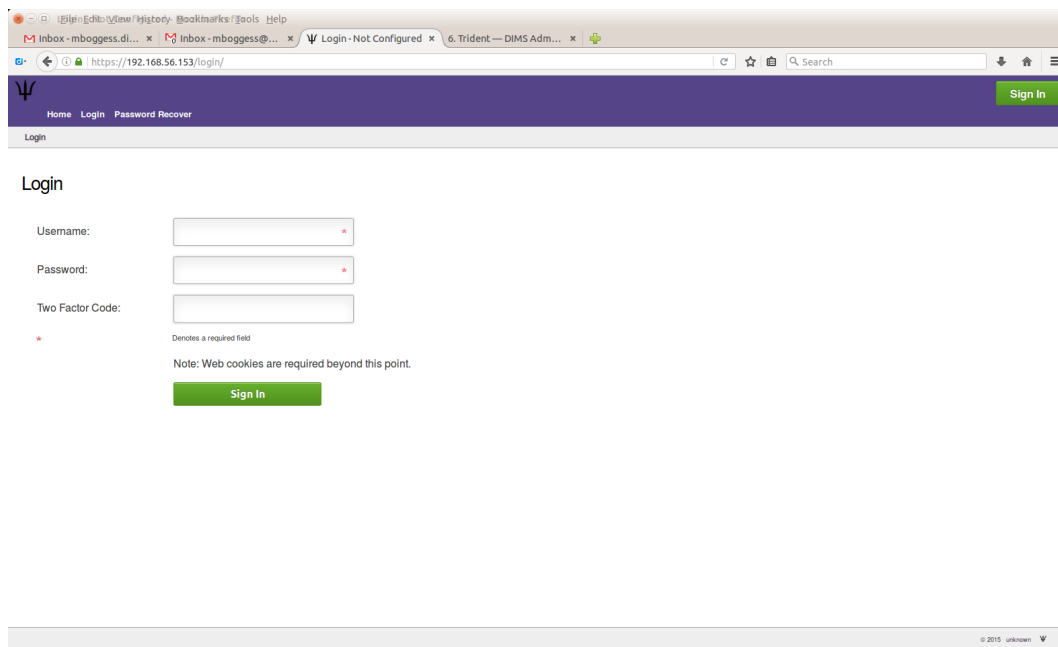


Fig. 7.2: Trident login page

The next page that opens will be a more or less blank page until you set up some trust groups:

In the top right corner will be your profile image (though it will just say “Profile Image” until you upload one), as well as the Trident system name (unconfigured at the beginning), your username, your “UserMode” or status, and the logout link. The “UserMode” is either “Regular” or “Sysadmin”. You must have system administration access in order to anything besides edit your own profile and look at trust group information of trust groups you are in.

To switch to a “Sysadmin” UserMode, click the “Regular” UserMode link in the top right corner. This will swap you to “Sysadmin” status and the page will slightly change. This is shown below:

Changing to “sysadmin” allows you to add and configure trust groups, to have access to the Trident command line interface, tcli (or “tickly”), and to view and monitor reports, logs, and settings for this particular Trident system.

### 7.7.1 User configurations

This section walks through the configuration of a user who has sysadmin privileges. There are a couple differences between what a “regular” user can configure and what a “sysadmin” user can configure. The “password reset” section is not available to users without sysadmin privileges. Additionally, there are a couple profile items hidden from regular users.

To begin, click the “User” tab at the top of the page. This will take you to a table of contents page with links to various things you can edit or look at for your user. These are also itemized in the second row at the top of the page.

To edit the user’s profile, click the “Profile” link, either in the table of contents list or in the second row at the top of the page. This will take you to a page where you can edit profile information for the user.

To update the profile, make sure to scroll all the way through all the options, and at the end of the page, there is the “Update Profile” button. This will leave you at the Profile page, but if you scroll all the way back down, you’ll see a notice about how many fields were update and how many were not modified.

You can change your user’s username:

You can change your user’s password:



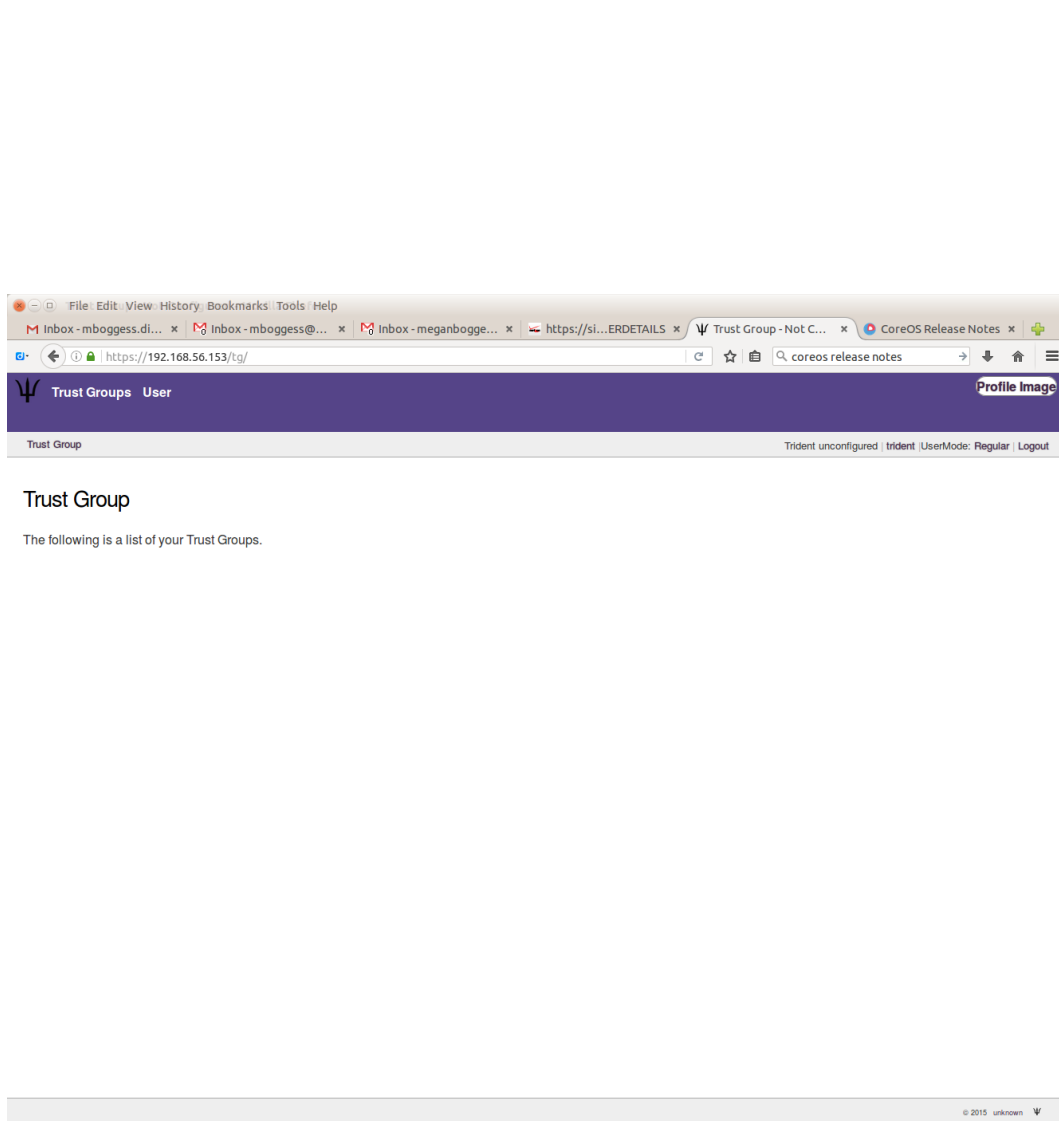


Fig. 7.3: Trident initial login page

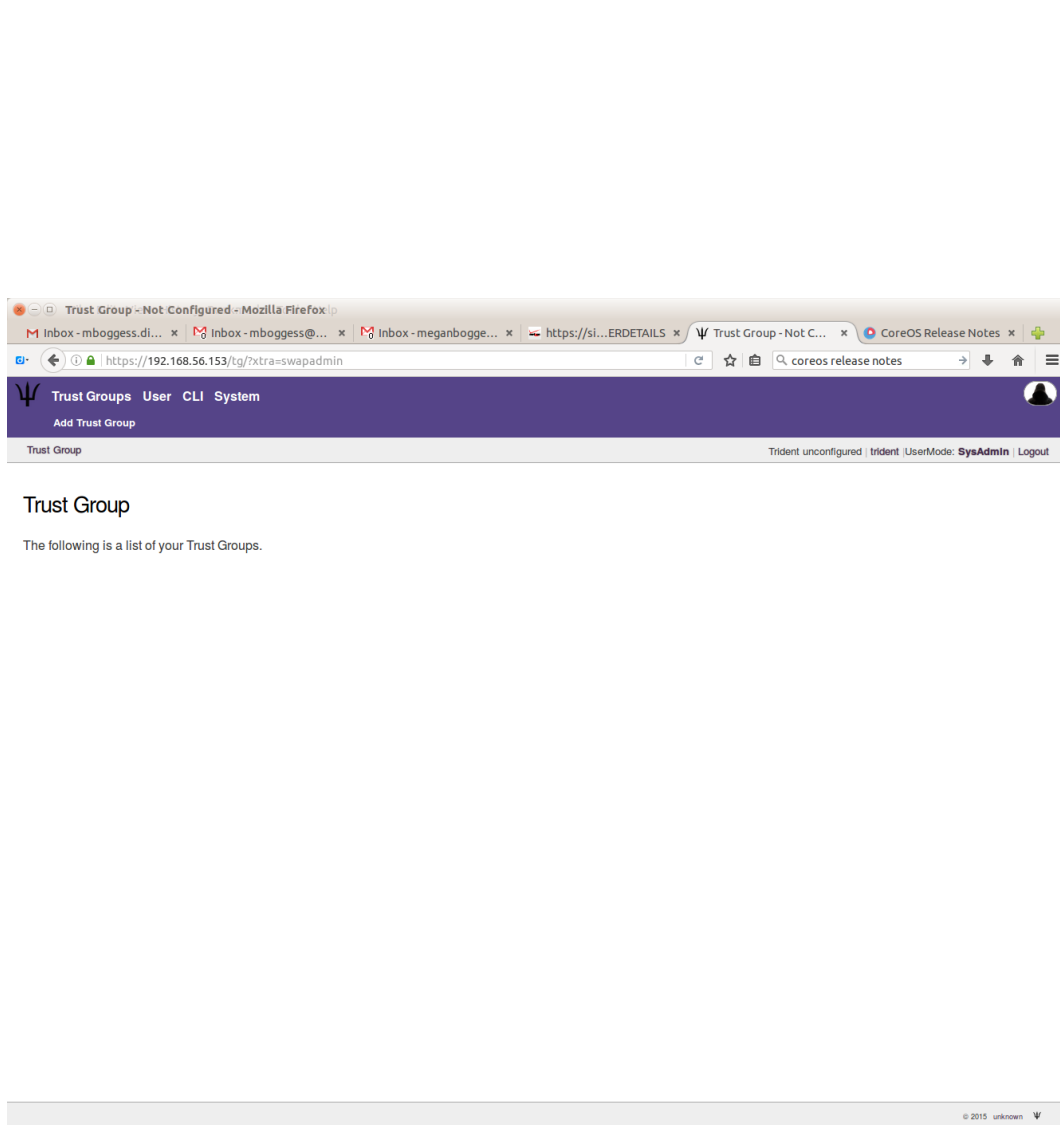


Fig. 7.4: Change to sysadmin

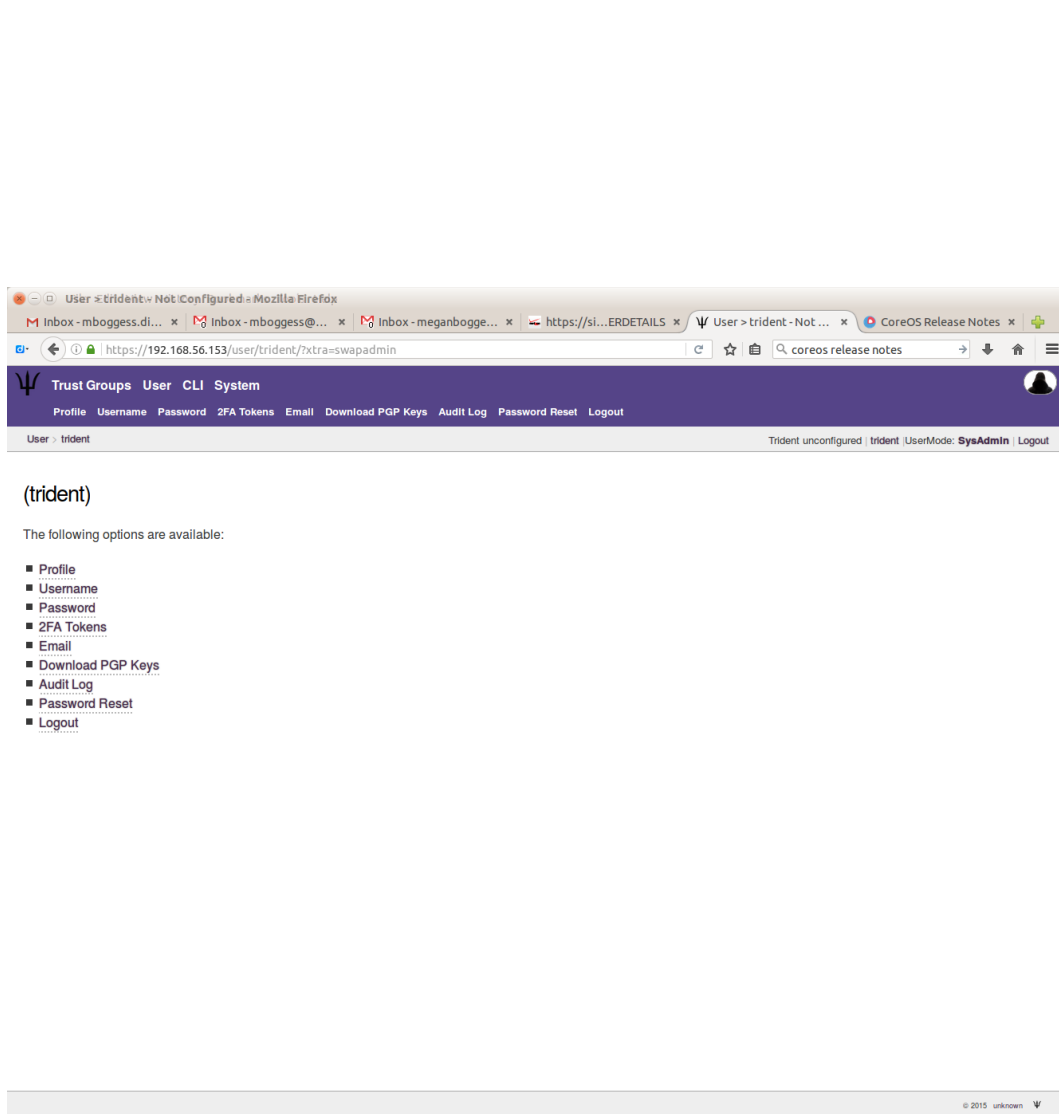
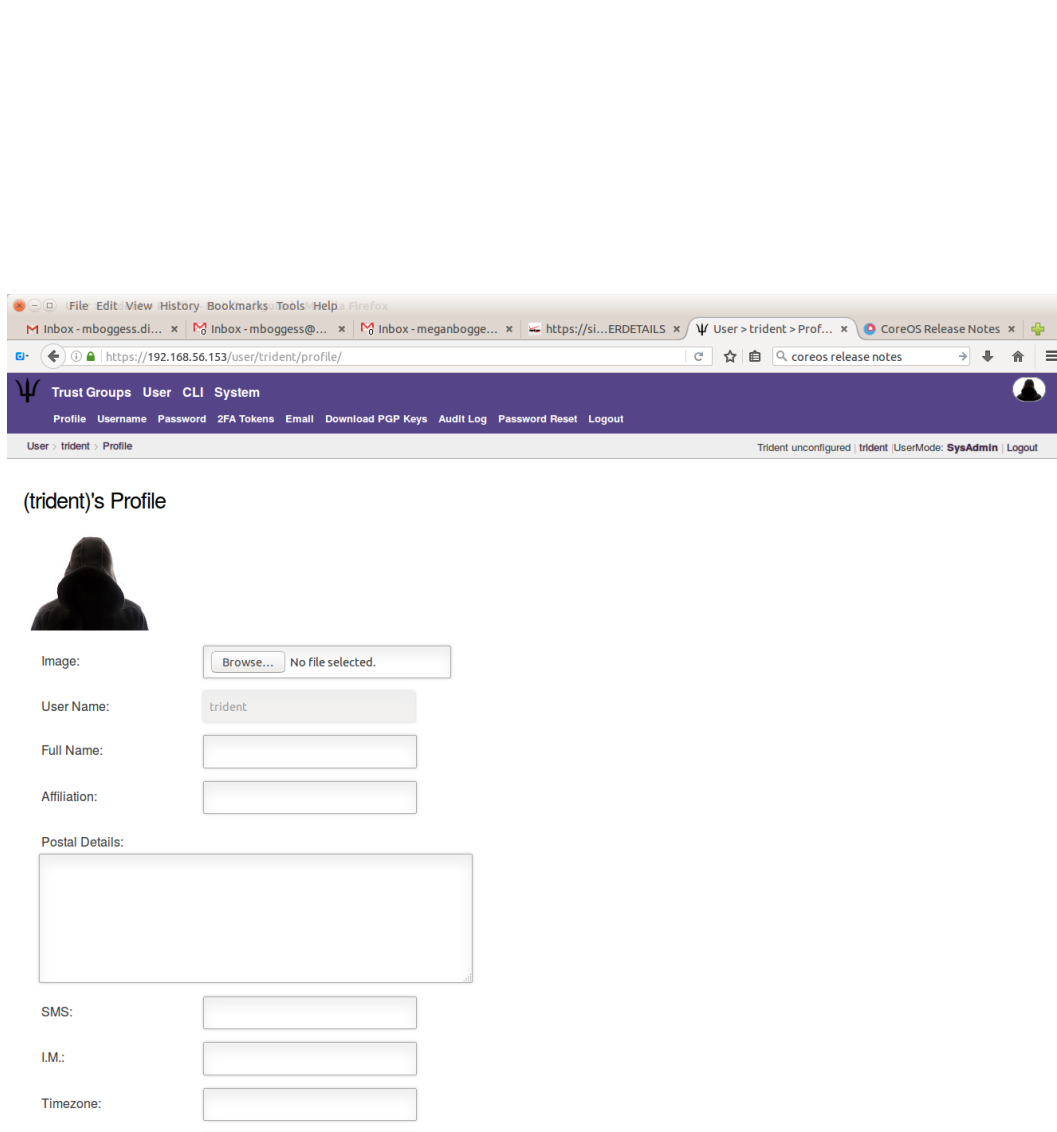


Fig. 7.5: Options for editing a user



The screenshot shows a web browser window with the URL `https://192.168.56.153/user/trident/profile/`. The browser's address bar and tabs are visible at the top. The page has a purple header with the DIMS logo and navigation links: **Trust Groups**, **User**, **CLI**, and **System**. Below the header, a sub-header contains links: **Profile**, **Username**, **Password**, **2FA Tokens**, **Email**, **Download PGP Keys**, **Audit Log**, **Password Reset**, and **Logout**. The main content area is titled **(trident)'s Profile** and features a profile picture placeholder. Below the picture are several form fields: **Image:** with a **Browse...** button and **No file selected.** text; **User Name:** with a text box containing `trident`; **Full Name:** with an empty text box; **Affiliation:** with an empty text box; **Postal Details:** with a large empty text area; **SMS:** with an empty text box; **IM:** with an empty text box; and **Timezone:** with an empty text box. The bottom of the page shows the user's current status: **Trident unconfigured | trident (UserMode: SysAdmin) | Logout**.

Fig. 7.6: Profile options

The screenshot shows a web browser window with the URL `https://192.168.56.153/user/trident/profile/`. The browser tabs include 'Inbox - mboguess.di...', 'Inbox - mboguess@...', 'Inbox - meganbogge...', 'https://si...ERDETAILS', 'User > trident > Prof...', and 'CoreOS Release Notes'. The form contains the following fields and controls:

- IM:** A text input field.
- Timezone:** A dropdown menu showing 'Eastern Standard Time'.
- Telephone:** A text input field containing '222-555-1234'.
- Airport:** A text input field.
- Biography:** A large text area.
- System Administrator:** A checkbox that is checked.
- Number of failed Login Attempts:** A numeric input field with the value '0'.
- Email Disabled:** A checkbox.
- Hide email address:** A checkbox.
- Furlough:** A checkbox.
- Entered:** A date/time input field showing '2016-09-08T18:37'.
- Last Activity:** A date/time input field showing '2016-09-09T06:15'.
- Update Profile:** A green button at the bottom of the form.

At the bottom right of the page, there is a copyright notice: '© 2015 unknown'.

Fig. 7.7: Profile update

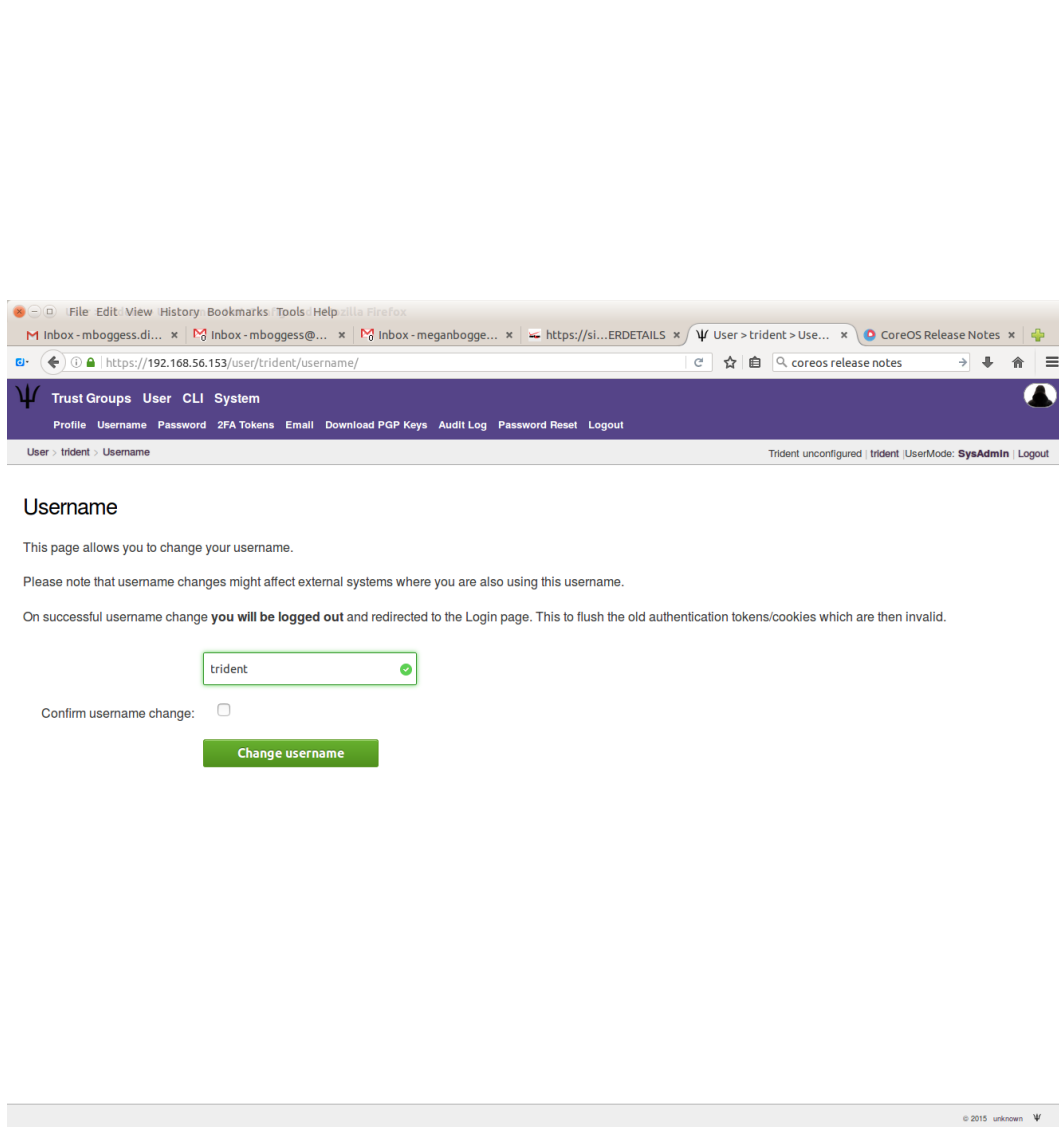


Fig. 7.8: Change user's username

The screenshot shows a web browser window with the URL `https://192.168.56.153/user/trident/password/`. The page has a purple header with navigation links: **Trust Groups**, **User**, **CLI**, and **System**. Below the header, there's a sub-menu with **Profile**, **Username**, **Password**, **2FA Tokens**, **Email**, **Download PGP Keys**, **Audit Log**, **Password Reset**, and **Logout**. The main content area is titled **Password** and contains the text: "This page allows you to change your password." Below this, there are three input fields: **Username:** (pre-filled with "trident"), **New password:**, and **Repeat new password:**. Each of the last two fields has a red asterisk icon on the right. A legend below the fields shows a red asterisk and the text "Denotes required field". At the bottom of the form is a green button labeled **Change Password**. The footer of the page shows "© 2015 unknown" and a small logo.

Fig. 7.9: Change user's password

You can set up two-factor authentication:

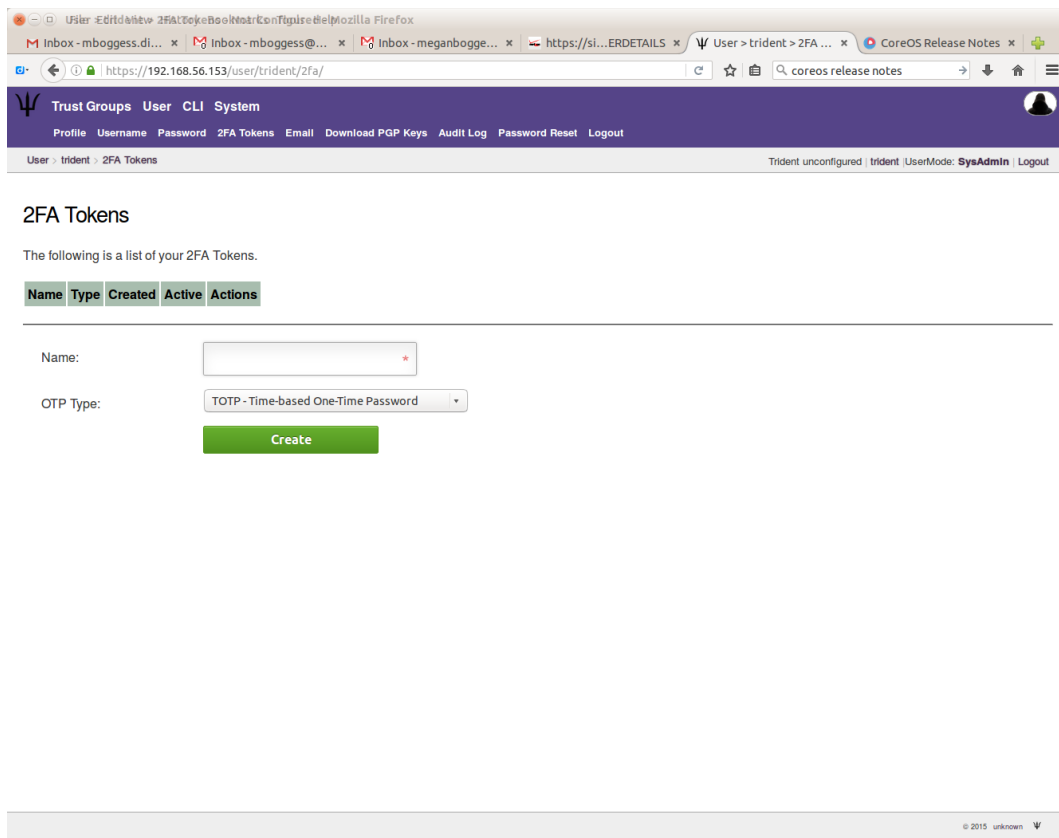


Fig. 7.10: Setup two-factor authentication

You must add and verify your email address to receive emails from trust groups to which you belong. First, “create” your email:

Once you submit your email address, you must get a verification code. Click the “Verify” button on this page to get the verification code sent to you via email:

Once you receive the email with the code, put the code in the “Verification Code” box on the following page:

If it is a valid verification code, your email’s status will change from “Unverified” to “Verified”.

You can also download your user’s PGP keys:

You can also view an audit log for your user:

As a “sysadmin” user, you can do all of these things for all users under your administration. A list of these users can be found by clicking the “User” tab in the second row at the top of the page, when in “Sysadmin” UserMode.

Additionally, only a sysadmin can reset another user’s password or remove an email address.

## 7.7.2 Sysadmin configurations

Sysadmins can set up trust groups, view information about their system, and use the Trident command line interface, tcli (or “tickly”), through the web app. This section walks through these features.



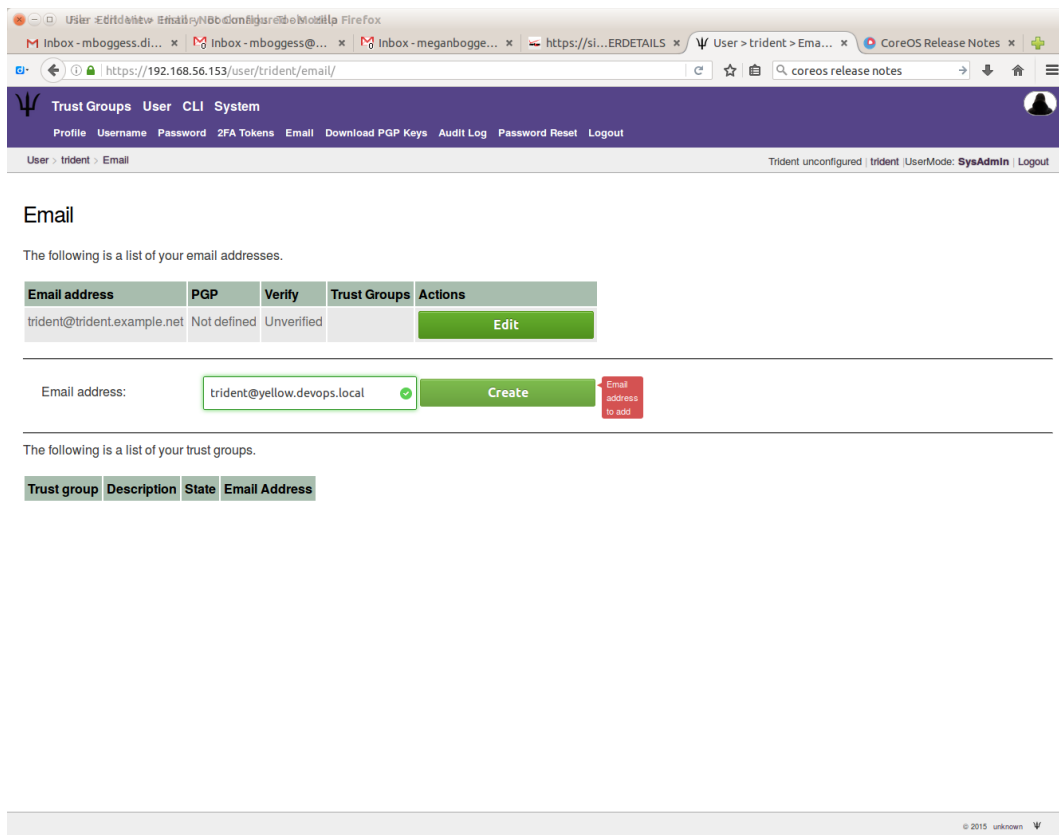


Fig. 7.11: Create user email

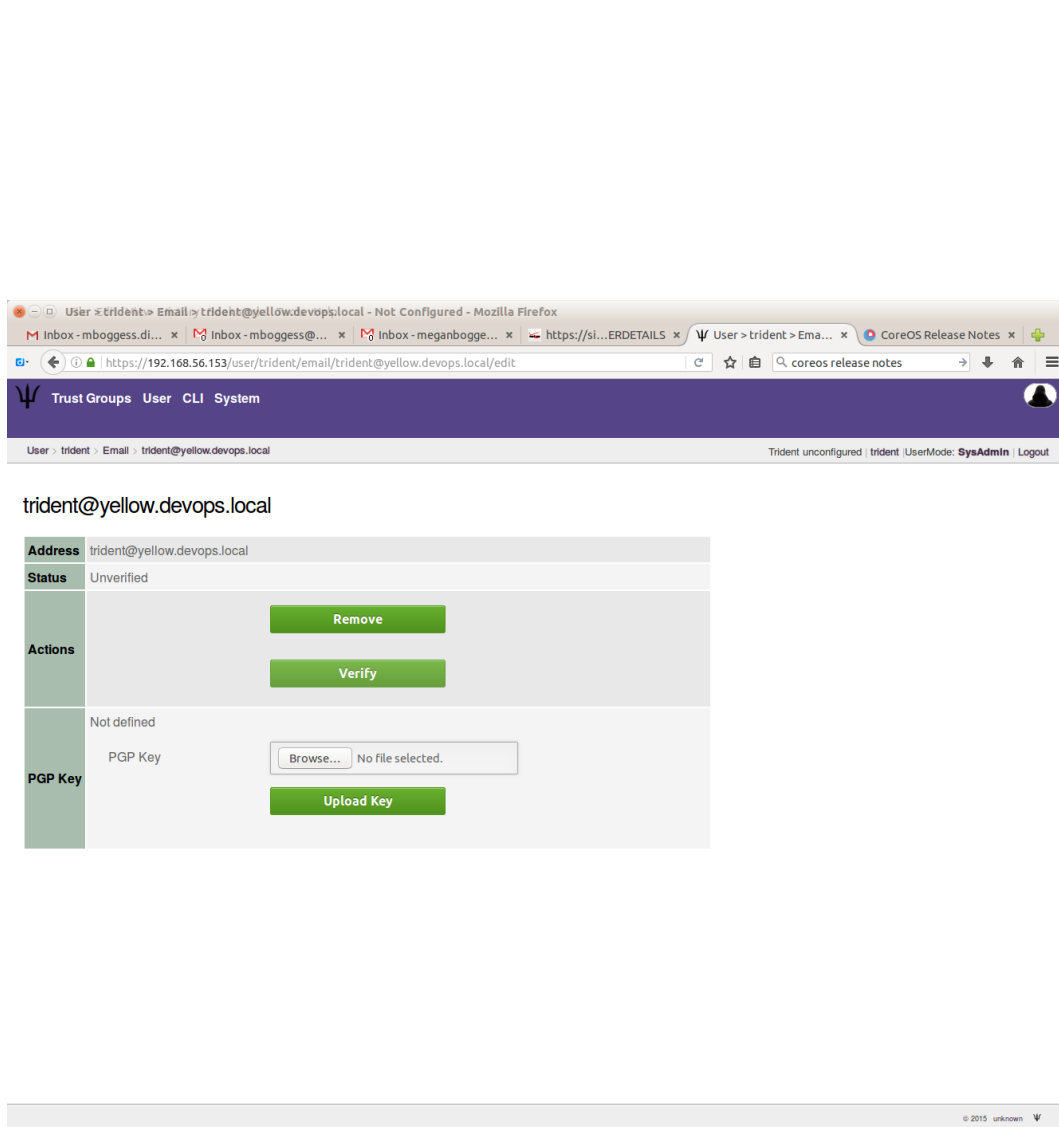


Fig. 7.12: Verify user email

The screenshot shows a web browser window with the URL `https://192.168.56.153/user/trident/email/trident@yellow.devops.local/edit`. The page title is "User > trident > Email > trident@yellow.devops.local". The navigation bar includes links for "Trust Groups", "User", "CLI", and "System". The breadcrumb trail is "User > trident > Email > trident@yellow.devops.local". The status bar shows "Trident unconfigured | trident | UserMode: SysAdmin | Logout".

The main content area is titled "trident@yellow.devops.local" and contains a form with the following sections:

<b>Address</b>	trident@yellow.devops.local	
<b>Status</b>	Verification in Process	
<b>Actions</b>		<a href="#">Remove</a>
	Verification Code	<input type="text" value="-cH2ESucP58pEkre"/>
		<a href="#">Confirm</a>
		<a href="#">Resend</a>
<b>PGP Key</b>	Not defined	
	PGP Key	<a href="#">Browse...</a> No file selected.
		<a href="#">Upload Key</a>

The footer of the page contains the text "© 2015 unknown" and a small logo.

Fig. 7.13: Submit verification code

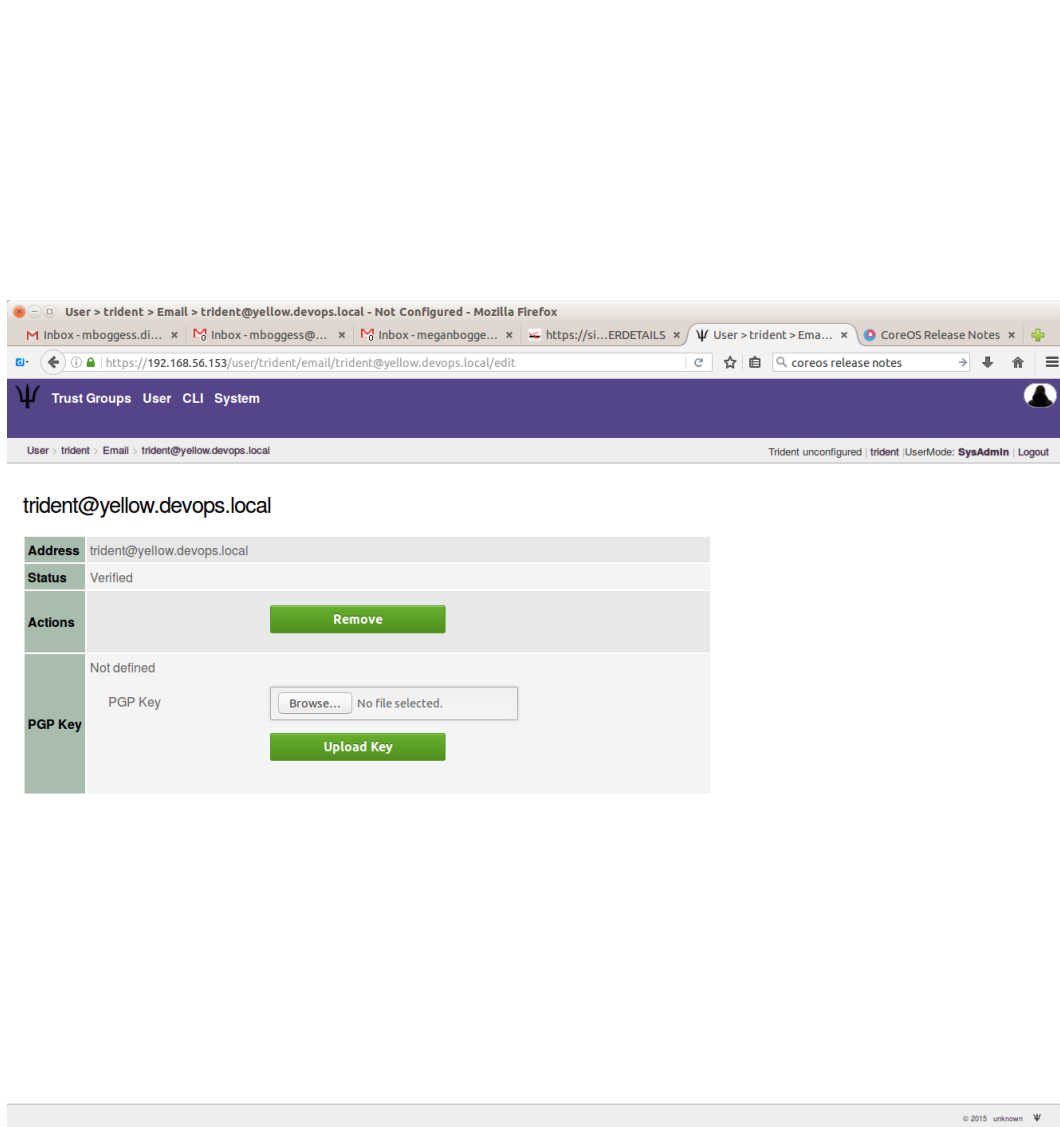


Fig. 7.14: Verified email status

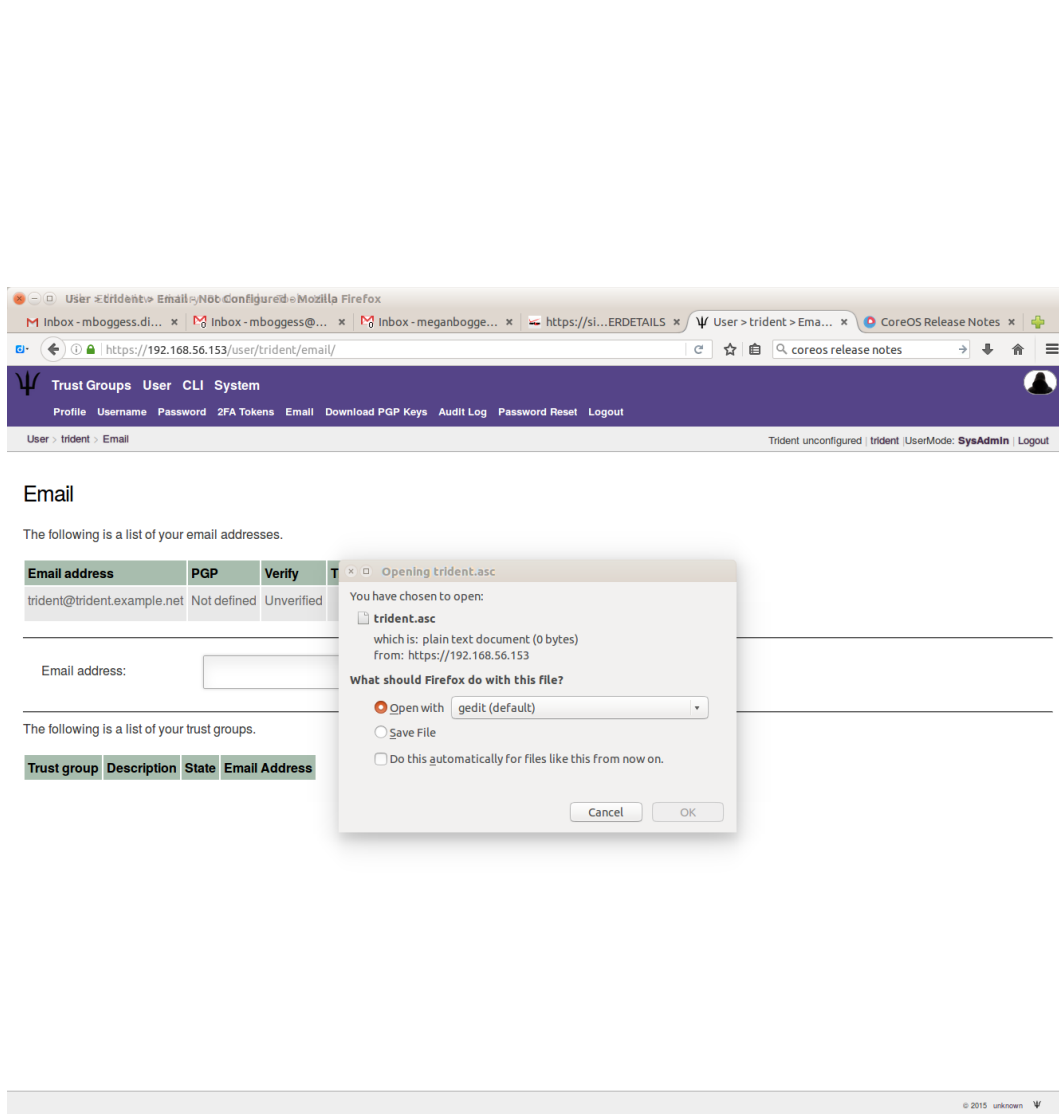


Fig. 7.15: Download PGP keys

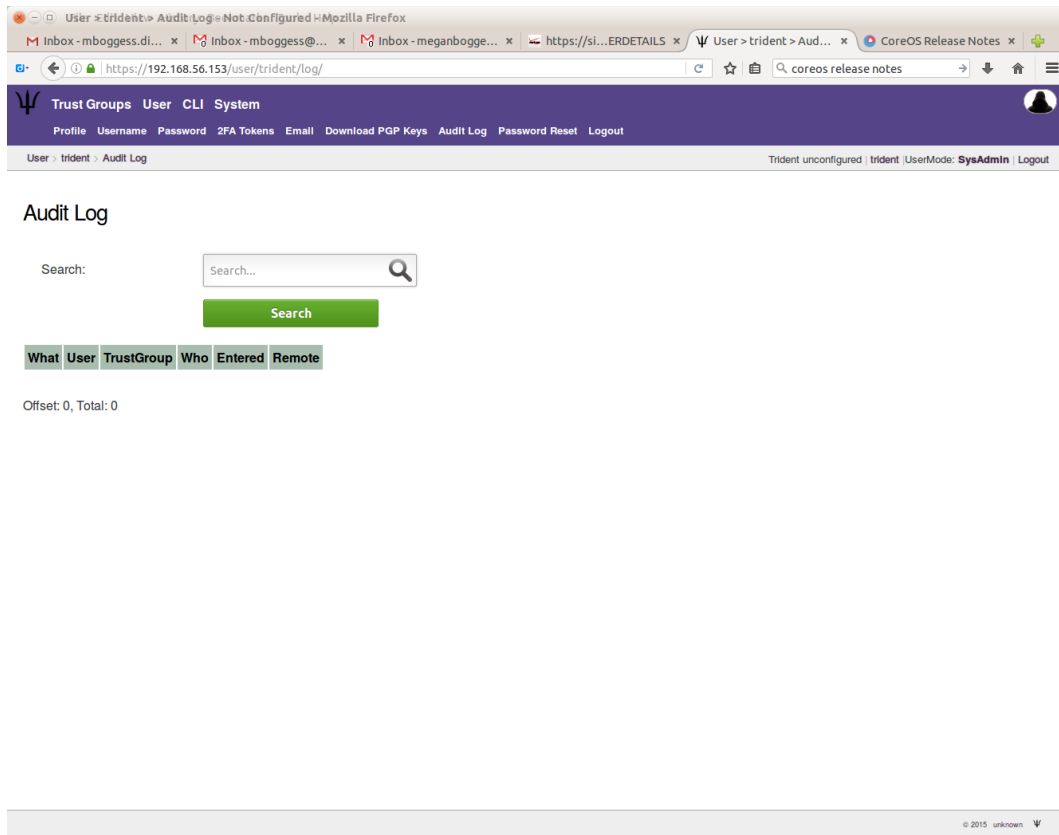


Fig. 7.16: View user audit log

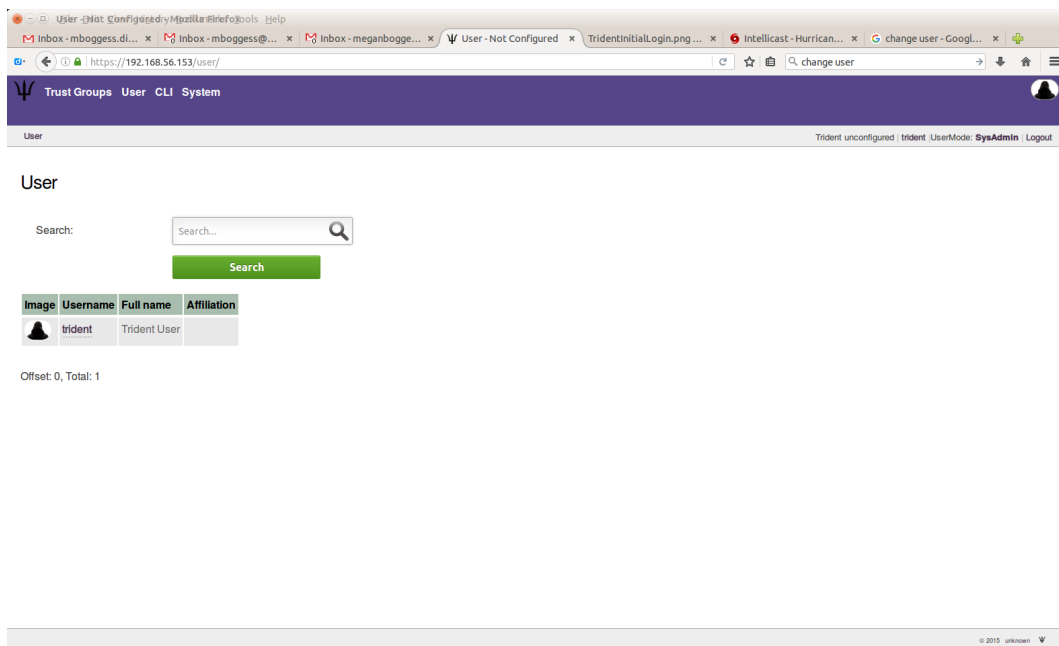


Fig. 7.17: View user list as sysadmin

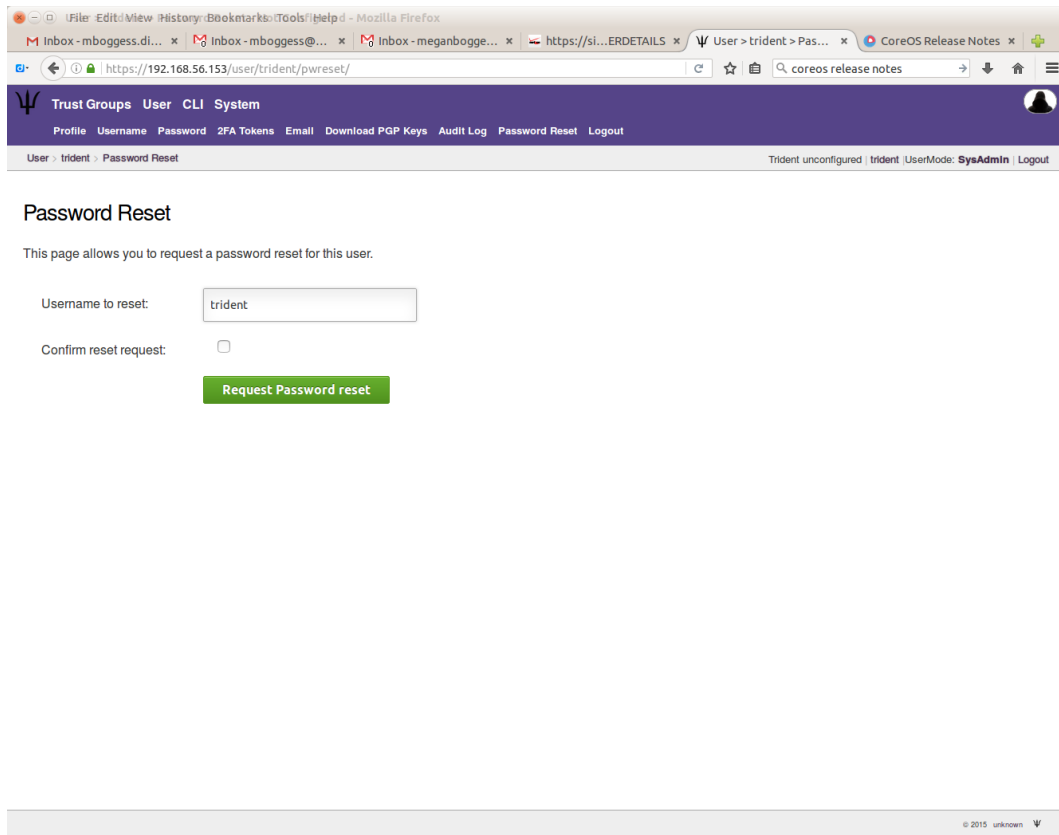


Fig. 7.18: Reset a user's password as sysadmin

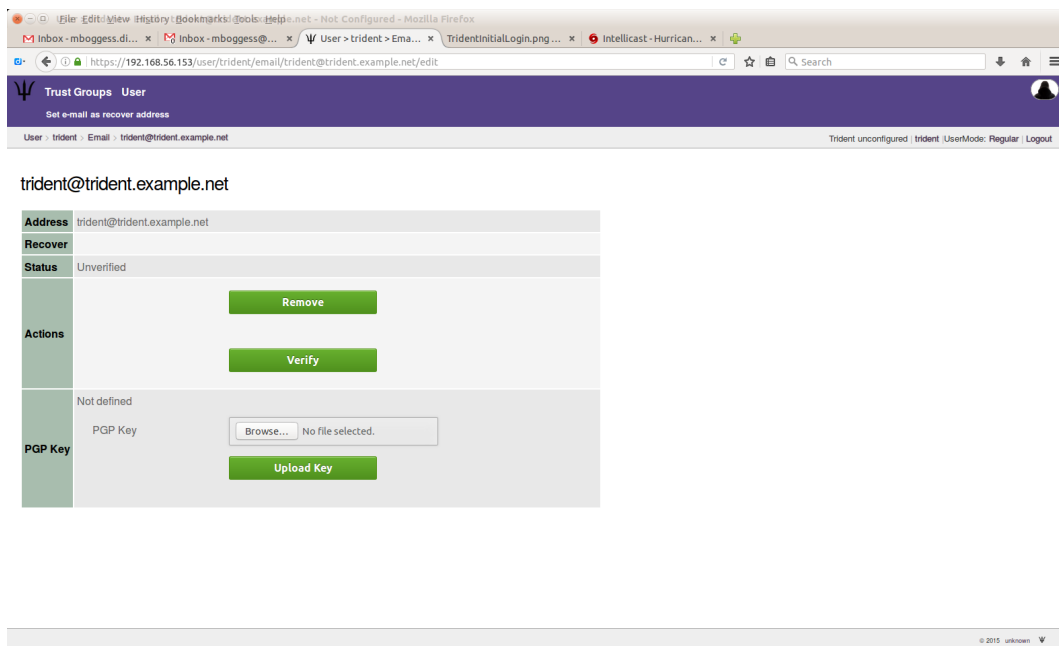


Fig. 7.19: Remove an email as sysadmin

## Trust group configurations

The initial login page will list your trust groups. If you don't have any, or to add new ones, click the "Add Trust Group" link in the second row at the top of the page.

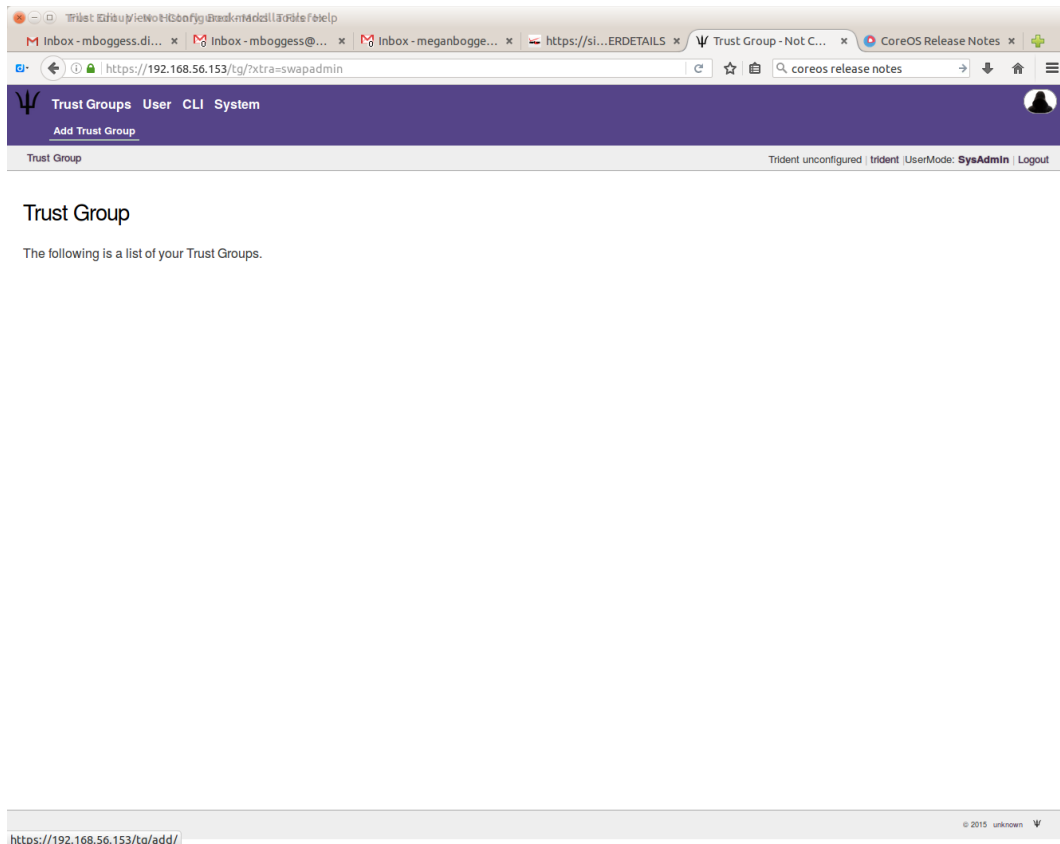


Fig. 7.20: No trust groups, yet.

The following page will start the configuration of the trust group, starting with a name for the trust group.

**Warning:** If there isn't at least one verified email address, this will fail.

Once you have at least one trust group, clicking the "Trust Group" tab at the top of the page will give you an index of the trust groups you have access to. This list can be seen as a regular user or as a sysadmin user, as can be seen by this page (shown from the regular user perspective):

As a sysadmin user, however, you can do much more than just view a list of trust groups. For all trust groups under your administration, you can manage users, set up mailing lists, view audit logs, set up and use wiki and file storage, as well as set other configurations and download PGP keys.

In order to have access to the wiki and file storage, you must set that up via the group settings:

You must select the "Wiki Module" and "Files Module" if you want to use those features:

Trust group wiki:

Trust group files:



The screenshot shows a web browser window with the URL `https://192.168.56.153/tg/add/`. The browser's address bar and tabs are visible at the top. The web application has a dark purple header with a logo on the left and navigation links: "Trust Groups", "User", "CLI", and "System". On the right side of the header is a user profile icon. Below the header, a breadcrumb trail reads "Trust Group > add". The main content area is titled "Add Trust Group" and includes a sub-header: "This form allows the creation of a new Trust Group. Only the name is requested here, further details are configured in the edit menu after the group has been created." Below this text is a form with a label "Group Name:" followed by a text input field containing "PenTest" and a green checkmark icon. A green "Create" button is positioned below the input field. At the bottom right of the page, a footer contains the text "© 2015 Trident User" and a small logo.

Trident unconfigured | trident | UserMode: **SysAdmin** | Logout

### Add Trust Group

This form allows the creation of a new Trust Group. Only the name is requested here, further details are configured in the edit menu after the group has been created.

Group Name:

Create

© 2015 Trident User

Fig. 7.21: Add a trust group

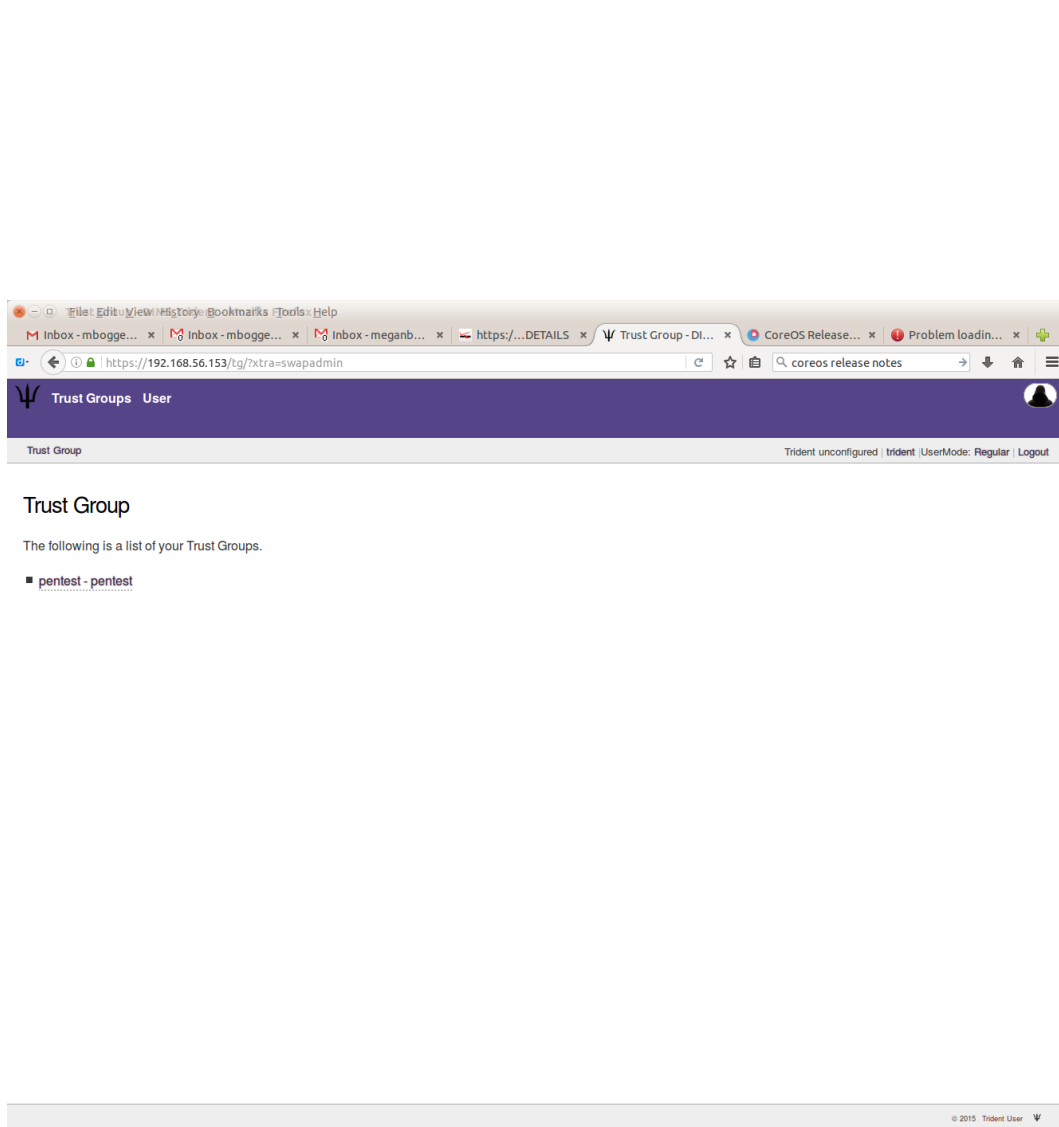


Fig. 7.22: List of trust groups

The screenshot shows a web browser window displaying the DIMS Administrator interface. The browser's address bar shows the URL `https://192.168.56.153/tg/PenTest/settings/`. The application's header is purple with a navigation menu containing 'Trust Groups', 'User', 'CLI', and 'System'. Below this, a secondary menu includes 'Settings', 'Members', 'Nominate', 'PGP Keys', 'Mailing List', and 'Audit Log'. The current page is titled 'Settings' and is for the 'pentest' trust group. The settings are organized into a list of fields with labels on the left and input controls on the right. The 'Please Vouch' checkbox is checked, and 'Nominations Enabled' is also checked. The 'Idle Guard' is set to 168:00:00. The 'Maximum Inactivity' is set to 4320:00:00. The 'Can Time Out' checkbox is unchecked. The 'Wiki Module' checkbox is unchecked. The 'Trident' status is shown as 'unconfigured' in the top right corner of the page content area.

Group Name:	<input type="text" value="pentest"/>
Description:	<input type="text" value="pentest"/>
PGP Required:	<input type="checkbox"/>
Please Vouch:	<input checked="" type="checkbox"/>
Vouch TG Admins Only:	<input type="checkbox"/>
Minimum Inbound Vouches:	<input type="text" value="0"/>
Minimum Outbound Vouches:	<input type="text" value="0"/>
Target Invouches:	<input type="text" value="0"/>
Maximum Inactivity:	<input type="text" value="4320:00:00"/>
Can Time Out:	<input type="checkbox"/>
Maximum Vouch Days:	<input type="text" value="0"/>
Idle Guard:	<input type="text" value="168:00:00"/>
Nominations Enabled:	<input checked="" type="checkbox"/>
Wiki Module:	<input type="checkbox"/>

Fig. 7.23: Some trust group settings

The screenshot shows a web browser window with the URL `https://192.168.56.153/tg/PenTest/settings/`. The page has a purple header with navigation links: **Trust Groups**, **User**, **CLI**, and **System**. Below these are sub-links: **Settings**, **Members**, **Nominate**, **PGP Keys**, **Mailing List**, and **Audit Log**. The main content area is titled **Settings** and contains the following configuration options for the 'pentest' group:

- Group Name:** pentest
- Description:** pentest
- PGP Required:** ☐
- Please Vouch:** ☒
- Vouch TG Admins Only:** ☐
- Minimum Inbound Vouches:** 0
- Minimum Outbound Vouches:** 0
- Target Invouches:** 0
- Maximum Inactivity:** 4320:00:00
- Can Time Out:** ☐
- Maximum Vouch Days:** 0
- Idle Guard:** 168:00:00
- Nominations Enabled:** ☒
- Wiki Module:** ☐

The top right of the page shows the status: **Trident unconfigured | trident (UserMode: SysAdmin | Logout)**.

Fig. 7.24: Some trust group settings

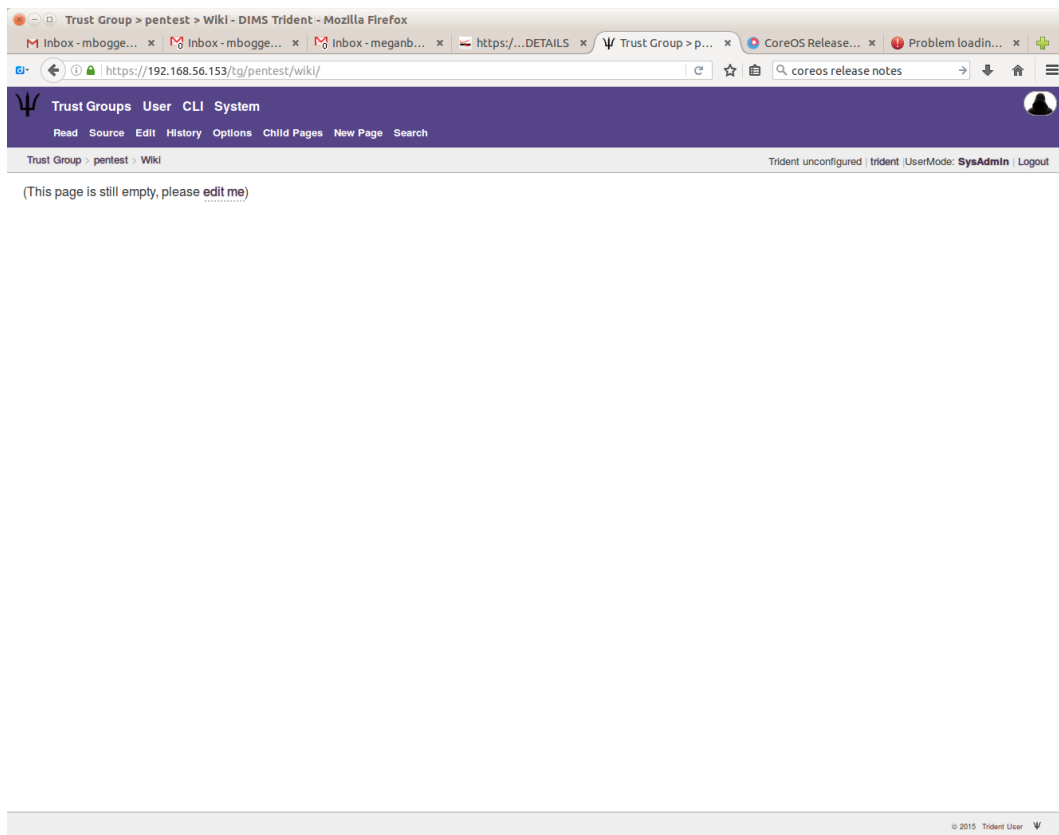


Fig. 7.25: Empty trust group wiki

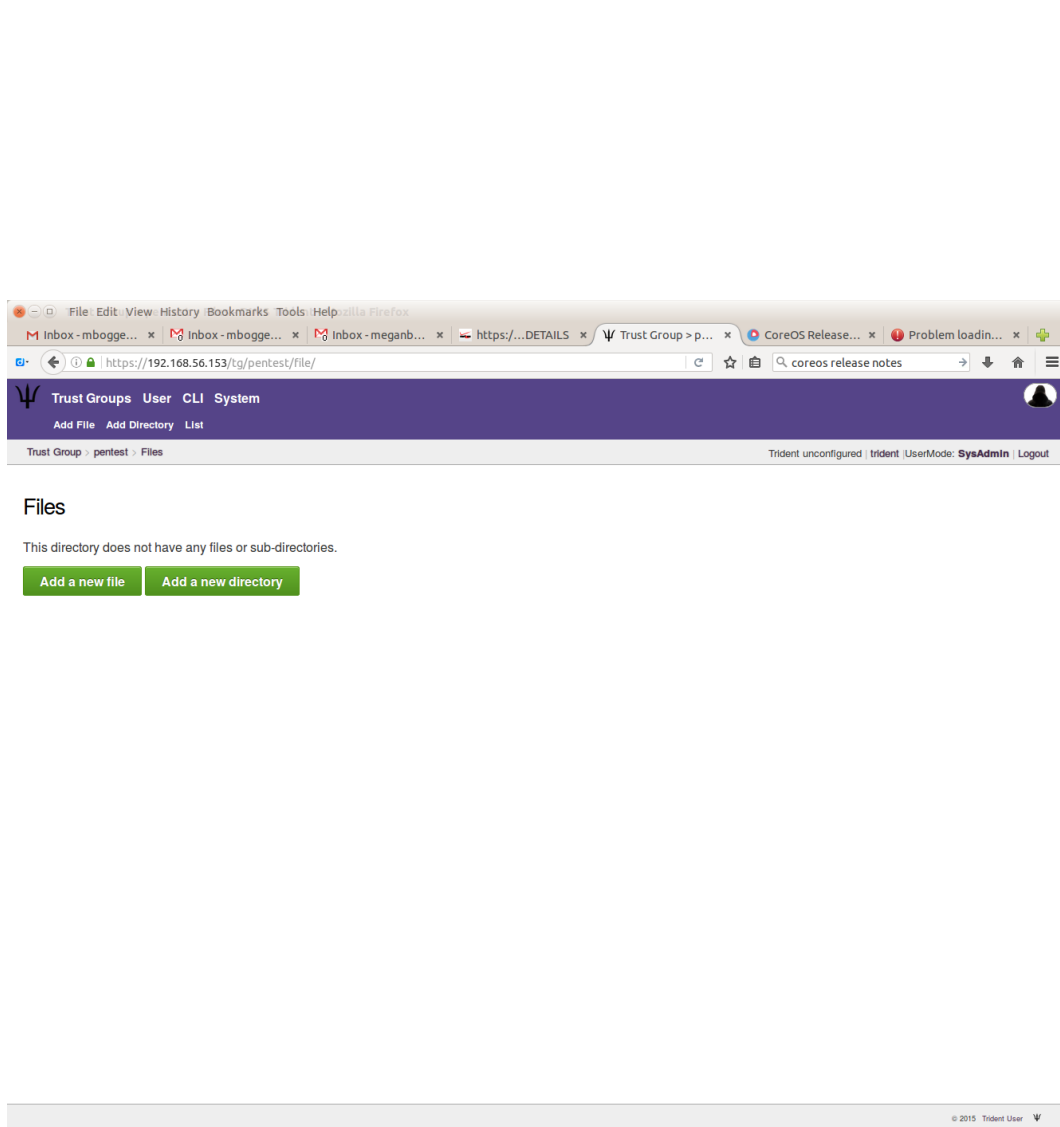


Fig. 7.26: Empty trust group file storage

You can then click the tabs near the top of the page or the green buttons in the middle of the page to “Add” a file or directory or to list the files and directories.

Download PGP keys:

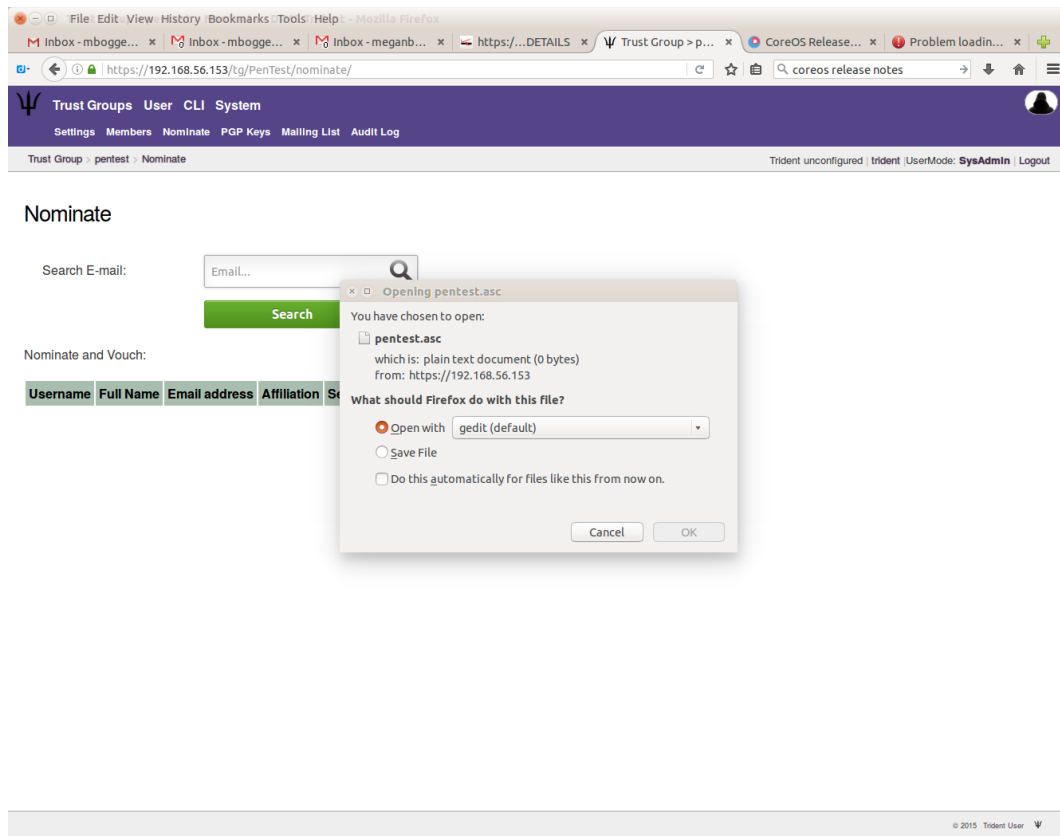


Fig. 7.27: Download trust group PGP keys

See list of trust group members:

To nominate a user, you must search for them via their email address:

To add mailing lists, choose a trust group, then click the “Mailing List” tab in the second row at the top of the page. There are some default mailing lists when you add a trust group:

Click the “New Mailing List” in the second row at the top of the page. On the next page, give your mailing list a name:

You can then see the newly added mailing list:

Once the mailing list is created, you can update its settings, subscribe or unsubscribe users, and view the PGP key.


To update a mailing list’s settings, choose a mailing list, then click the “Settings” tab in the second row at the top of the page.

If no users have been subscribed to a mailing list, you’ll see the following page:

To add a user to a mailing list, choose a trust group and a mailing list, then click the “Subscribe” tab in the second row at the top of the page. Type in the username of the user you’d like to subscribe to the list.

If the user already exists on a mailing list, you’ll see the following:

The screenshot shows a web browser window displaying the DIMS Administrator interface. The browser's address bar shows the URL `https://192.168.56.153/tg/PenTest/members/`. The interface has a purple header bar with the following navigation links: **Trust Groups**, **User**, **CLI**, and **System**. Below these links are **Settings**, **Members**, **Nominate**, **PGP Keys**, **Mailing List**, and **Audit Log**. The current page is titled **Members** and shows the **Trust Group** as **pentest**. The user is logged in as **SysAdmin**. The main content area is titled **Members** and includes a search bar with the text "Search:" and a green **Search** button. Below the search bar is a table with the following columns: **Image**, **Username**, **Full name**, **Email**, **Affiliation**, **Admin**, **Status**, **Activity**, **Vouches**, and **Actions**. The table contains one row for the user **trident (user)**, with the following details: **Full name**: Trident User, **Email**: trident@yellow.devops.local, **Affiliation**: University of Washington, **Admin**: yes, **Status**: active, **Activity**: 0 Days, **Vouches**: 0(by) -> 0(for). The **Actions** column for this user contains two buttons: **Block Member** and **Demote from Admin**. Below the table, there is a note: "(Bold) includes vouch for or by you" and "ψ = has PGP key". At the bottom of the page, it says "Offset: 0, Total: 1". The footer of the page shows "© 2015 Trident User" and a small logo.

Image	Username	Full name	Email	Affiliation	Admin	Status	Activity	Vouches	Actions
	trident (user)	Trident User	trident@yellow.devops.local	University of Washington	yes	active	0 Days	0(by) -> 0(for)	<b>Block Member</b> <b>Demote from Admin</b>

(Bold) includes vouch for or by you  
ψ = has PGP key

Offset: 0, Total: 1

© 2015 Trident User

Fig. 7.28: List of trust group members



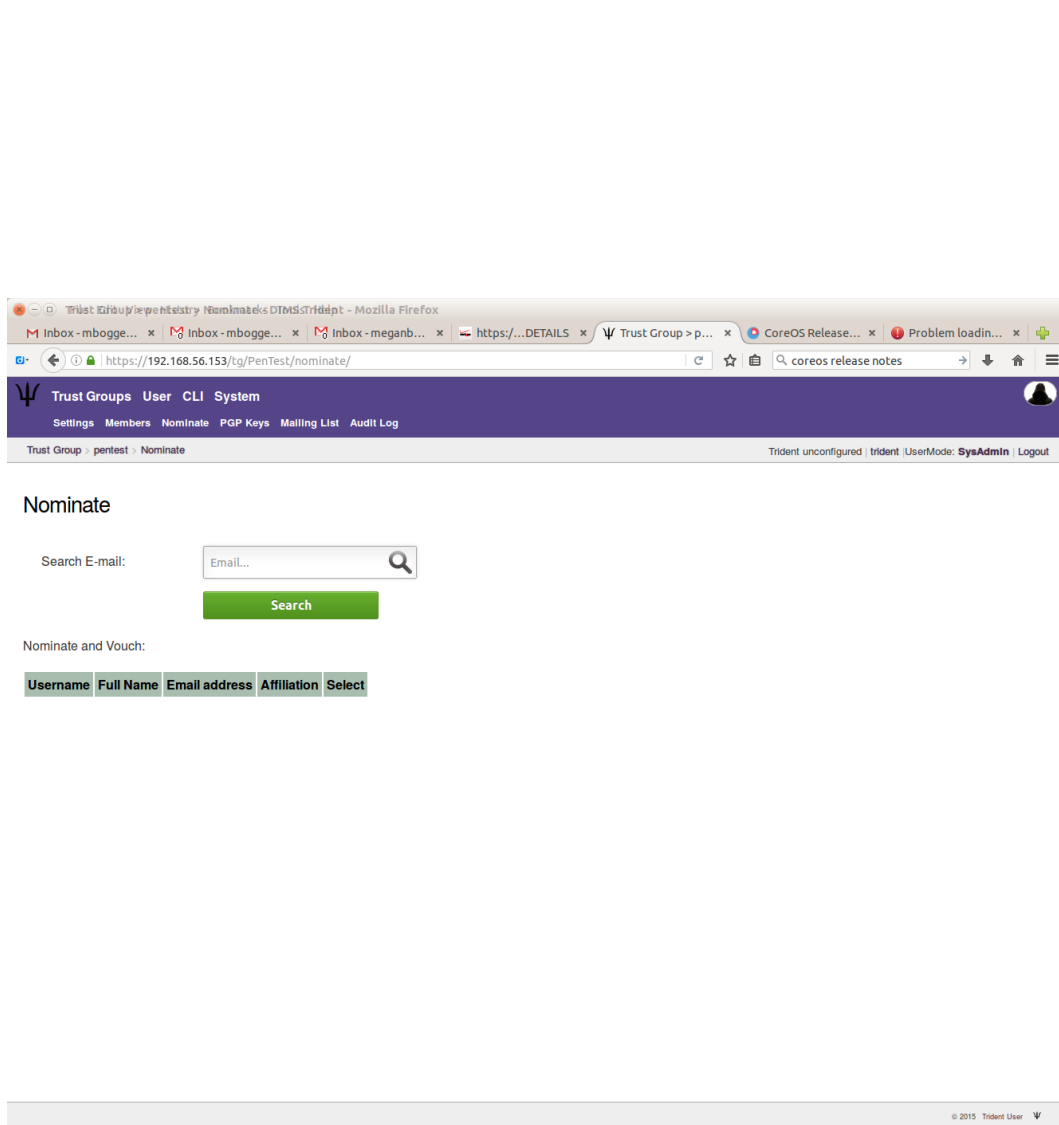


Fig. 7.29: Search by email to nominate user

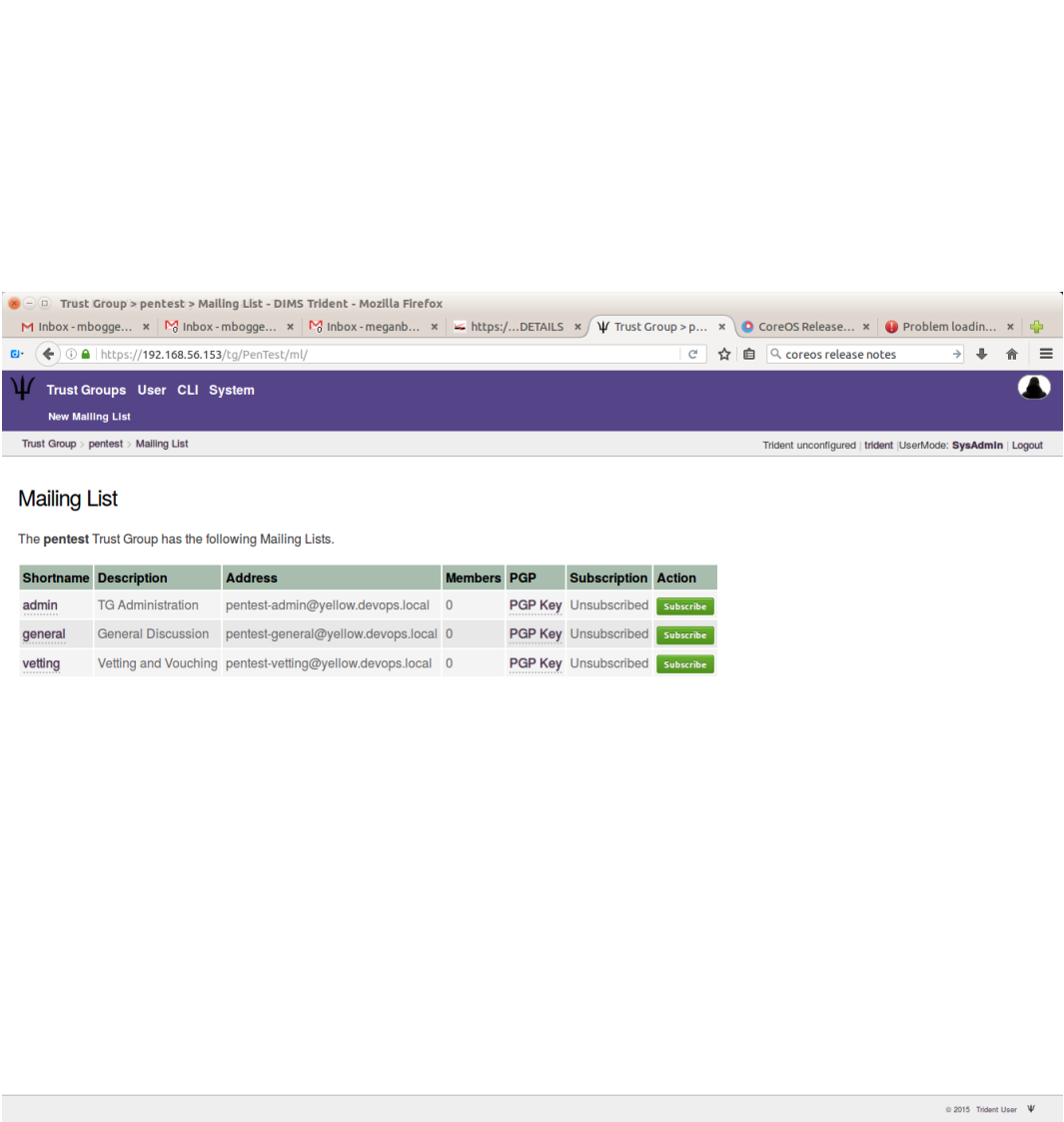


Fig. 7.30: Default trust group mailing lists

The screenshot shows a web browser window with the URL `https://192.168.56.153/user/trident/tg/pentest/ml/new/`. The browser's address bar and tabs are visible at the top. The application's header is dark purple with a logo on the left and navigation links: "Trust Groups", "User", "CLI", and "System". On the right of the header is a user profile icon. Below the header is a breadcrumb trail: "User > trident > pentest > Mailing List > new". On the right side of the header, it says "Trident unconfigured | trident | UserMode: SysAdmin | Logout".

### Add Mailing List

This form allows the creation of a new Mailing List. Only the name is requested here, further details are configured in the edit menu after the list has been created.

Group Name:

List Name:

Creates the Mailing List

At the bottom right of the page, there is a small copyright notice: "© 2015 Trident User" with a logo.

Fig. 7.31: Add trust group mailing list

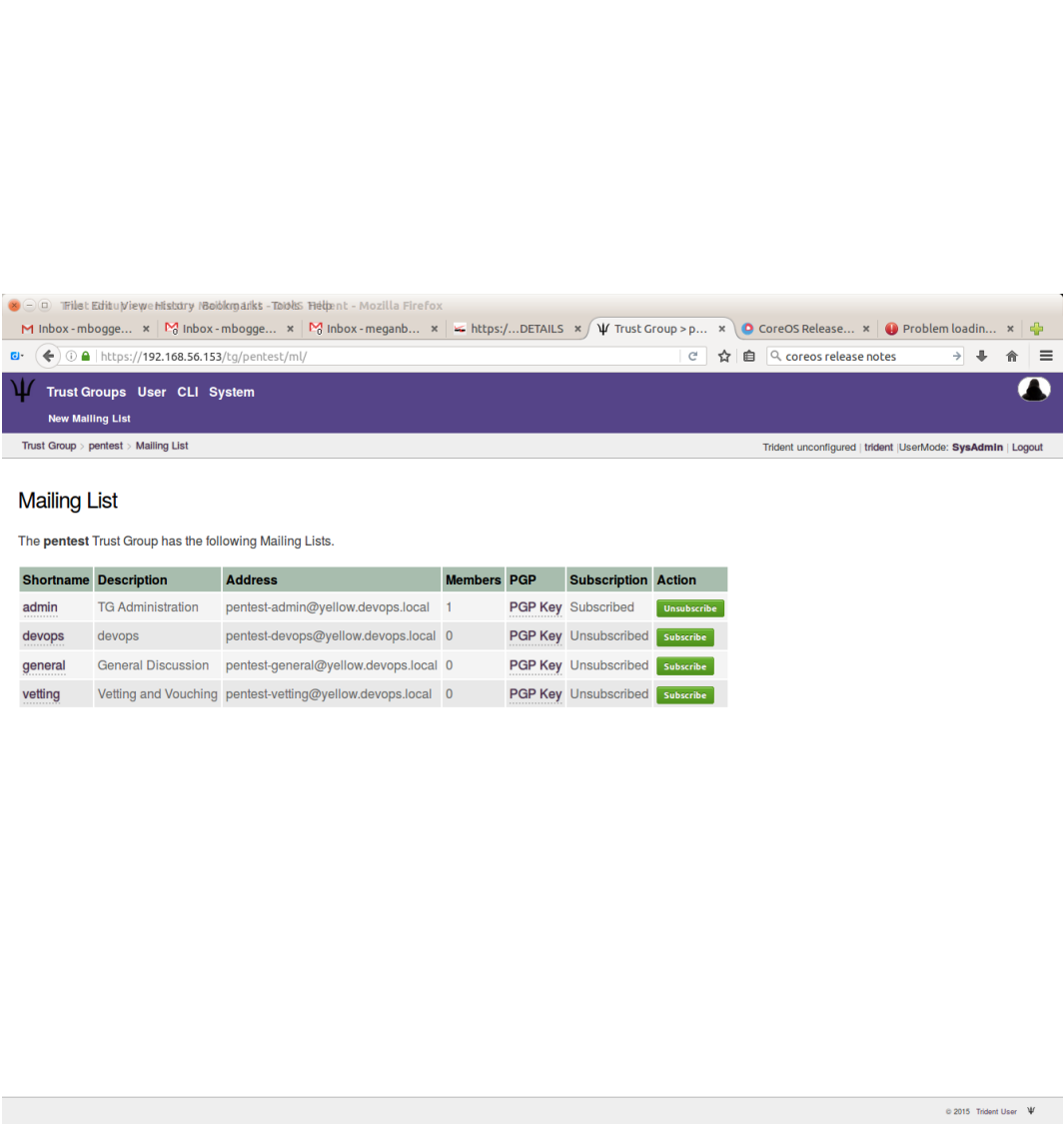


Fig. 7.32: Default and added mailing list index

The screenshot shows a web browser window with the URL `https://192.168.56.153/tg/pentest/ml/devops/settings/`. The page has a purple header with navigation links: **Trust Groups**, **User**, **CLI**, and **System**. Below these are links for **Settings**, **Subscribe**, **Unsubscribe**, and **PGP Key**. A breadcrumb trail shows `Trust Group > pentest > Mailing List > devops > Settings`. The user is logged in as **SysAdmin**. The main content area is titled **Settings** and contains the following fields:

- List Name:** `devops`
- Trust Group:** `pentest`
- Description:** `devops`
- Members Only:** ☒
- Can Add Self:** ☒
- Automatic:** ☐
- Always Encrypt:** ☐
- Update Configuration:** (Green button)
- Public PGP Key:** (Empty text area)

The footer of the page indicates `© 2015 Trident User`.

Fig. 7.33: Update mailing list settings

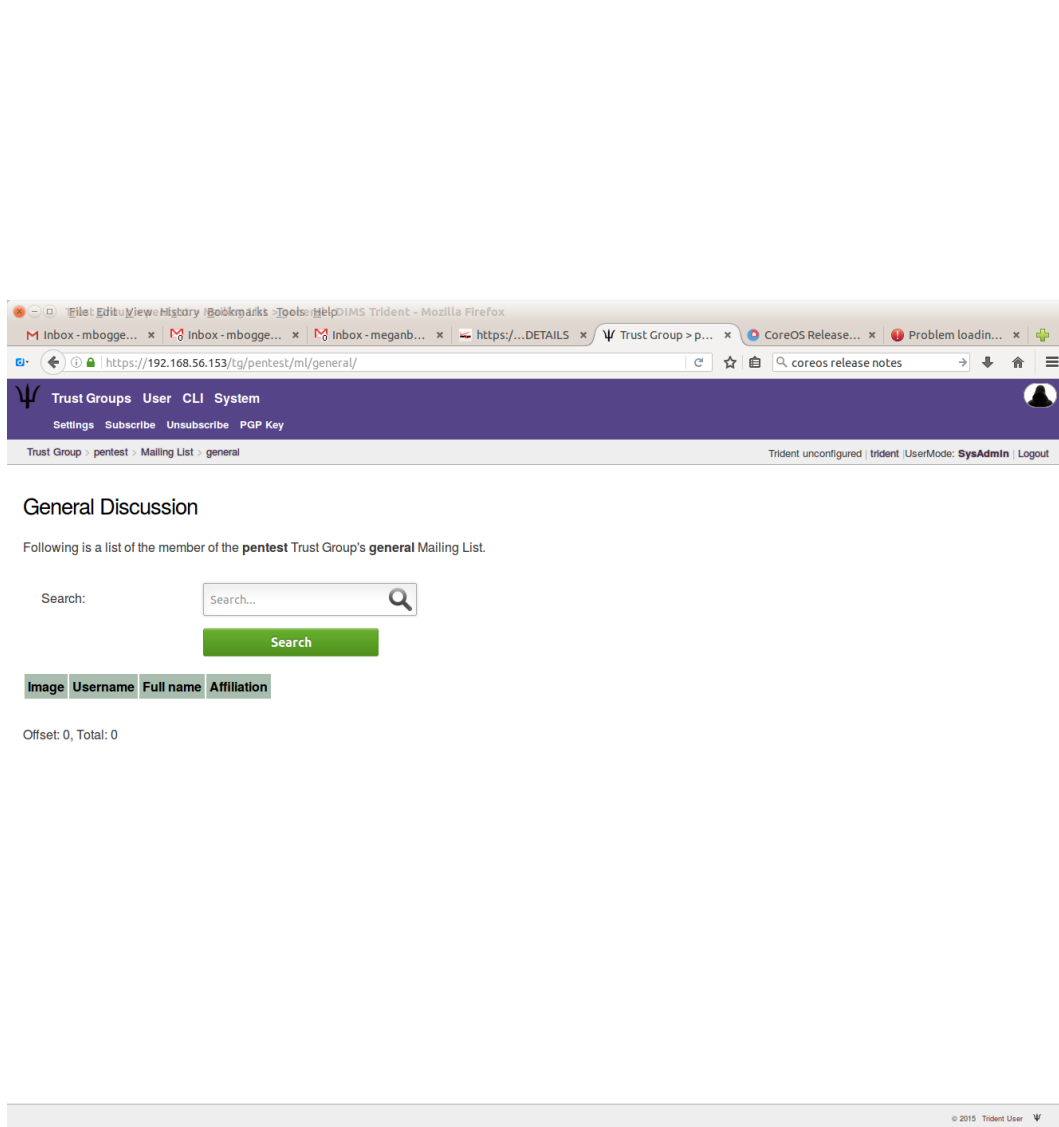


Fig. 7.34: No members on mailing list

The screenshot shows a web browser window with the URL `https://192.168.56.153/user/trident/tg/pentest/ml/general/subscribe`. The browser's address bar and tabs are visible at the top. The application's header is purple and contains the Trident logo, navigation links (Trust Groups, User, CLI, System), and a user profile icon. Below the header, a breadcrumb trail reads: `User > trident > pentest > Mailing List > general > Subscribe`. The main content area is titled 'Subscribe' and contains three input fields: 'Group Name' with the value 'pentest', 'List Name' with the value 'general', and 'User Name' with the value 'trident'. A green checkmark is visible next to the 'User Name' field. Below these fields is a green 'Subscribe' button. At the bottom right of the page, a small copyright notice reads: '© 2015 Trident User'.

Subscribe

Group Name:

List Name:

User Name:

© 2015 Trident User

Fig. 7.35: Add member to mailing list

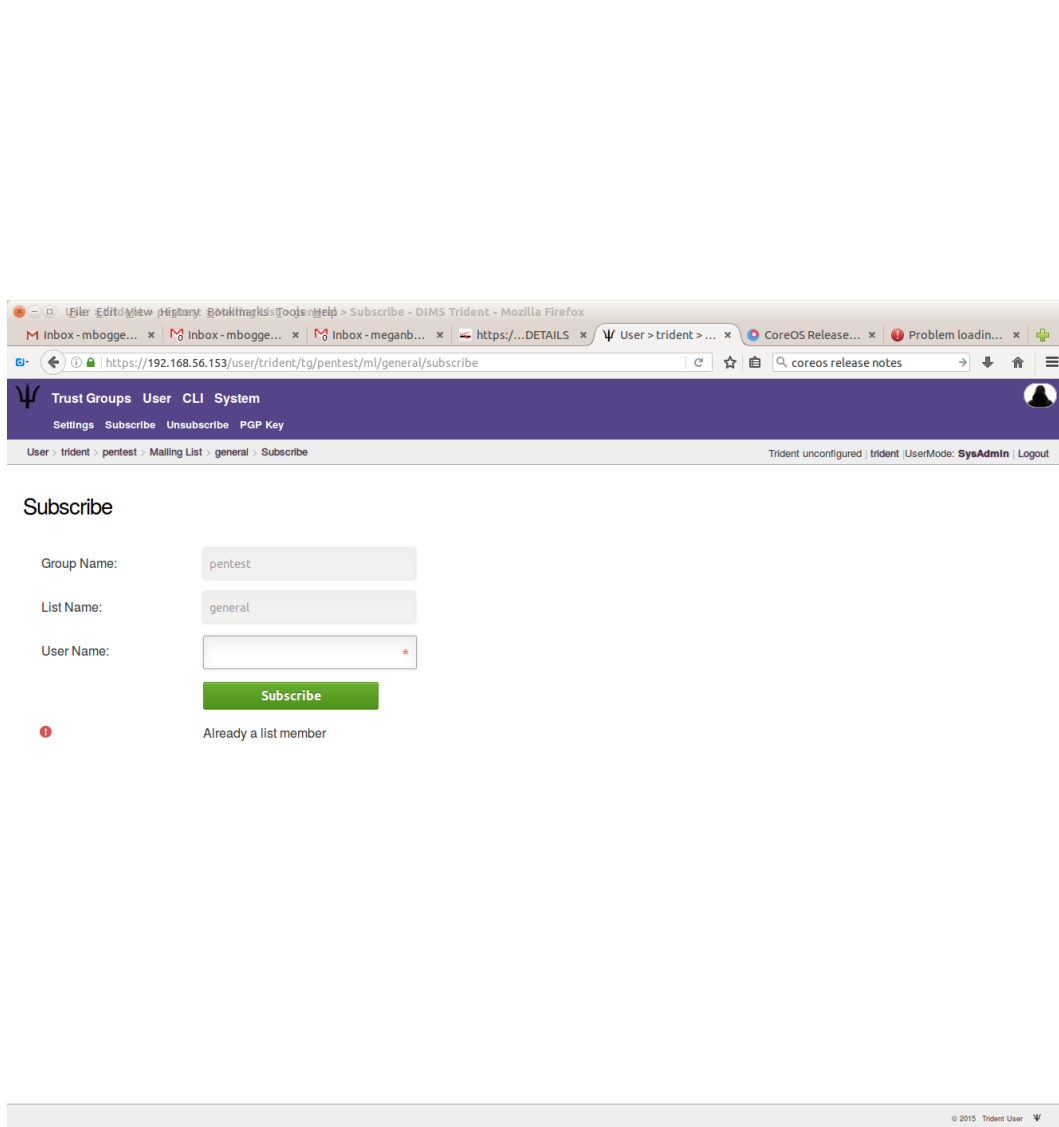


Fig. 7.36: Already member on mailing list



To see the users on a mailing list, choose a trust group and a mailing list, and you'll see a list of users and basic information about them:

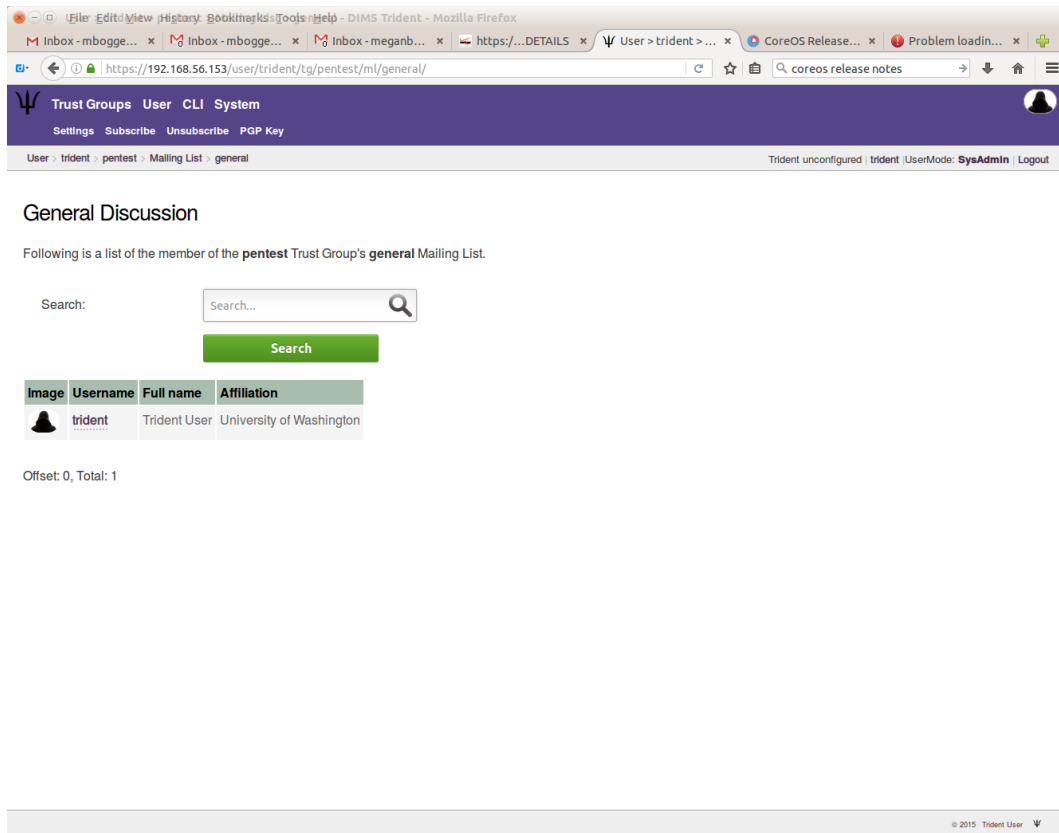


Fig. 7.37: List of users on mailing list

As a user, you can see which mailing lists you are subscribed to by particular trust groups:

To unsubscribe a user, choose a trust group and a mailing list, then click the “Unsubscribe” tab in the second row at the top of the page. Then give the username you’d like to unsubscribe from the given mailing list, and click “Unsubscribe”.

### 7.7.3 System information

To view the Trident System information, you must be a sysadmin. Click the “System” tab in the top row at the top of the page.

To view the audit log, click the “Audit Log” link in the index, or click the “Audit Log” tab in the second row at the top of the page.

To view the report, click the “Report” link in the index, or click the “Report” tab in the second row at the top of the page.

To change the system settings, click the “Settings” link in the index, or click the “Settings” tab in the second row at the top of the page.

Don’t forget to click the “Update Settings” button at the bottom of the page for the changes to take affect.

The screenshot shows a web browser window with the URL `https://192.168.56.153/user/trident/tg/pentest/ml/`. The page title is "Mailing List". Below the title, it says "The pentest Trust Group has the following Mailing Lists." followed by a table.

Shortname	Description	Address	Members	PGP	Subscription	Action
admin	TG Administration	pentest-admin@yellow.devops.local	1	PGP Key	Subscribed	<a href="#">Unsubscribe</a>
devops	devops	pentest-devops@yellow.devops.local	1	PGP Key	Subscribed	<a href="#">Unsubscribe</a>
general	General Discussion	pentest-general@yellow.devops.local	1	PGP Key	Subscribed	<a href="#">Unsubscribe</a>
vetting	Vetting and Vouching	pentest-vetting@yellow.devops.local	0	PGP Key	Unsubscribed	<a href="#">Subscribe</a>

At the bottom of the page, there is a footer that reads "© 2015 Trident User" followed by a small icon.

Fig. 7.38: Mailing list subscription status

The screenshot shows a web browser window with the URL `https://192.168.56.153/user/trident/tg/pentest/ml/general/unsubscribe/`. The page has a purple header with navigation links: Trust Groups, User, CLI, System, Settings, Subscribe, Unsubscribe, and PGP Key. The breadcrumb trail is `User > trident > pentest > Mailing List > general > Unsubscribe`. The main content area is titled 'Unsubscribe' and contains three input fields: 'Group Name' (filled with 'pentest'), 'List Name' (filled with 'general'), and 'User Name' (empty). A red error message 'The User Name' with a red circle icon is next to the 'User Name' field. Below the fields is a green 'Unsubscribe' button. The footer shows '© 2015 Trident User'.

Unsubscribe

Group Name:

List Name:

User Name:

**The User Name**

© 2015 Trident User

Fig. 7.39: Unsubscribe a user

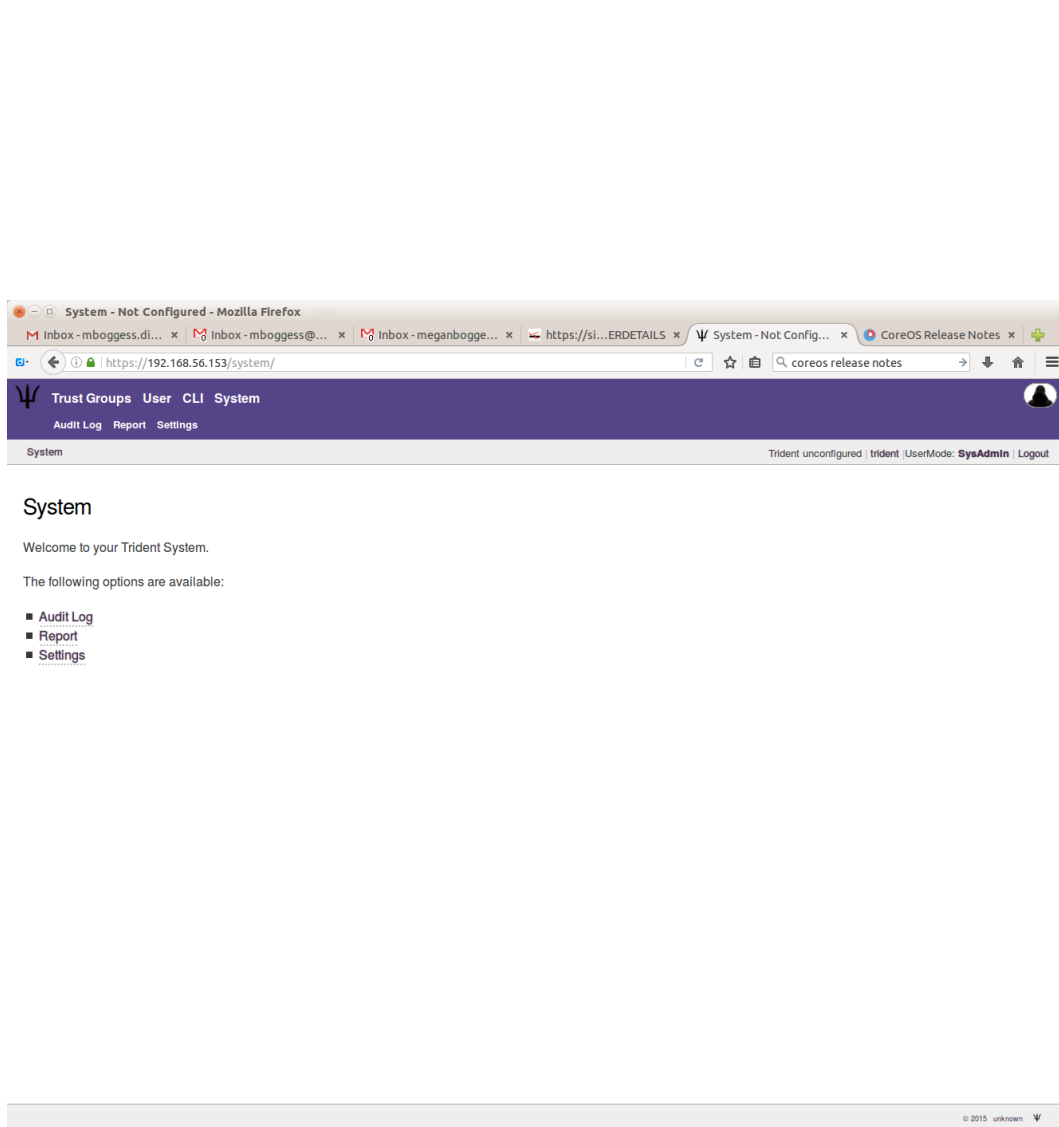



Fig. 7.40: Trident system information options

**Audit Log**

Search:  

**Search**

What	User	TrustGroup	Who	Entered	Remote
Confirmed email address 9f935580107976eff3e4d543ed7c964aa56c6d3689434897865056ec3af5585c	trident		trident	2016-09-09 06:46:58.274049 +0000 +0000	127.0.0.1
Send Verification Code to trident@yellow.devops.local	trident		trident	2016-09-09 06:44:43.79424 +0000 +0000	127.0.0.1
Added email address trident@yellow.devops.local for member trident	trident		trident	2016-09-09 06:44:26.25278 +0000 +0000	127.0.0.1
Update member WHERE ident = trident property descr from " to 'Trident User'	trident		trident	2016-09-09 06:43:45.427443 +0000 +0000	127.0.0.1
Update member WHERE ident = trident property tel_info from " to '222-555-1234'	trident		trident	2016-09-09 06:43:45.425556 +0000 +0000	127.0.0.1
Update member WHERE ident = trident property tz_info from " to 'Eastern Standard Time'	trident		trident	2016-09-09 06:43:45.424239 +0000 +0000	127.0.0.1
Update member WHERE ident = trident property affiliation from " to 'University of Washington'	trident		trident	2016-09-09 06:43:45.422423 +0000 +0000	127.0.0.1
Update member WHERE ident = trident property sysadmin from 'no' to 'yes'	trident		trident	2016-09-09 06:43:45.418176 +0000 +0000	127.0.0.1
Login: Updated Activity for trident				2016-09-09 06:15:22.432985 +0000 +0000	127.0.0.1
Login: Updated Activity for trident				2016-09-08 18:39:41.82553 +0000 +0000	127.0.0.1

Fig. 7.41: Trident system audit log

The screenshot shows a web browser window displaying the Trident system report. The browser's address bar shows the URL `https://192.168.56.153/system/report/`. The page has a purple header with navigation links: **Trust Groups**, **User**, **CLI**, and **System**. Below these are links for **Audit Log**, **Report** (which is active), and **Settings**. A user profile icon is visible in the top right corner. The main content area is titled **Report** and contains the following text:

```
Trident

Copyright: (C) 2015 The Trident Project
Portions (C) 2015 National Cyber Forensics Training Alliance
Website: https://trident.li
Version: unconfigured

Daemon started at 2016-09-09 06:14:32.735048046 -0700 PDT
Daemon running for 33m50.597545821s

Database schema version is 12
System schema version is 12

Database contents:
Mailing Lists: 0
Members: 1
Member Emails: 2
Member Vouches: 0
Trust Groups: 0
Wiki Pages: 0
Wiki Page Revisions: 0

Top 10 Largest Database Tables:
member_email      48 kB
member            48 kB
languages         48 kB
language_skill    48 kB
member_detail_types 48 kB
member_state      32 kB
wiki_page_rev     32 kB
audit_history     32 kB
second_factor_types 32 kB
file_rev          32 kB
```

The footer of the page shows the URL `https://192.168.56.153/system/report/` and a copyright notice: © 2015 unknown.

Fig. 7.42: Trident system report

System > Settings - Not Configured - Mozilla Firefox

https://192.168.56.153/system/settings/

Trust Groups User CLI System

Audit Log Report Settings

System > Settings

Trident unconfigured | trident (UserMode: SysAdmin) Logout

### Settings

System Name: Not Configured

Welcome Text:

Not Configured

Name of the Administrator(s): Trident User

Administrator email address: trident@yellow.devops.local

Copyright Years: 2015

Email Domain: yellow.devops.local

Public URL: https://yellow.devops.local

People Domain:

CLI Enabled: ☒

API Enabled: ☒

Auth/OpenID Enabled: ☒

Fig. 7.43: Trident system settings

Setting password rules is not recommended. Please use XKCD style passwords instead.

Enforce Rules: ☐

Minimal Password Length (suggested: 12):

Minimum amount of Letters:

Minimum amount of Uppercase characters:

Minimum amount of Lowercase characters:

Minimum amount of Numbers:

Minimum amount of Special characters:

IP Restrict SysAdmin:

Header Image:

Logo Image:

Unknown Person Image:

Show Trident Version in UI: ☒

Show Sysadmin E-mail to non-members: ☒

[Update Settings](#)

© 2015 unknown

Fig. 7.44: Update Trident system settings



## 7.7.4 Basic tcli use

To use tcli via the web app, you must be a sysadmin user. Click the “CLI” tab at the top of the page.

To get started, you can type the “help” command into the box, and you’ll get useful information on how to run tcli:

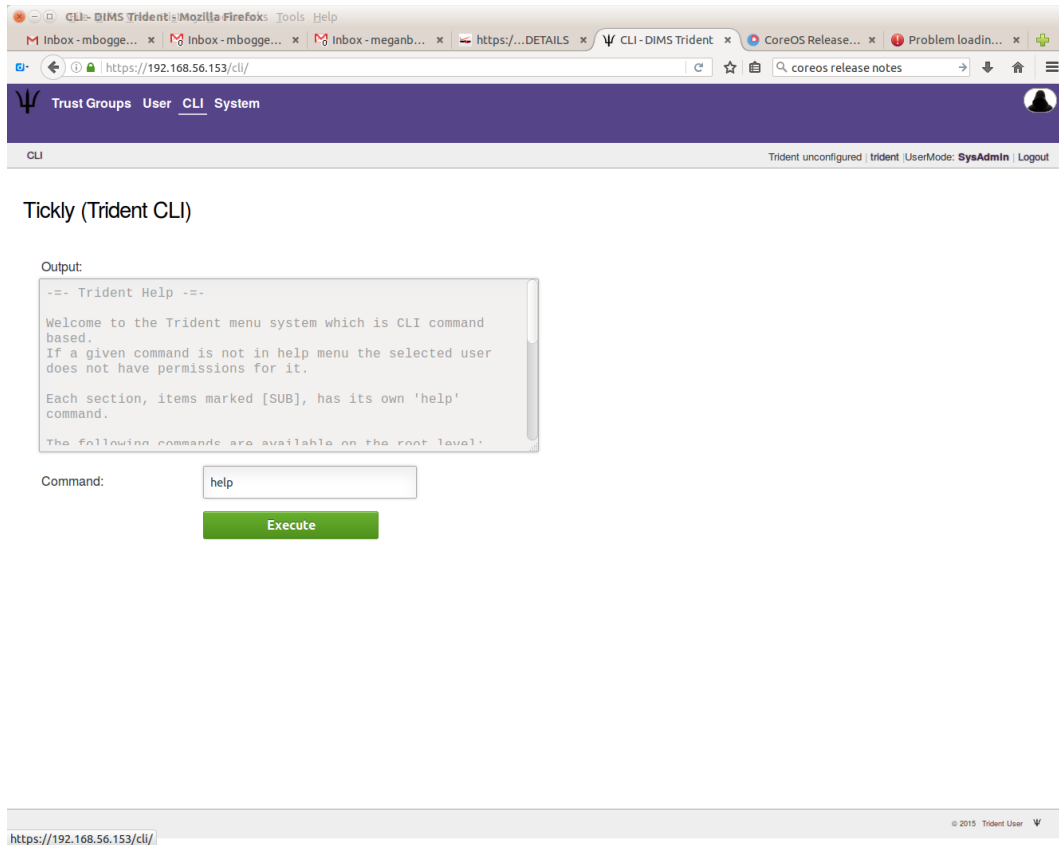


Fig. 7.45: Get tcli help

Anything you can run on the command line using tcli, you can run via the web app.

## 7.8 Upgrading configuration across Trident versions

One of the challenges with integrating open source applications into a continuous delivery or automated deployment environment has to do with managing customizations across changes in ongoing releases. From one version of a program to another, the contents of configuration files may change, they may be split into more configuration files, or merged from many into a smaller number, or their names and/or directory paths changed.

The first challenge with automating the configuration and installation of an open source application requires figuring out which files to put under Ansible control, and how to template those files so as to use variables in a way that supports customized deployments.

Each time a new release comes out, opportunities for things to break exist. Simply updating the version number and re-installing may work, but it may also break one or more things in the application. Some things that break will be easy to detect when starting a service, or running the application, but other problems may not be detected until long into the execution of some application or service that cause problems that are much harder to debug due to time between updating and encountering the problem.

To manage the upgrade process, one or more of the following tasks must be performed.

1. Differencing the contents of files under Ansible control to determine when configuration customization changes are necessary, or whether it is safe to just update and move on.
2. Differencing the contents of the distribution archive, or resulting installed files, to detect file name changes, new configuration files, etc. Knowing when the contents of default files have changed in the face of continuous deployment of files that are under Ansible control, takes some getting used to. Having a development environment in which a default installation can be performed, or using a basic “vanilla” virtual machine to hand-install the new package to look at the resulting files, may be necessary.
3. Choosing how to handle file name changes for possible backward-compatibility or multi-version support. This may involve complicated Ansible when conditionals, file names containing version numbers, or other mechanisms that prevent situations where a change results in a situation where the playbook only works with versions  $\leq N$  or  $\geq N$  in a mutually-exclusive way.

To see how these problems manifest themselves, and how to detect and handle them, let’s take a look at two different releases of the `trident` portal system. We will compare two releases, versions 1.3.8 and 1.4.2.

We start by extracting the contents of each release’s `deb` archive file into a directory where we can examine and/or compare the files.

```
$ cd /tmp
$ dpkg -x /vm/cache/sources/trident-server_1.3.8_amd64.deb trident_1.3.8
$ dpkg -x /vm/cache/sources/trident-server_1.4.2_amd64.deb trident_1.4.2
```

We now have two parallel directories in `/tmp`. Using the Unix `diff` program, we can see which files differ in content, or differ in existence (i.e., occur in one directory, but not the other).

Here is an example of changes to file contents:

```
$ diff -r trident_1.3.8/ trident_1.4.2/
diff -r trident_1.3.8/etc/init.d/trident trident_1.4.2/etc/init.d/trident
109a110,113
> rotate)
>     start-stop-daemon --stop --quiet --signal USR1 --exec ${DAEMON} --pidfile $
↪{PIDFILE} --name ${DNAME}
>     ;;
>
116c120
<     log_action_msg "Usage: ${SCRIPTNAME} {start|stop|restart|status}" || true
---
>     log_action_msg "Usage: ${SCRIPTNAME} {start|stop|restart|status|rotate}" || ↪
↪true
diff -r trident_1.3.8/etc/trident/nginx/trident-server.inc trident_1.4.2/etc/trident/
↪nginx/trident-server.inc
11,12d10
< #       include
< # ----->8
13a12,13
> #       ssl_certificate ...
> #       ...
15c15,17
<
---
> #       include /etc/trident/nginx/trident-server.inc
> # }
> # ----->8
23c25,28
<     location /css/ {
```

```

---
>         location ~ ^/(css|gfx|js)/ {
>             expires 7d;
>             root /usr/share;

```

Here are examples of file system changes, specifically those files in the webroot directory:

```

$ diff -r trident_1.3.8/ trident_1.4.2/ | grep '^Only' | grep '/webroot'
Only in trident_1.3.8/usr/share/trident/webroot/css: epiceditor
Only in trident_1.3.8/usr/share/trident/webroot/css: form.css
Only in trident_1.3.8/usr/share/trident/webroot/css: style.css
Only in trident_1.4.2/usr/share/trident/webroot/css: trident.css
Only in trident_1.3.8/usr/share/trident/webroot: favicon.ico
Only in trident_1.3.8/usr/share/trident/webroot/gfx: gm.jpg
Only in trident_1.3.8/usr/share/trident/webroot/gfx: info.png
Only in trident_1.3.8/usr/share/trident/webroot/gfx: invalid.png
Only in trident_1.3.8/usr/share/trident/webroot/gfx: logo.png
Only in trident_1.3.8/usr/share/trident/webroot/gfx: red_asterisk.png
Only in trident_1.3.8/usr/share/trident/webroot/gfx: search.png
Only in trident_1.3.8/usr/share/trident/webroot/gfx: unknown_person.jpg
Only in trident_1.3.8/usr/share/trident/webroot/gfx: valid.png
Only in trident_1.3.8/usr/share/trident/webroot/gfx: warning.png
Only in trident_1.3.8/usr/share/trident/webroot/gfx: xkcd_password_strength.png
Only in trident_1.3.8/usr/share/trident/webroot: js
Only in trident_1.3.8/usr/share/trident/webroot: robots-ok.txt
Only in trident_1.3.8/usr/share/trident/webroot: robots.txt

```

We can see that one file (`form.css`) was removed between release 1.3.8 and 1.4.2, while one file (`style.css`) was renamed, possibly including the now-absent `form.css` file, to a new file named `trident.css`. By looking at the contents of the `form.css` file, it is clear that `.styled_form` is one of the unique elements defined in this file. Looking at the contents of the same directory from both versions seems to support the hypothesis that this file was merged:

```

$ grep -r styled_form trident_1.3.8/usr/share/trident/webroot/css/
trident_1.3.8/usr/share/trident/webroot/css/style.css:form#wikiform.styled_form
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form .form_hint, .styled_
↪form .required
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form ul
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form li
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form h2
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form label
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input, .fakebutton
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form textarea
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input[type=number]
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input[type=radio]
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input[type=submit],
↪.fakebutton
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input, .styled_form
↪textarea, .fakebutton
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input:focus, .
↪styled_form textarea:focus, .fakebutton
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input:required, .
↪styled_form textarea:required
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form
↪input:required:valid, .styled_form textarea:required:valid
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input:focus:invalid,
↪.styled_form textarea:focus:invalid
trident_1.3.8/usr/share/trident/webroot/css/form.css:form.styled_form li.info label,
↪form.styled_form li.error label, form.styled_form li.okay label, form.styled_form
↪li.warning label, form.styl

```

```

ed_form li.required label
trident_1.3.8/usr/share/trident/webroot/css/form.css:form.styled_form li.info label
trident_1.3.8/usr/share/trident/webroot/css/form.css:form.styled_form li.error label
trident_1.3.8/usr/share/trident/webroot/css/form.css:form.styled_form li.okay label
trident_1.3.8/usr/share/trident/webroot/css/form.css:form.styled_form li.warning label
trident_1.3.8/usr/share/trident/webroot/css/form.css:form.styled_form li.required_
↪label
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input: hover + .form_
↪hint, .styled_form textarea: hover + .form_hint
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form_
↪input: required: valid + .form_hint, .styled_form textarea: required: valid + .form_
↪hint,
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form_
↪input: required: valid + .form_hint::before, .styled_form textarea: required: valid + .
↪form_hint::before
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input[type=submit],_
↪.fakebutton, .styled_button input
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input[type=submit],_
↪.fakebutton, .styled_button input
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form_
↪input[type=submit]:disabled
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input[type=submit].
↪deny
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form_
↪input[type=checkbox], input[type=radio]
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form_
↪input[type=checkbox]:checked, input[type=radio]:checked
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form_
↪input[type=checkbox]:disabled, input[type=radio]:disabled
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form_
↪input[type=checkbox]:checked:disabled, input[type=radio]:checked:disabled
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form_
↪input[type=checkbox]:after, input[type=radio]:after
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form_
↪input[type=checkbox]:disabled:after, input[type=radio]:disabled:after
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input[type="checkbox"
↪"]:checked:after, input[type="radio"]:checked:after
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form input[type="checkbox"
↪"]:focus
trident_1.3.8/usr/share/trident/webroot/css/form.css:.styled_form textarea.console

```

```

$ grep -r styled_form trident_1.4.2/usr/share/trident/webroot/css/
trident_1.4.2/usr/share/trident/webroot/css/trident.css:.login form.styled_form
trident_1.4.2/usr/share/trident/webroot/css/trident.css:.login .styled_form input
trident_1.4.2/usr/share/trident/webroot/css/trident.css:.login .styled_form_
↪input[type="submit"]

```

The problem now is how to support one CSS file named `style.css` for (at least) version 1.3.8, but a file named `trident.css` for (at least) version 1.4.2. There still remains the question, “When did this change occur, and how do we instruct Ansible which file to use?”

If, on the other hand, the file name has not changed but its contents vary significantly (e.g., one uses a variable named `file_root` and the other has changed to using a variable named `file_roots`), it becomes more complicated in managing a file with one name, but two different contents. This requires differentiating files by metadata (i.e., the name must include a version number or some other unique string), or the use of Jinja conditionals must be done. The latter mechanism of Jinja conditional inclusion, is a bit simpler and is easiest to manage in terms of file differencing as the mechanism for maintaining the contents of different versions of the file.

For example, here is how the difference between content in the file `trident.conf.j2` can be managed using Jinja conditionals:

```
# {{ ansible_managed }} [ansible-playbooks v{{ ansibleplaybooks_version }}]
#
#####
# Trident Configuration
#####
# Except for comment lines (anything starting with '#')
# this file is in the JSON format, thus mind the commas
# and quotes otherwise Trident can't properly use it.
#
# This file should only be readable by the Trident user
#####

{
{% if trident.version in [ '1.3.8' ] %}
    "file_root": "/usr/share/trident/",
{% endif %}
{% if trident.version in [ '1.4.2' ] %}
    # Where the dbschemas, webroot and templates are located
    "file_roots": [ "/usr/share/trident/", "/usr/share/pitchfork/" ],
{% endif %}

    # Where variable files are stored
    "var_root": "/var/lib/trident/",

    # TODO(dittrich): Try to get this to rsyslog for sorting, not separate logging
    # Log File location (logrotate rotates it)
    "logfile": "/var/log/trident/trident.log",

    # Crypto Keys for JWT (in directory relative to config dir)
    "jwt_key_prv": "jwt.prv",
    "jwt_key_pub": "jwt.pub",

{% if trident.version in [ '1.4.2' ] %}
    # Content Security Policy
    "csp": "default-src 'self'",

    # CSS: Cascading Style Sheets
    "css": [ "trident", "blockquote", "code", "crumbs", "diff", "form", "loader",
→ "messages", "search", "table", "wiki" ],

    # Javascript: global Javascript for every page
    # (Should actually always be empty)
    "javascript": [],

    # X-Forwarded-For Trusted IP list
    # CIDR prefixes from which we trust the XFF header
    "xff_trusted_cidr": [ "127.0.0.1/8" ],

    # Weak Password Dictionaries
    "pw_weakdicts": [ "10k_most_common.txt" ],
{% endif %}

{% if trident.version in [ '1.3.8' ] %}
    #####
    # PostgreSQL Database details
    #####
}
```

```
# PSQL local unix socket
# Uses PSQL peer authentication
# This works out of the box on Debian
#####
{% endif %}
{% if trident.version in [ '1.4.2' ] %}
#####
# PostgreSQL Database details
#####
# Requires configuration of pg_hba.conf!
#
# local unix socket (Debian):
#   "db_host": "/var/run/postgresql/",
#   "db_port": "5432",
#
# remote:
#   "db_host": "db.example.org",
#   "db_port": "5432",
#####
{% endif %}
```

## 7.9 Emails and other non-official documentation

- Email from Linda in response to Megan asking for any additional documentation.

```
To: Megan Boggess <mboggess@uw.edu>
From: Linda Parsons <linda.parsons@nextcentury.com>
Date: April 13, 2016
Subject: Trident emails and any other documentation
```

Hi Megan,

Yes, the new project is fun, and I hope things are going well for you too... There isn't any documentation on Trident other than what they provide at [trident.li](http://trident.li) and on their github pages - have Dave get you access to their repo. I relied on that documentation to do all the Docker and Ansible stuff.

The README in the `dims-dockerfiles` repo is the one that describes what I did. I may have comments in Ansible files as well that are descriptive - I don't have access to the code at the moment. I had the deployment done (or at least a working version to get you started) from build through Ansible deployment of two docker containers... but there is still work to be done and you will need to make the Ansible deployment fit with how you guys are doing things now.

the Postgresql container, and one to actually create the .deb build files to install Trident. The "build-trident" (or "trident-build" - not sure but it has "build" in the name of the directory) has a script that will pull the current source in our git repo (which in turn is from their trident repo - someone needs to keep that synchronized) and will create the .deb files and push them to our sources repo. That is so the actual Docker images can be created using them. I made a change to the file that controls the packaging so that it didn't require additional software like nginx, postfix, etc. - this is better for docker since we may not want all the services on all the containers that need this software. For example, to create the database on the postgresql container, you need trident

installed as well just so you can run their scripts. Anyway, the .deb packages don't force the user to install those services, but of course you will install them if you need them. So, I've got nginx and trident on the main trident image. The one thing that needs to be done is to also install and configure postfix on that image. I had been hoping we could use a separate docker container for that, but it would require changes to their source code. So you will need to modify that Dockerfile to install and configure postfix.

Maybe you could look through the dims-dockerfile stuff and the Ansible playbooks and then get back to me if you have questions. I could do a quick hangout to answer them. Also note there are two docker images for the postgresql container - one for the default one that is installed in a new environment, and one to install a copy of our ops-trust database. The second was used to get the trident system up and running on hub.prisem.washington.edu so we could use it and have the Dashboard be able to get data from that database. It was also necessary at the time since there apparently is a bug in a new install and the sysadmin can't create trust groups from within the UI (I have an issue in github for that but no one has responded). However, it cannot be used for new systems.

Another thing that needs to be worked out is how to do the certificates for the machine running the trident docker containers. Also, if you look at the Ansible playbooks, there are commands to start the containers in a development mode and in secure (production) mode. We are currently using development mode since we don't have the certs - production mode for the docker containers hasn't been tested.

I don't really have any emails to the trident guys... we had talked about emailing Vixie about the bug I mentioned above but I had to leave before that was done. I'm not sure why they haven't responded to the bug report on github. Anyway, what I knew was from reading through their docs many times and also from what I knew about Postgres databases in general, and then from actually building the system. So I think from reading the Dockerfiles and the Ansible playbooks you will get a good brain dump.

You should be able to build and deploy the trident system locally as long as you have a VM to install it on and a consul cluster running as well (need the consul stuff so the docker containers can talk to each other on the overlay network). Its better to use just the regular postgres-trident docker container for postgres (which creates a new database) - then you'll see the bug I mentioned. It is imperative that they fix that or let us know what we're doing wrong if anything (I posted a log to the github issue that shows the database errors that are being produced). It will also allow you to be able to test adding postfix to the mix.

Last I looked to they had not fixed the firewall issue that was preventing us from accessing the old ops-trust machines - not sure if that has been fixed yet.

Linda

- There is an Ansible role called `trident-docker-deploy` located in `$GIT/ansible-playbooks/roles`. This role creates a volume container to be paired with a DIMS postgres container (if it doesn't already exist), and a DIMS postgres container and DIMS Trident container.

The Dockerfiles and related files and scripts for these containers can be viewed at:

- Postgres: `$GIT/dims-dockerfiles/dockerfiles/postgres-trident`
- Trident: `$GIT/dims-dockerfiles/dockerfiles/trident`

- Additionally, Linda created a couple "helper" containers. One container updates `source.prisem.washington.edu` and another builds off the "fresh-install" DIMS postgres container to install a copy of the DIMS OPS-Trust database.

These can be viewed at:

- Build: `$GIT/dims-dockerfiles/dockerfiles/trident-build`
- Original Database: `$GIT/dims-dockerfiles/dockerfiles/postgres-trident-clone`



---

## AMQP and RabbitMQ

---

This chapter covers configuration and debugging of [RabbitMQ](#), a popular [AMQP](#) message bus service.

### 8.1 RabbitMQ use in DIMS

[AMQP](#) (specifically [RabbitMQ](#)) is discussed in Sections [DIMS architectural design](#) and [System Software Architecture of DIMS Architecture Design v 2.10.0](#), and the specifics of the server initially configured for use in DIMS is documented in Section `dimsasbuilt:rabbitmq` of `dimsasbuilt:dimsasbuilt`. Its use for processing logs within DIMS is discussed in Section `dimsparselogs:introtologparsing` of `dimsparselogs:parsinglogswithdims`.

**Attention:** While [RabbitMQ](#) is documented extensively on their web site, it is sometimes hard to interpret what it says. Another very useful resource is [Chapter 8: Administering RabbitMQ from the Web](#) from [RabbitMQ in Action: Distributed messaging for everyone](#), by Alvaro Videla and Jason J. W. Williams.

### 8.2 Basic Service Administration

[RabbitMQ](#) is started/stopped/restarted/queried for status just like any other Ubuntu service using the `service` command as root. Its configuration files and settings are found in `/etc/rabbitmq` and `/etc/default/rabbitmq-server`, and its log files in `/var/log/rabbitmq/`.

```
root@rabbitmq:~# cd /etc/rabbitmq
root@rabbitmq:/etc/rabbitmq# tree
.
+- enabled_plugins
+- rabbitmq.config
+- rabbitmq.conf.d
+- rabbitmq-env.conf

1 directory, 3 files
```

```
root@rabbitmq:/etc/rabbitmq# cat rabbitmq.config
[
  {kernel,
   [{inet_dist_listen_min, 45000},
    {inet_dist_listen_max, 45000}]
  },
  ].
```

```
root@rabbitmq:/var/log/rabbitmq# cat /etc/default/rabbitmq-server
ulimit -n 1024
```

---

**Note:** The `ulimit` setting here controls the number of open file handles a process can have. A server with lots of connections needs a higher limit than the default, hence this setting. See [\[rabbitmq-discuss\] Increasing the file descriptors limit](#) and [mozilla/opsec-puppet](#) and [Increase RabbitMQ file descriptor limit and memory watermark without restart](#).

```
root@b52:/etc/rabbitmq# rabbitmqctl status | grep -A 4 file_descriptors
{file_descriptors,
  [{total_limit, 924}, {total_used, 3}, {sockets_limit, 829}, {sockets_used, 1}],
 {processes, [{limit, 1048576}, {used, 200}]},
 {run_queue, 0},
 {uptime, 82858}}
```

```
root@rabbitmq:/etc/rabbitmq# cd /var/log/rabbitmq
root@rabbitmq:/var/log/rabbitmq# tree
.
+- rabbit@rabbitmq.log
+- rabbit@rabbitmq-sasl.log
+- shutdown_log
+- startup_log

0 directories, 4 files
```

## 8.3 Managing RabbitMQ

**RabbitMQ** can be administered in two ways: (1) manually, using the built-in web interface, or (2) using command line tools like `rabbitmqctl` and `rabbitmqadmin`.

To get access to the management interface, you must enable `rabbitmq_management` in the **RabbitMQ** configuration:

```
root@rabbitmq:/etc/rabbitmq# cat rabbitmq-env.conf
#RABBITMQ_NODE_IP_ADDRESS=10.142.29.170
RABBITMQ_NODE_PORT=5672
RABBITMQ_SERVER_START_ARGS="-rabbitmq_management listener [{port, 15672}]"

# Source other environment files (that include ONLY variable settings,
# not RabbitMQ configuration
for ENVFILE in `ls /etc/rabbitmq/rabbitmq.conf.d | sort -r`; do
  . /etc/rabbitmq/rabbitmq.conf.d/$ENVFILE
done
```

Once you do this, and restart the server, two things become available. The first is a web interface, and the second is access to a downloadable (from the [RabbitMQ](#) server itself) script named `rabbitmqadmin`.

### 8.3.1 Using the web interface

You can see the web management interface in Figure *RabbitMQ Mangement Interface Login Screen* and Figure *RabbitMQ Mangement Interface Home Screen*.

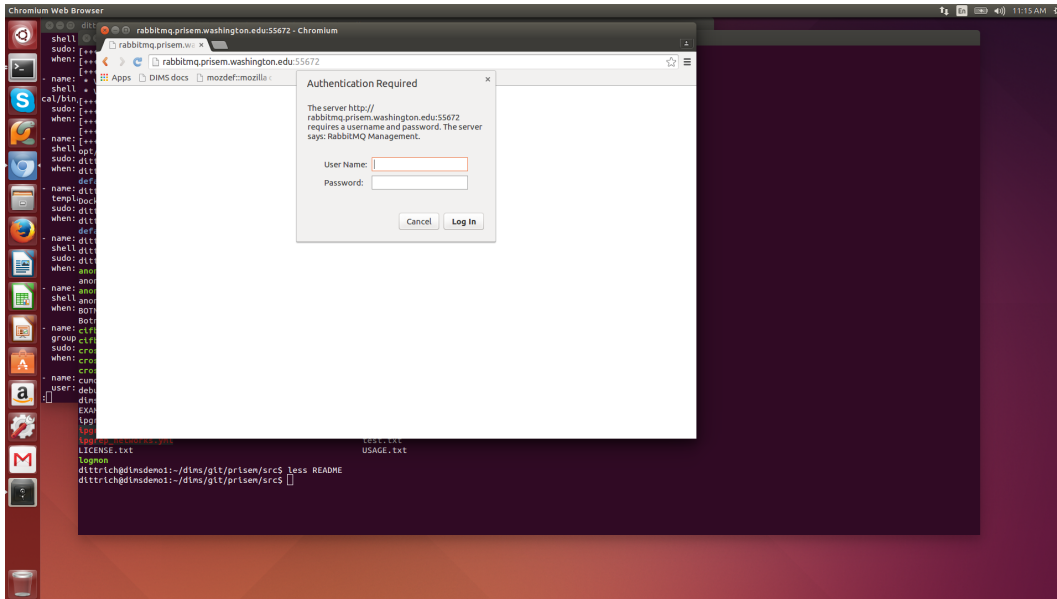


Fig. 8.1: RabbitMQ Mangement Interface Login Screen

### 8.3.2 Using the command line

The [RabbitMQ](#) service daemons are started like any other service on Ubuntu 14.04.

```
root@b52:~# service rabbitmq-server restart
* Restarting message broker rabbitmq-server
...done.
```

There are multiple ways with Linux to discover the listening port number. You can identify the process names with `ps` or `pstree` to map to output of `netstat`, use `lsof`, and the `epmd` command:

```
root@b52:~# pstree -p | less
init(1) +- ...
          | -lightdm(2599) +-Xorg(2648)
          | ...
          | -lightdm(3363) +-init(4946) +-at-spi-bus-laun(5140) +-dbus-
-> daemon(5144)
          | | | | -rabbitmq-server(19303) ---beam.
-> smp(19311) +-inet_gethost(19492) ---inet_gethos+
          | | | | |
-> | -{beam.smp}(19408)
          | | | | |
-> | -{beam.smp}(19409)
```

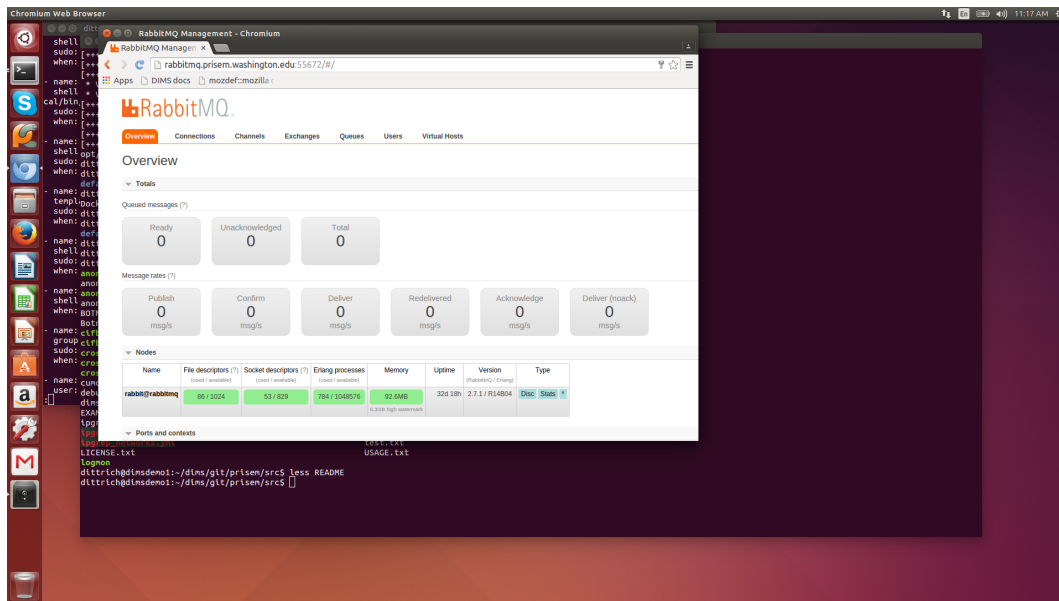


Fig. 8.2: RabbitMQ Mangement Interface Home Screen

```

↪ | ...
↪ | |
↪ | |-{beam.smp} (19451)
↪ | |
↪ | |-{beam.smp} (19452)
  | ...

```

```

root@b52:~# netstat -pan | grep beam
tcp        0      0 0.0.0.0:45000          0.0.0.0:*              LISTEN      19311/
↪beam.smp
tcp        0      0 127.0.0.1:51156        127.0.0.1:4369          ESTABLISHED 19311/
↪beam.smp
tcp6       0      0 :::5672                :::*                    LISTEN      19311/
↪beam.smp

```

```

root@b52:~# lsof -i | grep beam
beam.smp 19311      rabbitmq  8u  IPv4 27589259    0t0  TCP *:45000 (LISTEN)
beam.smp 19311      rabbitmq  9u  IPv4 27589261    0t0  TCP localhost:51156->
↪localhost:epmd (ESTABLISHED)
beam.smp 19311      rabbitmq 16u  IPv6 27580219    0t0  TCP *:amqp (LISTEN)

```

```

root@b52:~# epmd -names
epmd: up and running on port 4369 with data:
name rabbit at port 45000

```

There are two ways of getting the exact same information on the runtime status of [RabbitMQ](#). The first uses `rabbitmqctl` directly. The second uses `service rabbitmq-server status`. They are both shown here:

```

root@rabbitmq:/etc/rabbitmq# rabbitmqctl status
Status of node rabbit@rabbitmq ...
[{pid,8815},

```

```
{running_applications,
  [{rabbitmq_management,"RabbitMQ Management Console","0.0.0"},
   {rabbitmq_management_agent,"RabbitMQ Management Agent","0.0.0"},
   {amqp_client,"RabbitMQ AMQP Client","0.0.0"},
   {rabbit,"RabbitMQ","2.7.1"},
   {os_mon,"CPO CXC 138 46","2.2.7"},
   {sasldb,"SASL CXC 138 11","2.1.10"},
   {rabbitmq_mochiweb,"RabbitMQ Mochiweb Embedding","0.0.0"},
   {webmachine,"webmachine","1.7.0-rmq0.0.0-hg"},
   {mochiweb,"MochiMedia Web Server","1.3-rmq0.0.0-git"},
   {inets,"INETC CXC 138 49","5.7.1"},
   {mnesia,"MNESIA CXC 138 12","4.5"},
   {stdlib,"ERTS CXC 138 10","1.17.5"},
   {kernel,"ERTS CXC 138 10","2.14.5"}]},
{os,{unix,linux}},
{erlang_version,
  "Erlang R14B04 (erts-5.8.5) [source] [64-bit] [smp:16:16] [rq:16] [async-
→threads:30] [kernel-poll:true]\n"},
{memory,
  [{total,31080064},
   {processes,11445592},
   {processes_used,11433880},
   {system,19634472},
   {atom,1336577},
   {atom_used,1313624},
   {binary,117880},
   {code,14301212},
   {ets,1142776}]},
{vm_memory_high_watermark,0.39999999996434304},
{vm_memory_limit,6730807705}}
...done.
```

```
root@rabbitmq:/etc/rabbitmq# service rabbitmq-server status
Status of node rabbit@rabbitmq ...
[{pid,8815},
 {running_applications,
  [{rabbitmq_management,"RabbitMQ Management Console","0.0.0"},
   {rabbitmq_management_agent,"RabbitMQ Management Agent","0.0.0"},
   {amqp_client,"RabbitMQ AMQP Client","0.0.0"},
   {rabbit,"RabbitMQ","2.7.1"},
   {os_mon,"CPO CXC 138 46","2.2.7"},
   {sasldb,"SASL CXC 138 11","2.1.10"},
   {rabbitmq_mochiweb,"RabbitMQ Mochiweb Embedding","0.0.0"},
   {webmachine,"webmachine","1.7.0-rmq0.0.0-hg"},
   {mochiweb,"MochiMedia Web Server","1.3-rmq0.0.0-git"},
   {inets,"INETC CXC 138 49","5.7.1"},
   {mnesia,"MNESIA CXC 138 12","4.5"},
   {stdlib,"ERTS CXC 138 10","1.17.5"},
   {kernel,"ERTS CXC 138 10","2.14.5"}]},
 {os,{unix,linux}},
 {erlang_version,
  "Erlang R14B04 (erts-5.8.5) [source] [64-bit] [smp:16:16] [rq:16] [async-
→threads:30] [kernel-poll:true]\n"},
 {memory,
  [{total,31103832},
   {processes,11469280},
   {processes_used,11457568},
   {system,19634552},
```

```
{atom,1336577},
{atom_used,1313689},
{binary,117880},
{code,14301212},
{ets,1142776}}},
{vm_memory_high_watermark,0.399999999996434304},
{vm_memory_limit,6730807705}}
...done.
```

The following shows how to get a copy of the `rabbitmqadmin` script and make it executable from the command line.

```
root@rabbitmq:/etc/rabbitmq# wget http://localhost:55672/cli/rabbitmqadmin
root@rabbitmq:/etc/rabbitmq# chmod +x rabbitmqadmin
```

---

**Note:** These steps should be done immediately after initial [RabbitMQ](#) installation when creating Ansible playbooks, the script turned into a Jinja2 template, and installed into the `$PATH` for direct access from the command line (as opposed to being run with a relative path after changing directory into the `/etc/rabbitmq` directory as shown here).

---

The `rabbitmqadmin` script has a help option that provides information on how to use it.

```
root@rabbitmq:/etc/rabbitmq# ./rabbitmqadmin help subcommands
Usage
=====
  rabbitmqadmin [options] subcommand

  where subcommand is one of:

Display
=====

  list users [<column>...]
  list vhosts [<column>...]
  list connections [<column>...]
  list exchanges [<column>...]
  list bindings [<column>...]
  list permissions [<column>...]
  list channels [<column>...]
  list parameters [<column>...]
  list queues [<column>...]
  list policies [<column>...]
  list nodes [<column>...]
  show overview [<column>...]

Object Manipulation
=====

  declare queue name=... [node=... auto_delete=... durable=... arguments=...]
  declare vhost name=... [tracing=...]
  declare user name=... password=... tags=...
  declare exchange name=... type=... [auto_delete=... internal=... durable=...
↪arguments=...]
  declare policy name=... pattern=... definition=... [priority=... apply-to=...]
  declare parameter component=... name=... value=...
  declare permission vhost=... user=... configure=... write=... read=...
```

```

declare binding source=... destination=... [arguments=... routing_key=...
↪destination_type=...]
delete queue name=...
delete vhost name=...
delete user name=...
delete exchange name=...
delete policy name=...
delete parameter component=... name=...
delete permission vhost=... user=...
delete binding source=... destination_type=... destination=... properties_key=...
close connection name=...
purge queue name=...

Broker Definitions
=====

export <file>
import <file>

Publishing and Consuming
=====

publish routing_key=... [payload=... payload_encoding=... exchange=...]
get queue=... [count=... requeue=... payload_file=... encoding=...]

* If payload is not specified on publish, standard input is used

* If payload_file is not specified on get, the payload will be shown on
  standard output along with the message metadata

* If payload_file is specified on get, count must not be set

```

Here rabbitmqadmin is used to get a list of the currently defined exchanges:

```

root@rabbitmq:/etc/rabbitmq# ./rabbitmqadmin list exchanges
+-----+-----+-----+-----+-----+-----+
| vhost |      name      | type  | auto_delete | durable | internal |
+-----+-----+-----+-----+-----+-----+
| /      |                  | direct | False       | True    | False    |
| /      | amq.direct      | direct | False       | True    | False    |
| /      | amq.fanout       | fanout | False       | True    | False    |
| /      | amq.headers      | headers | False       | True    | False    |
| /      | amq.match        | headers | False       | True    | False    |
| /      | amq.rabbitmq.log | topic  | False       | True    | False    |
| /      | amq.rabbitmq.trace | topic  | False       | True    | False    |
| /      | amq.topic        | topic  | False       | True    | False    |
| /      | devops           | fanout | False       | True    | False    |
| /      | log_task         | direct | False       | True    | False    |
| /      | logs             | fanout | False       | False   | False    |
+-----+-----+-----+-----+-----+-----+

```

We can now define a new fanout exchange where we can direct log messages for later processing using rabbitmqadmin, rather than the web interface:

```

root@rabbitmq:/etc/rabbitmq# ./rabbitmqadmin declare exchange name=health type=fanout
↪auto_delete=false durable=true internal=false
exchange declared
root@rabbitmq:/etc/rabbitmq# ./rabbitmqadmin list exchanges

```

vhost	name	type	auto_delete	durable	internal
/		direct	False	True	False
/	amq.direct	direct	False	True	False
/	amq.fanout	fanout	False	True	False
/	amq.headers	headers	False	True	False
/	amq.match	headers	False	True	False
/	amq.rabbitmq.log	topic	False	True	False
/	amq.rabbitmq.trace	topic	False	True	False
/	amq.topic	topic	False	True	False
/	devops	fanout	False	True	False
/	health	fanout	False	True	False
/	log_task	direct	False	True	False
/	logs	fanout	False	False	False

After creating all of the broker objects we wish to have in the default server (using either the web interface and/or rabbitmqadmin) you can export a JSON file that can be put under Ansible control for later import into a newly instantiated RabbitMQ server. (See *Loading rabbitmq config at startup*.)

**Caution:** There are passwords in this output (which are redacted here). Keep this file secure and *do not put it in a public source repository* without encryption or templating (e.g., with Jinja2).

```

root@rabbitmq:/etc/rabbitmq# ./rabbitmqadmin export broker-objects.json
Exported definitions for localhost to "broker-objects.json"
root@rabbitmq:/etc/rabbitmq# python -m json.tool broker-objects.json
{
  "bindings": [
    {
      "arguments": {},
      "destination": "log_task",
      "destination_type": "queue",
      "routing_key": "log_task",
      "source": "log_task",
      "vhost": "/"
    },
    {
      "arguments": {},
      "destination": "log_test_queue",
      "destination_type": "queue",
      "routing_key": "",
      "source": "test_exchange",
      "vhost": "/"
    },
    {
      "arguments": {},
      "destination": "taskqueue",
      "destination_type": "queue",
      "routing_key": "",
      "source": "test_exchange",
      "vhost": "/"
    },
    {
      "arguments": {},

```



```

        "destination": "test_exchange",
        "destination_type": "queue",
        "routing_key": "test_exchange",
        "source": "test_exchange",
        "vhost": "/"
    }
},
"exchanges": [
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "internal": false,
        "name": "test_exchange",
        "type": "direct",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "internal": false,
        "name": "devops",
        "type": "fanout",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "internal": false,
        "name": "test",
        "type": "fanout",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "internal": false,
        "name": "health",
        "type": "fanout",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": false,
        "internal": false,
        "name": "logs",
        "type": "fanout",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "internal": false,

```

```

        "name": "log_task",
        "type": "direct",
        "vhost": "/"
    }
},
"permissions": [
    {
        "configure": ".*",
        "read": ".*",
        "user": "rpc_user",
        "vhost": "/",
        "write": ".*"
    },
    {
        "configure": ".*",
        "read": ".*",
        "user": "logmatrix",
        "vhost": "/",
        "write": ".*"
    },
    {
        "configure": ".*",
        "read": ".*",
        "user": "hutchman",
        "vhost": "/",
        "write": ".*"
    }
],
"queues": [
    {
        "arguments": {},
        "auto_delete": false,
        "durable": false,
        "name": "crosscor_test_0.5.5",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "name": "taskqueue",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": false,
        "name": "cifbulk_v1_0.5.5",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "name": "test_exchange",
        "vhost": "/"
    },
    {

```

```

        "arguments": {},
        "auto_delete": false,
        "durable": false,
        "name": "anon_0.5.5",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "name": "log_task",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": false,
        "name": "cifbulk_v1_test_0.5.5",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": false,
        "name": "crosscor_0.5.5",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "name": "log_queue_test",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": true,
        "name": "log_test_queue",
        "vhost": "/"
    },
    {
        "arguments": {},
        "auto_delete": false,
        "durable": false,
        "name": "anon_test_0.5.5",
        "vhost": "/"
    }
],
"rabbit_version": "2.7.1",
"users": [
    {
        "name": "hutchman",
        "password_hash": "REDACTED",
        "tags": "administrator"
    },
    {
        "name": "logmatrix",

```

```
        "password_hash": "REDACTED",
        "tags": "administrator"
    },
    {
        "name": "rpc_user",
        "password_hash": "REDACTED",
        "tags": ""
    }
],
"vhosts": [
    {
        "name": "/"
    }
]
}
```

## 8.4 Management with Ansible playbooks

---

RaspberryPi and Docker

---

This chapter covers installing and configuring Docker on a [RaspberryPi 2](#) for prototyping Docker container microservices and supporting DIMS deployment using PXE boot support.

## 9.1 Installing HypriotOS w/Docker

**Note:** The Raspberry Pi uses a micro SD card to hold the operating system it will boot. To run *any* operating system, you must first create a bootable micro SD card. You can find many pages with instructions on [How to Flash an SD Card for Raspberry Pi](#). This section uses one such set of instructions for a ARM-based Linux distribution with Docker installed on it.

The folks at Hypriot have instructions for [Getting started with Docker on your Raspberry Pi](#), that step through the process of install one of their pre-configured *SD card images* to your Raspberry Pi. Mac users can take advantage of a command-line script to flash the SD card image on GitHub in the repo [hypriot/flash](#).

```
[dimsenv] dittrich@27b:~/git () $ git clone https://github.com/hypriot/flash.git
Cloning into 'flash'...
remote: Counting objects: 100, done.
remote: Total 100 (delta 0), reused 0 (delta 0), pack-reused 100
Receiving objects: 100% (100/100), 25.54 KiB | 0 bytes/s, done.
Resolving deltas: 100% (42/42), done.
Checking connectivity... done.
[dimsenv] dittrich@27b:~/git () $ git checkout -b dims
[dimsenv] dittrich@27b:~/git (dims) $ cd flash
[dimsenv] dittrich@27b:~/git/flash (dims) $ ls
AUTHORS      Darwin      LICENSE     Linux      README.md
[dimsenv] dittrich@27b:~/git/flash (dims) $ tree
.
+- AUTHORS
+- Darwin
|   +- flash
```

```

+- LICENSE
+- Linux
|   +- flash
+- README.md

2 directories, 5 files
[dimsenv] dittrich@27b:~/git/flash (dims) $ cd Darwin
[dimsenv] dittrich@27b:~/git/flash/Darwin (dims) $ brew install pv
==> Downloading https://homebrew.bintray.com/bottles/pv-1.6.0.yosemite.bottle.1.tar.gz
brew install awscli/usr/bin/curl -fLA Homebrew 0.9.5 (Ruby 2.0.0-481; OS X 10.10.5)
↳ https://homebrew.bintray.com/bottles/pv-1.6.0.yosemite.bottle.1.tar.gz -C 0 -o /
↳ Library/Caches/Homebrew/pv-1.6.0.yosemite.bottle.1.tar.gz.incomplete
% Total      % Received % Xferd  Average Speed   Time    Time       Time  Current
             Dload  Upload   Total   Spent    Left     Speed
100 34692  100 34692    0     0  10668      0  0:00:03  0:00:03 --:--:-- 10671
==> Verifying pv-1.6.0.yosemite.bottle.1.tar.gz checksum
==> Pouring pv-1.6.0.yosemite.bottle.1.tar.gz
tar xf /Library/Caches/Homebrew/pv-1.6.0.yosemite.bottle.1.tar.gz
==> Finishing up
ln -s ../Cellar/pv/1.6.0/bin/pv pv
ln -s ../../../Cellar/pv/1.6.0/share/man/man1/pv.1 pv.1
==> Summary
/usr/local/Cellar/pv/1.6.0: 4 files, 84K

```

If you need to enable wireless, create an `occidentalis.txt` file with the SSID and password for connecting to your wireless access point. PXE boot over ethernet will use the wired interface, but you may want to enable wireless for remote management of the Raspberry Pi.

```

[dimsenv] dittrich@27b:~/git/flash/Darwin (dims) $ vi occidentalis.txt
# hostname for your Hypriot Raspberry Pi:
hostname=dims-rpi

# basic wireless networking options:
wifi_ssid=REDACTED
wifi_password=REDACTED

```

**Note:** The instructions below assume that you have created an `occidentalis.txt` file. Remove that from the command line if you did not create one.

Insert a micro SD card into one of the memory slots and run the flash script, referencing the most recent version of the `hypriot-rpi` image file from the [SD card images](#) page.

```

[dimsenv] dittrich@27b:~/git/flash/Darwin (dims*) $ ./flash -c occidentalis.txt http://
↳ /downloads.hypriot.com/hypriot-rpi-20151004-132414.img.zip

Downloading http://downloads.hypriot.com/hypriot-rpi-20151004-132414.img.zip ...
% Total      % Received % Xferd  Average Speed   Time    Time       Time  Current
             Dload  Upload   Total   Spent    Left     Speed
100 449M  100 449M    0     0  3025k      0  0:02:32  0:02:32 --:--:-- 118k
Uncompressing /tmp/image.img.zip ...
Archive:  /tmp/image.img.zip
  inflating: /tmp/hypriot-rpi-20151004-132414.img
Use /tmp/hypriot-rpi-20151004-132414.img
Filesystem      512-blocks      Used Available Capacity    iused    ifree %iused  Mounted
↳ on

```

```

/dev/disk1      974749472 905546856 68690616      93% 113257355 8586327 93% /
devfs           686      686      0 100%      1188      0 100% /dev
map -hosts       0        0      0 100%       0      0 100% /net
map auto_home    0        0      0 100%       0      0 100% /home
/dev/disk2s2     15328216 5154552 10173664      34% 644317 1271708 34% /Users/
↳dittrich/dims/git
/dev/disk3s1     130780 47284 83496 37%      512      0 100% /
↳Volumes/NO NAME

Is /dev/disk3s1 correct? y
Unmounting disk3 ...
Unmount of all volumes on disk3 was successful
Unmount of all volumes on disk3 was successful
Flashing /tmp/hypriot-rpi-20151004-132414.img to disk3 ...
Password:
 1.4GiB 0:03:45 [6.34MiB/s]
↳[=====
↳] 100%

dd: /dev/rdisk3: Invalid argument
0+22889 records in
0+22888 records out
1499987968 bytes transferred in 225.533768 secs (6650835 bytes/sec)
Copying occidental.is.txt to /Volumes/NO NAME/occidental.is.txt ...
Unmounting and ejecting disk3 ...
Unmount of all volumes on disk3 was successful
Unmount of all volumes on disk3 was successful
Disk /dev/disk3 ejected
Finished.

```

Insert the SD card into the Raspberry Pi and power it on. It will use DHCP to get an IP address, so these instructions require that you find the system on the network. (In this case, the IP address was identified to be 192.168.0.104.)

Copy your SSH key to the Raspberry Pi for remote SSH access.

```

[dimsenv] dittrich@27b:~/git/flash/Darwin (dims*) $ ssh-copy-id -i ~/.ssh/dims_
↳dittrich_rsa.pub root@192.168.0.104

/opt/local/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
↳out any that are already installed
/opt/local/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
↳prompted now it is to install the new keys
root@192.168.0.104's password:

Number of key(s) added:      1

Now try logging into the machine, with:  "ssh 'root@192.168.0.104'"
and check to make sure that only the key(s) you wanted were added.

```

Since this is the first boot, now is a good time to update the operating system.

```

[dimsenv] dittrich@27b:~ () $ slogin -i ~/.ssh/dims_dittrich_rsa root@192.168.0.104
Linux dims-rpi 3.18.11-hypriotos-v7+ #2 SMP PREEMPT Sun Apr 12 16:34:20 UTC 2015
↳armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 31 06:24:35 2015 from 192.168.0.5
HypriotOS: root@dims-rpi in ~
$ apt-get update
Get:1 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Get:2 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
...
HypriotOS: root@dims-rpi in ~
$ aptitude safe-upgrade
The following packages will be upgraded:
  bind9-host curl dpkg libbind9-80 libcurl3 libcurl3-gnutls libdns88 libexpat1
↳ libisc84 libisccc80 libiscfg82 liblwres80 libsqlite3-0 libssl1.0.0 openssl sudo
↳ tzdata wpasupplicant
18 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 8,700 kB of archives. After unpacking 957 kB will be freed.
Do you want to continue? [Y/n/?] y
Get: 1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main dpkg armhf 1.16.
↳ 16+rpil [2,599 kB]
...
Setting up sudo (1.8.5p2-1+nmu3) ...
Setting up wpasupplicant (1.0-3+deb7u2) ...

Current status: 0 updates [-18].
```

If you are not in central Europe, you may want to also set the time zone.

```
HypriotOS: root@dims-rpi in ~
$ dpkg-reconfigure tzdata

Current default time zone: 'US/Pacific-New'
Local time is now:      Fri Oct 30 22:29:49 PDT 2015.
Universal Time is now:  Sat Oct 31 05:29:49 UTC 2015.
```

## 9.2 Installing a Persistent Docker Container

The Hypriot web page shows how to download and run a Docker container to serve a web page to prove the Raspberry Pi is online and working. As soon as you reboot the Raspberry Pi, the container will stop and you will have to log in and manually re-run it.

The container can be made persistent across reboots using `supervisord`, which is demonstrated in this section.

### 9.2.1 Install and Test the Container

Start by running the Docker container as described in [Getting started with Docker on your Raspberry Pi](#), to make sure it can run standalone and that you can connect to it over the network.

```
HypriotOS: root@dims-rpi in ~
$ docker run -d -p 80:80 hypriot/rpi-busybox-httpd
Unable to find image 'hypriot/rpi-busybox-httpd:latest' locally
latest: Pulling from hypriot/rpi-busybox-httpd
78666be98989: Pull complete
65c121b6f9de: Pull complete
```



```
4674ad400a98: Pull complete
d0cb6fa4fa79: Pull complete
Digest: sha256:c00342f952d97628bf5dda457d3b409c37df687c859df82b9424f61264f54cd1
Status: Downloaded newer image for hypriot/rpi-busybox-httpd:latest
e0131b218070ef8a0c82a8bde07b749a4d3e3b4fb7ca15930e3148c1252dee1d
```

```
HypriotOS: root@dims-rpi in ~
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
e0131b218070	hypriot/rpi-busybox-httpd:latest	"/bin/busybox httpd	7
seconds ago	Up 6 seconds	0.0.0.0:80->80/tcp	admiring_heisenberg

Validate the port (in this case, tcp6/80 is bound) are now actively listening.

```
HypriotOS: root@dims-rpi in ~
$ netstat -pan
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	2105/ssh
tcp	0	184	192.168.0.104:22	192.168.0.5:61271	ESTABLISHED	1518/ssh: root [priv
tcp6	0	0	:::80	:::*	LISTEN	11430/docker-proxy
tcp6	0	0	:::22	:::*	LISTEN	763/ssh
udp	0	0	0.0.0.0:7712	0.0.0.0:*		1951/dhclient
udp	0	0	0.0.0.0:68	0.0.0.0:*		1951/dhclient
udp	0	0	172.17.42.1:123	0.0.0.0:*		1717/ntpd
udp	0	0	192.168.0.104:123	0.0.0.0:*		1717/ntpd
udp	0	0	127.0.0.1:123	0.0.0.0:*		1717/ntpd
udp	0	0	0.0.0.0:123	0.0.0.0:*		1717/ntpd
udp	0	0	0.0.0.0:5353	0.0.0.0:*		1822/avahi-daemon:
udp	0	0	0.0.0.0:42246	0.0.0.0:*		1822/avahi-daemon:
...						

If you can connect to the server, you will see Hypriot's page:

## 9.2.2 Install and Test Supervisor

Now install the supervisor package.

```
HypriotOS: root@dims-rpi in ~
$ apt-get install supervisor
Reading package lists... Done
Building dependency tree
```

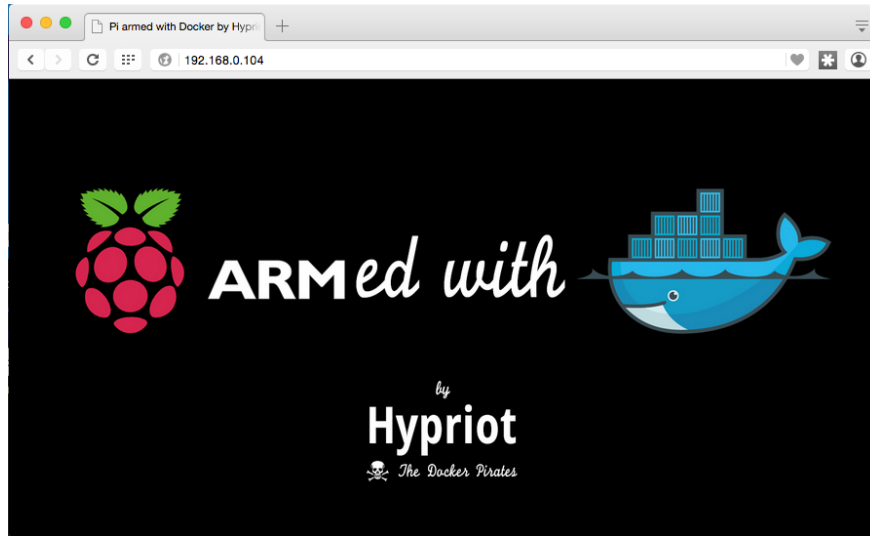


Fig. 9.1: Hypriot test page

```

Reading state information... Done
The following extra packages will be installed:
  file libmagic1 mime-support python python-medusa python-meld3 python-minimal python-
  ↳ pkg-resources python-support python2.7 python2.7-minimal
Suggested packages:
  python-doc python-tk python-medusa-doc python-distribute python-distribute-doc
  ↳ python2.7-doc binfmt-support
The following NEW packages will be installed:
  file libmagic1 mime-support python python-medusa python-meld3 python-minimal python-
  ↳ pkg-resources python-support python2.7 python2.7-minimal supervisor
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 5,273 kB of archives.
After this operation, 19.2 MB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main libmagic1 armhf 5.11-
  ↳ 2+deb7u8 [201 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian/ wheezy/main file armhf 5.11-
  ↳ 2+deb7u8 [53.1 kB]
...
Setting up python-meld3 (0.6.5-3.1) ...
Setting up supervisor (3.0a8-1.1+deb7u1) ...
Starting supervisor: supervisord.
Processing triggers for python-support ...

```

Verify that it is running.

```

HypriotOS: root@dims-rpi in ~
$ service supervisor status
supervisord is running

```

We will now configure the persistence mechanism (i.e., `supervisord` configuration file) that will employ an abstraction mechanism in the form of a script to actually start the container. Here is what the run script looks like:

```

HypriotOS: root@dims-rpi in ~
$ cat rpi-busybox-httpd.run
#!/bin/bash

```

```
NAME=${1:-rpi-busybox-httpd}

# Remove any stopped container with the specified name.
/usr/bin/docker rm $NAME 2>/dev/null

# Run the container with the specified name.
/usr/bin/docker run \
    -a stdout \
    --rm \
    --name $NAME \
    -p 80:80 \
    hypriot/rpi-busybox-httpd
```

The run script is then referenced in the supervisord configuration file that is placed into the `conf.d` directory along with any other configuration files that supervisord will manage. The command line is very simple.

```
HypriotOS: root@dims-rpi in ~
$ cat /etc/supervisor/conf.d/rpi-busybox-httpd.conf
[program:rpi-busybox-httpd]
command=/root/rpi-busybox-httpd.run "%(program_name)s_%(process_num)02d"
autostart=true
autorestart=true
startretries=100
numprocs=1
process_name=%(program_name)s_%(process_num)02d
user=root
env=HOSTNAME="dims-rpi",SHELL="/bin/bash",USER="root",PATH="/usr/local/sbin:/usr/
↳ local/bin:/usr/sbin:/usr/bin:/sbin:/bin",LANG="en_US"
```

Make sure that supervisord can restart with this configuration file in place, and that port `tcp6/80` is still listening.

```
HypriotOS: root@dims-rpi in ~
$ service supervisor restart
Restarting supervisor: supervisord.
HypriotOS: root@dims-rpi in ~
$ netstat -pan --inet
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address          State           PID/
↳ Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN          2105/
↳ sshd
tcp        0      184 192.168.0.104:22         192.168.0.5:61271        ESTABLISHED    2116/0
udp        0      0 0.0.0.0:7712            0.0.0.0:*                1951/
↳ dhclient
udp        0      0 0.0.0.0:68              0.0.0.0:*                1951/
↳ dhclient
udp        0      0 172.17.42.1:123         0.0.0.0:*                1717/
↳ ntpd
udp        0      0 192.168.0.104:123       0.0.0.0:*                1717/
↳ ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                1717/
↳ ntpd
udp        0      0 0.0.0.0:123             0.0.0.0:*                1717/
↳ ntpd
udp        0      0 0.0.0.0:5353            0.0.0.0:*                1822/
↳ avahi-daemon:
udp        0      0 0.0.0.0:42246           0.0.0.0:*                1822/
↳ avahi-daemon:
```

```

HypriotOS: root@dims-rpi in ~
$ docker ps
CONTAINER ID          IMAGE                                COMMAND                                CREATED
↪                      STATUS                            PORTS                                NAMES
53d51a7f1c17          hypriot/rpi-busybox-httpd:latest    "/bin/busybox httpd"                12
↪seconds ago          Up 11 seconds                      0.0.0.0:80->80/tcp                  rpi-busybox-httpd_00

```

Test the server remotely by loading the URL <http://192.168.0.105> from a browser on the same subnet to confirm the Hypriot test page seen in Figure [Hypriot test page](#) is still being served.

Now, reboot the Raspberry Pi to make sure that `supervisord` starts the container at boot time.

```

HypriotOS: root@dims-rpi in ~
$ /sbin/shutdown -r now

Broadcast message from root@dims-rpi (pts/0) (Sat Oct 31 18:06:08 2015):
The system is going down for reboot NOW!
HypriotOS: root@dims-rpi in ~
$ Connection to 192.168.0.104 closed by remote host.
Connection to 192.168.0.104 closed.

```

Log in remotely again and validate the container is running.

```

[dimsenv] dittrich@27b:~/git/homepage (develop*) $ !slo
slogin -i ~/.ssh/dims_dittrich_rsa root@192.168.0.104
Linux dims-rpi 3.18.11-hypriotos-v7+ #2 SMP PREEMPT Sun Apr 12 16:34:20 UTC 2015
↪armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 31 16:33:23 2015 from 192.168.0.5
HypriotOS: root@dims-rpi in ~
$ date
Sat Oct 31 18:07:25 PDT 2015
HypriotOS: root@dims-rpi in ~
$ docker ps
CONTAINER ID          IMAGE                                COMMAND                                CREATED
↪                      STATUS                            PORTS                                NAMES
3a8b96428ab4          hypriot/rpi-busybox-httpd:latest    "/bin/busybox httpd"                About a
↪minute ago          Up About a minute                  0.0.0.0:80->80/tcp                  rpi-busybox-httpd_00

```

Lastly, load the URL <http://192.168.0.105> one last time to confirm the Hypriot test page seen in Figure [Hypriot test page](#) is being served after the reboot.

You can also validate `supervisord` activity by checking its log files, which are placed by default in `/var/log/supervisor`:

```

HypriotOS: root@dims-rpi in ~
$ cd /var/log/supervisor
HypriotOS: root@dims-rpi in /var/log/supervisor
$ ls -l
total 12
-rw----- 1 root root    0 Nov  1 00:16 rpi-busybox-httpd_00-stderr---supervisor-
↪d50keu.log

```

```

-rw----- 1 root root 21 Nov 1 00:16 rpi-busybox-httpd_00-stdout---supervisor-
↳dos6Dz.log
-rw-r--r-- 1 root root 7495 Nov 1 00:16 supervisord.log
HypriotOS: root@dims-rpi in /var/log/supervisor
$ cat rpi-busybox-httpd_00-stdout---supervisor-dos6Dz.log
rpi-busybox-httpd_00
HypriotOS: pi@dims-rpi in /var/log/supervisor
$ cat supervisord.log
2015-10-30 22:32:54,750 CRIT Supervisor running as root (no user in config file)
2015-10-30 22:32:54,947 INFO RPC interface 'supervisor' initialized
2015-10-30 22:32:54,947 WARN cElementTree not installed, using slower XML parser for
↳XML-RPC
2015-10-30 22:32:54,948 CRIT Server 'unix_http_server' running without any HTTP
↳authentication checking
2015-10-30 22:32:54,951 INFO daemonizing the supervisord process
2015-10-30 22:32:54,954 INFO supervisord started with pid 4744
2015-10-31 02:17:12,001 CRIT Supervisor running as root (no user in config file)
2015-10-31 02:17:12,282 INFO RPC interface 'supervisor' initialized
2015-10-31 02:17:12,282 WARN cElementTree not installed, using slower XML parser for
↳XML-RPC
2015-10-31 02:17:12,283 CRIT Server 'unix_http_server' running without any HTTP
↳authentication checking
2015-10-31 02:17:12,286 INFO daemonizing the supervisord process
2015-10-31 02:17:12,289 INFO supervisord started with pid 1873
2015-10-31 18:03:22,227 WARN received SIGTERM indicating exit request
2015-10-31 18:03:27,621 CRIT Supervisor running as root (no user in config file)
2015-10-31 18:03:27,621 WARN Included extra file "/etc/supervisor/conf.d/rpi-busybox-
↳httpd.conf" during parsing
2015-10-31 18:03:27,815 INFO RPC interface 'supervisor' initialized
2015-10-31 18:03:27,816 WARN cElementTree not installed, using slower XML parser for
↳XML-RPC
2015-10-31 18:03:27,816 CRIT Server 'unix_http_server' running without any HTTP
↳authentication checking
2015-10-31 18:03:27,819 INFO daemonizing the supervisord process
2015-10-31 18:03:27,822 INFO supervisord started with pid 2501
2015-10-31 18:03:28,829 INFO spawned: 'rpi-busybox-httpd_00' with pid 2505
2015-10-31 18:03:29,832 INFO success: rpi-busybox-httpd_00 entered RUNNING state,
↳process has stayed up for > than 1 seconds (startsecs)
2015-10-31 18:06:09,939 WARN received SIGTERM indicating exit request
2015-10-31 18:06:09,943 INFO waiting for rpi-busybox-httpd_00 to die
2015-10-31 18:06:10,275 INFO stopped: rpi-busybox-httpd_00 (terminated by SIGTERM)
2015-10-31 18:06:10,277 WARN received SIGTERM indicating exit request
2015-10-31 18:06:18,801 CRIT Supervisor running as root (no user in config file)
2015-10-31 18:06:18,803 WARN Included extra file "/etc/supervisor/conf.d/rpi-busybox-
↳httpd.conf" during parsing
2015-10-31 18:06:19,149 INFO RPC interface 'supervisor' initialized
2015-10-31 18:06:19,149 WARN cElementTree not installed, using slower XML parser for
↳XML-RPC
2015-10-31 18:06:19,150 CRIT Server 'unix_http_server' running without any HTTP
↳authentication checking
2015-10-31 18:06:19,154 INFO daemonizing the supervisord process
2015-10-31 18:06:19,157 INFO supervisord started with pid 1894
2015-10-31 18:06:20,169 INFO spawned: 'rpi-busybox-httpd_00' with pid 2079
2015-10-31 18:06:21,537 INFO success: rpi-busybox-httpd_00 entered RUNNING state,
↳process has stayed up for > than 1 seconds (startsecs)

```

**Caution:** The above httpd container uses Busybox (presumably `ash`), and appears to possibly be ignoring any signals it is sent. A more robust container that traps signals and exits properly should be used (e.g., using `nginx`).

### 9.2.3 Extending to other Services

Extending `supervisord` control to other services is as simple as following the same steps as Section *Installing a Persistent Docker Container* with other run scripts and `supervisord` configuration files.

# CHAPTER 10

---

## Docker Datacenter

---

This chapter documents email exchanges between DIMS team members and Docker engineers about setting up and evaluating [Docker Datacenter](#).

### 10.1 Initial Inquiry

This section includes the pdf showing the basics of Docker Datacenter.

[pdf](#)

This pdf was sent along with the response to our initial inquiry to Docker about evaluating Docker Datacenter on 3/2/16.

Jeremy also set up a call with other Docker engineers on 3/2/16.

### 10.2 Docker Trusted Repository Issues

This section documents issues Megan was having when trying to set up a [Docker Trusted Registry](#) as part of a local Docker Datacenter.

### 10.3 Further Information

As more is learned about Docker Datacenter, particularly admin-related information, it will be documented here.

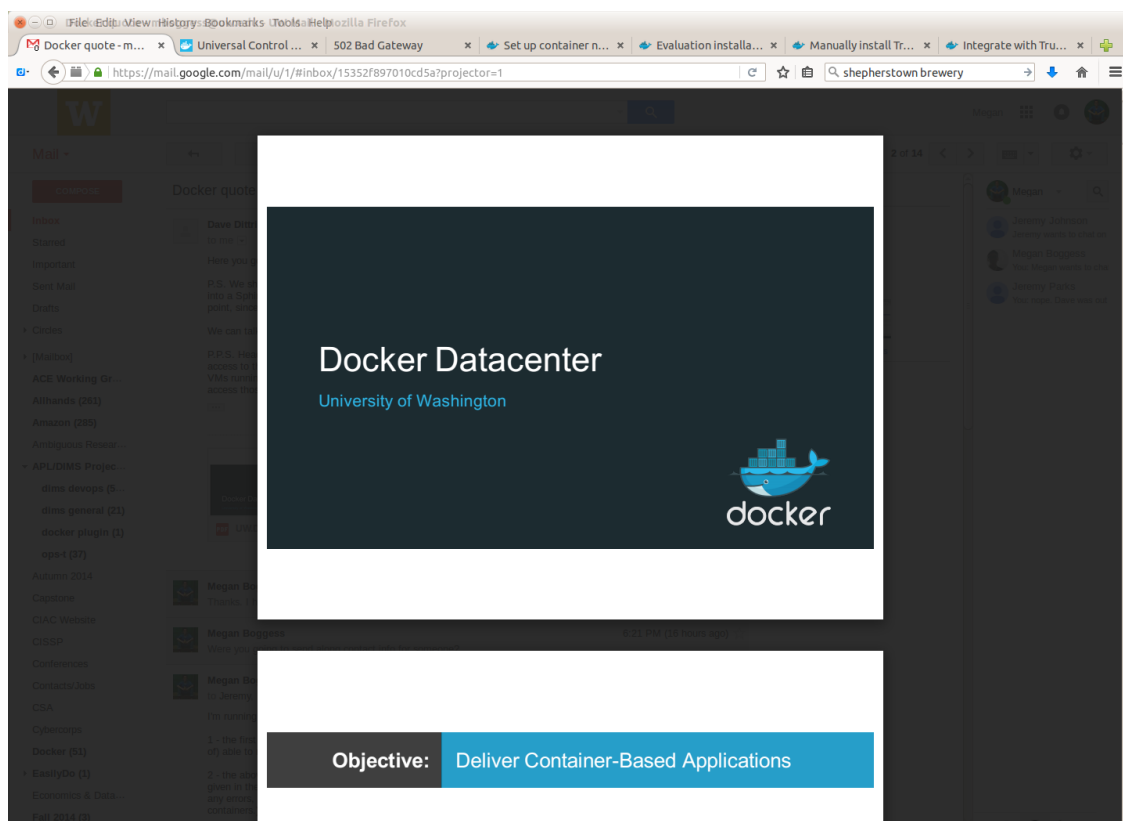


Fig. 10.1: Basics of Docker Datacenter pdf.



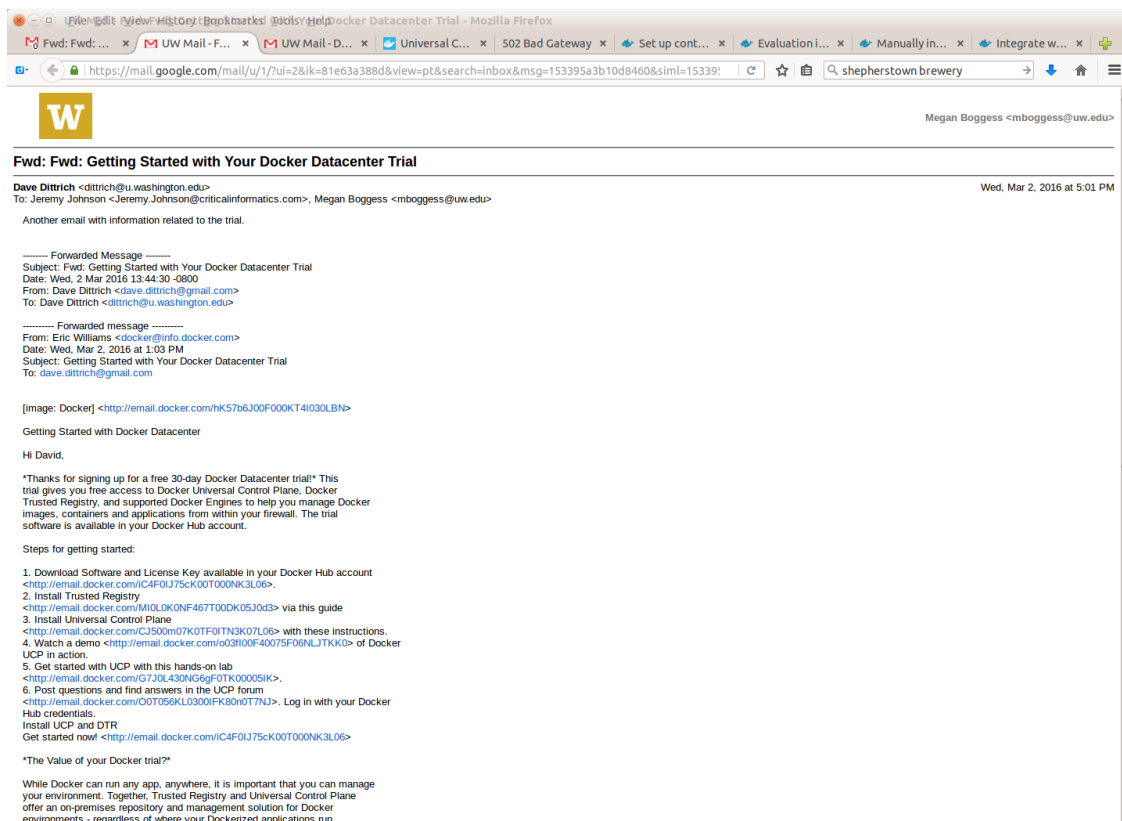


Fig. 10.2: Image 1 of email.

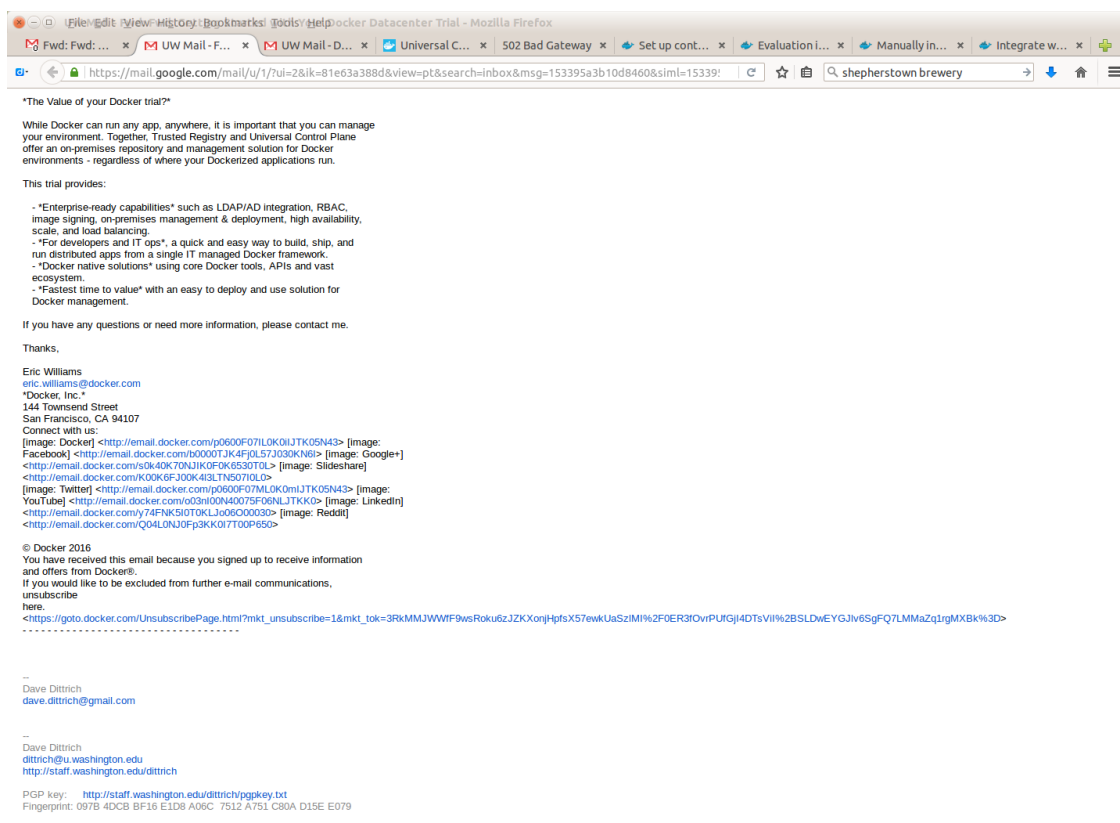


Fig. 10.3: Image 2 of email

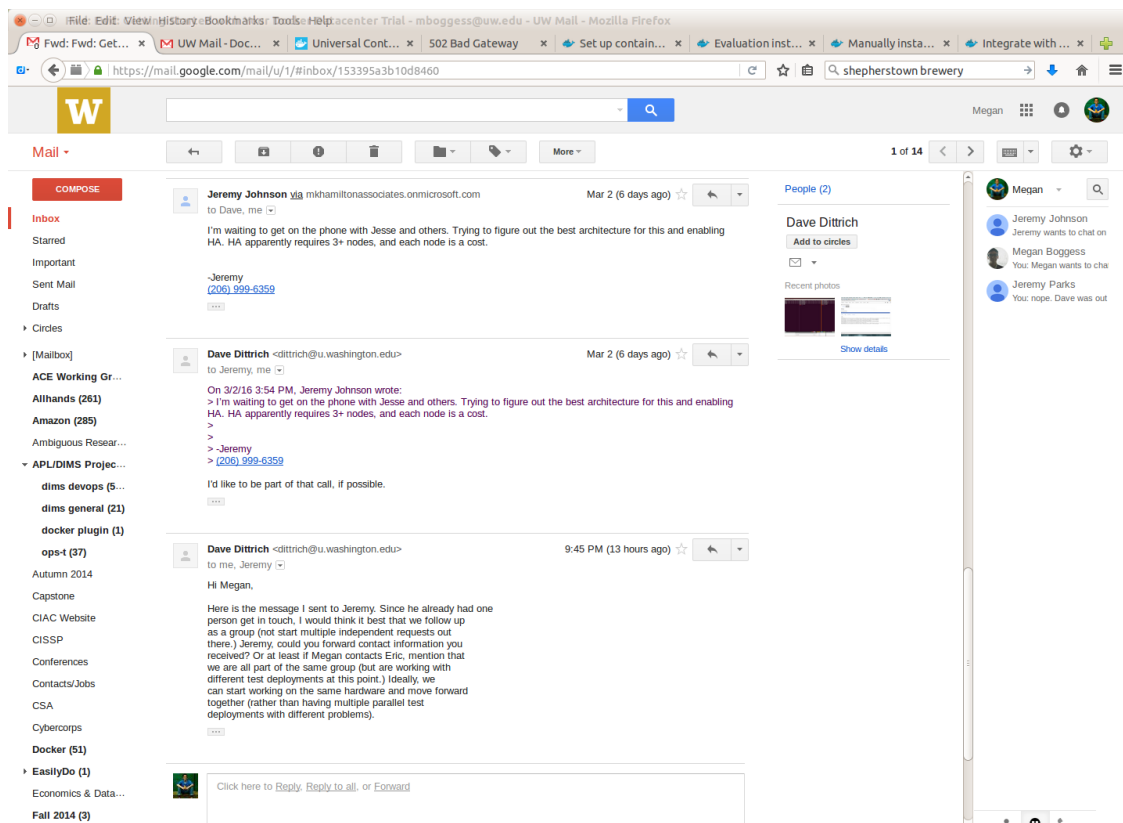


Fig. 10.4: Email re: call with Docker engineers.

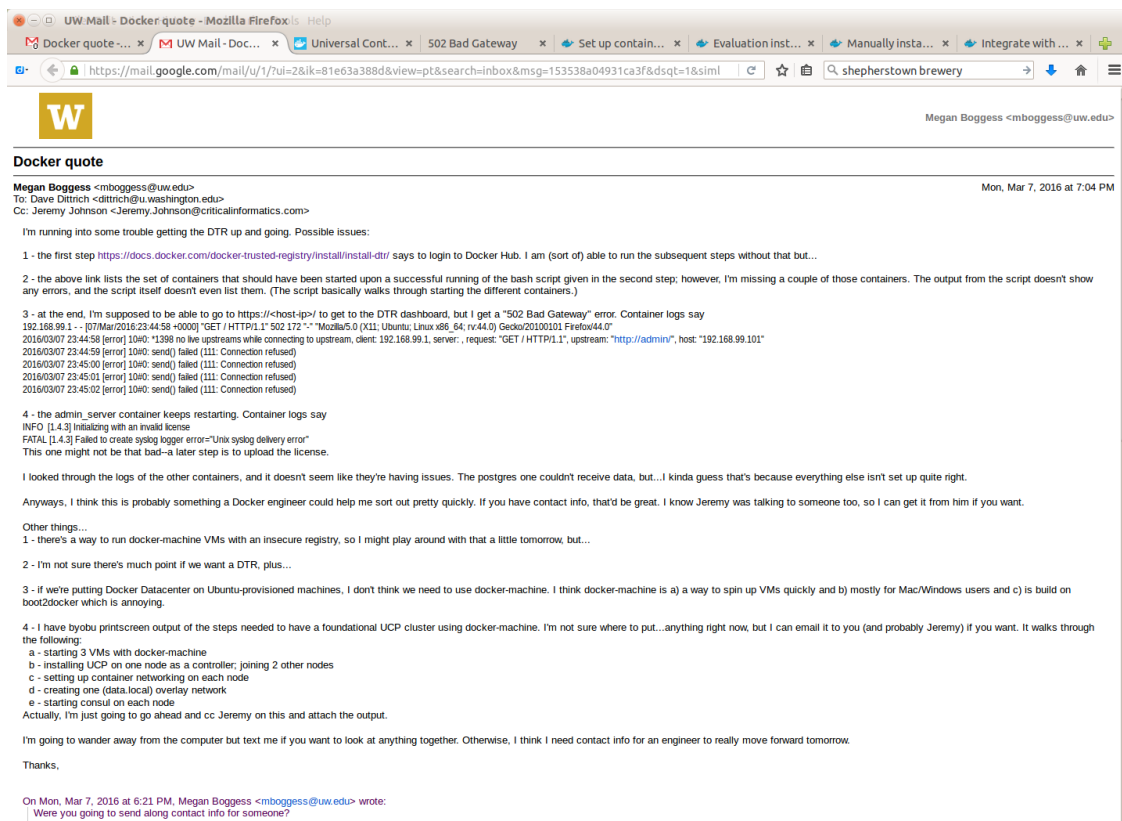


Fig. 10.5: DTR issues.

---

## Managing Long-running Services

---

This chapter covers the process of keeping a service program alive across system reboots, using `supervisord` or **Upstart**. Regardless of which of these mechanisms is used, the concept is similar:

- A program that provides a network service is supposed to be started when the system starts, and stopped when the system is brought down. This should be done cleanly, so that any required state is maintained across reboots.
- If the program exits for any reason, this reason should be checked and acted upon such that the desired goal of having the service be available when you want it to be available is maintained. This means that when the service program exists with an unexpected return code, it is restarted.

---

**Note:** If the program is supposed to be turned off, and it exits with an expected “normal” exit code, it is left off until it is explicitly started again.

---

The `supervisord` program is much simpler than Upstart, but in some cases is sufficient to get the job done with a minimum of effort, and is much easier to debug. Upstart, on the other hand, is very complex and feature-rich, lending to more sophisticated capabilities (e.g., monitoring multiple hierarchical dependent services to control starting and stopping service daemons in complex inter-dependent situations). This flexibility comes at the cost of much more difficulty in designing, developing, and most importantly *debugging* these services and requires significantly greater system administration and programming experience to accomplish. The section on Upstart includes some techniques for debugging services.

---

**Note:** Section *RaspberryPi and Docker* covers this topic in the specific context of a prototype Docker containerized service using the **HypriotOS** on a RaspberryPI. This section covers the same material in the context of the primary operating system used by the DIMS project, **Ubuntu**.

---

## 11.1 Services using supervisord

## 11.2 Services using Upstart

By default, Upstart does not log very much. To see the logging level currently set, do:

```
$ sudo initctl log-priority
message
```

To increase the logging level, do:

```
$ sudo initctl log-priority info
$ sudo initctl log-priority
info
```

Now you can follow the system logs using `sudo tail -f /var/log/syslog` and watch events. In this case, we want to see all of the `init` events associated with restarting the OpenVPN tunnel (which is the pathway used by the Consul agents for communicating.)

To know which events are associated with the action we are about to cause, use the `logger` program to insert markers immediately before the `restart` is triggered. Then wait until it looks like the service is completely restarted before inserting another marker and then copying the log output.

**Attention:** Because service are stopped and started asynchronously in the background, the only marker that is easy to accurately set is the one immediately before the `restart` is triggered. If another `&&` was added to insert a marker **immediately** after the `sudo service openvpn restart` command returned and the shell allowed the `logger` command to run, it would insert the marker in the middle of the actions going on in the background.

Be careful to keep this asynchrony in your mind and separate the act of the shell returning from the unrelated act of the service being restarted, or else you will not get the results you expect.

Additionally, on a busy system there may also be other events that show up in the log file between the `logger` command and the initiation of the `restart` action (and interspersed with the logs that are important for our purposes. You will need to carefully delete those log entries that are not important in order to minimize the “noise” of all the state transition messages from `init`.

- <http://askubuntu.com/questions/28281/what-events-are-available-for-upstart>

```
$ logger -t DITTRICH -p local0.info "Restarting OpenVPN" && sudo service openvpn_
↪restart
* Stopping virtual private network daemon(s)...
*   Stopping VPN '01_prsm_dimsdemo1'
...done.
*   Stopping VPN '02_uwapl_dimsdemo1'
...done.
* Starting virtual private network daemon(s)...
*   Autostarting VPN '01_prsm_dimsdemo1'
*   Autostarting VPN '02_uwapl_dimsdemo1'
$ logger -t DITTRICH -p local0.info "Done"
```

```
Jun  4 20:07:16 dimsdemo1.node.consul DITTRICH: Restarting OpenVPN
Jun  4 20:07:16 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[14113]: event_wait :_
↪Interrupted system call (code=4)
Jun  4 20:07:16 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[14113]: /sbin/ip route_
↪del 10.142.29.0/24
```

```

Jun  4 20:07:16 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[14113]: ERROR: Linux_
↳route delete command failed: external program exited with error status: 2
Jun  4 20:07:16 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[14113]: Closing TUN/TAP_
↳interface
Jun  4 20:07:16 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[14113]: /sbin/ip addr_
↳del dev tun0 10.86.86.4/24
Jun  4 20:07:16 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[14113]: Linux ip addr_
↳del failed: external program exited with error status: 2
Jun  4 20:07:16 dimsdemo1.node.consul NetworkManager[1055]:      SCPlugin-Ifupdown:_
↳devices removed (path: /sys/devices/virtual/net/tun0, iface: tun0)
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.461020] init: Handling queues-
↳device-removed event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.461202] init: Handling queues-
↳device-removed event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.461321] init: Handling net-
↳device-removed event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.461372] init: network-interface_
↳(tun0) goal changed from start to stop
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.461400] init: network-interface_
↳(tun0) state changed from running to stopping
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.461449] init: Handling stopping_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.461482] init: network-interface_
↳(tun0) state changed from stopping to killed
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.461517] init: network-interface_
↳(tun0) state changed from killed to post-stop
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.462204] init: network-interface_
↳(tun0) post-stop process (26911)
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.463454] init: network-interface_
↳(tun0) post-stop process (26911) exited normally
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.463512] init: network-interface_
↳(tun0) state changed from post-stop to waiting
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.463686] init: Handling stopped_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.463772] init: startpar-bridge_
↳(network-interface-tun0-stopped) goal changed from stop to start
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.463807] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from waiting to starting
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.463929] init: network-interface-
↳security (network-interface/tun0) goal changed from start to stop
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.463956] init: network-interface-
↳security (network-interface/tun0) state changed from running to stopping
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464026] init: Handling starting_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464080] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from starting to security
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464113] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from security to pre-start
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464146] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from pre-start to spawned
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464639] init: startpar-bridge_
↳(network-interface-tun0-stopped) main process (26914)
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464660] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from spawned to post-start
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464705] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from post-start to running
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464784] init: Handling stopping_
↳event

```

```

Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464903] init: network-interface-
↳security (network-interface/tun0) state changed from stopping to killed
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464936] init: network-interface-
↳security (network-interface/tun0) state changed from killed to post-stop
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.464967] init: network-interface-
↳security (network-interface/tun0) state changed from post-stop to waiting
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.465100] init: Handling started_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.465180] init: Handling stopped_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.465236] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) goal changed from stop_
↳to start
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.465267] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from_
↳waiting to starting
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.465339] init: Handling starting_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.465379] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from_
↳starting to security
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.465410] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from_
↳security to pre-start
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.465438] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from pre-
↳start to spawned
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466165] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) main process (26915)
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466190] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from_
↳spawned to post-start
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466244] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from post-
↳start to running
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466331] init: Handling started_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466610] init: startpar-bridge_
↳(network-interface-tun0-stopped) main process (26914) exited normally
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466667] init: startpar-bridge_
↳(network-interface-tun0-stopped) goal changed from start to stop
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466729] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from running to stopping
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466796] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) main process (26915)
↳exited normally
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466848] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) goal changed from start_
↳to stop
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466883] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from_
↳running to stopping
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466921] init: Handling stopping_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466959] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from stopping to killed
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.466990] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from killed to post-stop

```



```

Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.467020] init: startpar-bridge_
↳(network-interface-tun0-stopped) state changed from post-stop to waiting
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.467134] init: Handling stopping_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.467169] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from_
↳stopping to killed
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.467199] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from_
↳killed to post-stop
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.467248] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-stopped) state changed from post-
↳stop to waiting
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.467398] init: Handling stopped_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul kernel: [58061.467490] init: Handling stopped_
↳event
Jun  4 20:07:16 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[14113]: SIGTERM[hard,]_
↳received, process exiting
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: event_wait :_
↳Interrupted system call (code=4)
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: /sbin/ip route_
↳del 38.111.193.0/24
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: ERROR: Linux_
↳route delete command failed: external program exited with error status: 2
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: /sbin/ip route_
↳del 199.168.91.0/24
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: ERROR: Linux_
↳route delete command failed: external program exited with error status: 2
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: /sbin/ip route_
↳del 192.168.88.0/24
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: ERROR: Linux_
↳route delete command failed: external program exited with error status: 2
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: Closing TUN/TAP_
↳interface
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: /sbin/ip addr_
↳del dev tun88 10.88.88.5/24
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: Linux ip addr_
↳del failed: external program exited with error status: 2
Jun  4 20:07:17 dimsdemo1.node.consul NetworkManager[1055]:      SCPlugin-Ifupdown:_
↳devices removed (path: /sys/devices/virtual/net/tun88, iface: tun88)
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.504410] init: Handling queues-
↳device-removed event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.504612] init: Handling queues-
↳device-removed event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.504723] init: Handling net-
↳device-removed event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.504763] init: network-interface_
↳(tun88) goal changed from start to stop
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.504799] init: network-interface_
↳(tun88) state changed from running to stopping
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.504844] init: Handling stopping_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.504877] init: network-interface_
↳(tun88) state changed from stopping to killed
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.504907] init: network-interface_
↳(tun88) state changed from killed to post-stop
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.505652] init: network-interface_
↳(tun88) post-stop process (26927)

```

```

Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.506919] init: network-interface_
↳(tun88) post-stop process (26927) exited normally
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.506976] init: network-interface_
↳(tun88) state changed from post-stop to waiting
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.507159] init: Handling stopped_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.507234] init: startpar-bridge_
↳(network-interface-tun88-stopped) goal changed from stop to start
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.507263] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from waiting to starting
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.507431] init: network-interface-
↳security (network-interface/tun88) goal changed from start to stop
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.507470] init: network-interface-
↳security (network-interface/tun88) state changed from running to stopping
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.507511] init: Handling starting_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.507554] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from starting to security
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.507575] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from security to pre-start
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.507594] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from pre-start to spawned
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508094] init: startpar-bridge_
↳(network-interface-tun88-stopped) main process (26930)
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508133] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from spawned to post-start
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508181] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from post-start to running
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508275] init: Handling stopping_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508410] init: network-interface-
↳security (network-interface/tun88) state changed from stopping to killed
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508441] init: network-interface-
↳security (network-interface/tun88) state changed from killed to post-stop
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508473] init: network-interface-
↳security (network-interface/tun88) state changed from post-stop to waiting
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508609] init: Handling started_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508713] init: Handling stopped_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508803] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) goal changed from stop_
↳to start
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508863] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from_
↳waiting to starting
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.508967] init: Handling starting_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.509008] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from_
↳starting to security
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.509060] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from_
↳security to pre-start
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.509109] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from pre-
↳start to spawned
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.509733] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) main process (26931)

```

```

Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.509753] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from_
↳spawned to post-start
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.509804] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from_
↳post-start to running
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.509897] init: Handling started_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510246] init: startpar-bridge_
↳(network-interface-tun88-stopped) main process (26930) exited normally
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510303] init: startpar-bridge_
↳(network-interface-tun88-stopped) goal changed from start to stop
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510366] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from running to stopping
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510433] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) main process (26931)_
↳exited normally
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510501] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) goal changed from_
↳start to stop
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510535] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from_
↳running to stopping
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510573] init: Handling stopping_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510610] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from stopping to killed
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510642] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from killed to post-stop
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510672] init: startpar-bridge_
↳(network-interface-tun88-stopped) state changed from post-stop to waiting
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510785] init: Handling stopping_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510819] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from_
↳stopping to killed
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510849] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from_
↳killed to post-stop
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.510879] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-stopped) state changed from_
↳post-stop to waiting
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.511028] init: Handling stopped_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul kernel: [58061.511120] init: Handling stopped_
↳event
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[14127]: SIGTERM[hard,]_
↳received, process exiting
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26949]: OpenVPN 2.3.2_
↳x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [EPOLL] [PKCS11] [eurephia] [MH] [IPv6]_
↳built on Dec  1 2014
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26949]: Control Channel_
↳Authentication: tls-auth using INLINE static key file
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26949]: Outgoing Control_
↳Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC authentication
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26949]: Incoming Control_
↳Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC authentication
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26949]: Socket Buffers:_
↳R=[212992->131072] S=[212992->131072]

```

```

Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: NOTE: UID/GID_
↳downgrade will be delayed because of --client, --pull, or --up-delay
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: UDPv4 link_
↳local: [undef]
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: UDPv4 link_
↳remote: [AF_INET]140.142.29.115:500
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26963]: OpenVPN 2.3.2_
↳x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [EPOLL] [PKCS11] [eurephia] [MH] [IPv6]_
↳built on Dec  1 2014
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26963]: Control Channel_
↳Authentication: tls-auth using INLINE static key file
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26963]: Outgoing_
↳Control Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC_
↳authentication
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26963]: Incoming_
↳Control Channel Authentication: Using 160 bit message hash 'SHA1' for HMAC_
↳authentication
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26963]: Socket Buffers:_
↳R=[212992->131072] S=[212992->131072]
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: NOTE: UID/GID_
↳downgrade will be delayed because of --client, --pull, or --up-delay
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: UDPv4 link_
↳local: [undef]
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: UDPv4 link_
↳remote: [AF_INET]140.142.29.118:8989
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: TLS: Initial_
↳packet from [AF_INET]140.142.29.118:8989, sid=adf2b40a afa33d74
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: TLS: Initial_
↳packet from [AF_INET]140.142.29.115:500, sid=3cf9074f 2e93fa51
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: Data Channel_
↳Encrypt: Cipher 'AES-128-CBC' initialized with 128 bit key
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: Data Channel_
↳Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: Data Channel_
↳Decrypt: Cipher 'AES-128-CBC' initialized with 128 bit key
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: Data Channel_
↳Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: Control Channel:_
↳TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 2048 bit RSA
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: [eclipse-prisem]_
↳Peer Connection Initiated with [AF_INET]140.142.29.115:500
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: Data Channel_
↳Encrypt: Cipher 'AES-128-CBC' initialized with 128 bit key
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: Data Channel_
↳Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: Data Channel_
↳Decrypt: Cipher 'AES-128-CBC' initialized with 128 bit key
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: Data Channel_
↳Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: Control_
↳Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 2048 bit RSA
Jun  4 20:07:17 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: [server] Peer_
↳Connection Initiated with [AF_INET]140.142.29.118:8989
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: SENT CONTROL_
↳[eclipse-prisem]: 'PUSH_REQUEST' (status=1)
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: PUSH: Received_
↳control message: ...
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: OPTIONS IMPORT:_
↳timers and/or timeouts modified

```

```

Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: OPTIONS IMPORT: ↵
↵LZO parms modified
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: OPTIONS IMPORT: -
↵-ifconfig/up options modified
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: OPTIONS IMPORT: ↵
↵route options modified
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: OPTIONS IMPORT: ↵
↵route-related options modified
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: OPTIONS IMPORT: -
↵-ip-win32 and/or --dhcp-option options modified
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: ROUTE_GATEWAY ↵
↵192.168.0.1/255.255.255.0 IFACE=wlan0 HWADDR=d0:53:49:d7:9e:bd
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: TUN/TAP device ↵
↵tun0 opened
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: TUN/TAP TX queue ↵
↵length set to 100
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: do_ifconfig, tt->
↵ipv6=0, tt->did_ifconfig_ipv6_setup=0
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: /sbin/ip link ↵
↵set dev tun0 up mtu 1500
Jun  4 20:07:19 dimsdemo1.node.consul NetworkManager[1055]:      SCPlugin-Ifupdown: ↵
↵devices added (path: /sys/devices/virtual/net/tun0, iface: tun0)
Jun  4 20:07:19 dimsdemo1.node.consul NetworkManager[1055]:      SCPlugin-Ifupdown: ↵
↵device added (path: /sys/devices/virtual/net/tun0, iface: tun0): no ifupdown ↵
↵configuration found.
Jun  4 20:07:19 dimsdemo1.node.consul NetworkManager[1055]: <warn> /sys/devices/
↵virtual/net/tun0: couldn't determine device driver; ignoring...
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: /sbin/ip addr ↵
↵add dev tun0 10.86.86.4/24 broadcast 10.86.86.255
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.897552] init: Handling net-
↵device-added event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.897768] init: network-interface ↵
↵(tun0) goal changed from stop to start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.897831] init: network-interface ↵
↵(tun0) state changed from waiting to starting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.897933] init: Handling starting ↵
↵event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.898119] init: network-interface-
↵security (network-interface/tun0) goal changed from stop to start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.898175] init: network-interface-
↵security (network-interface/tun0) state changed from waiting to starting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.898246] init: Handling starting ↵
↵event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.898319] init: network-interface-
↵security (network-interface/tun0) state changed from starting to security
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.898373] init: network-interface-
↵security (network-interface/tun0) state changed from security to pre-start
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: /sbin/ip route ↵
↵add 10.142.29.0/24 via 10.86.86.1
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.899415] init: network-interface-
↵security (network-interface/tun0) pre-start process (27032)
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.899754] init: Handling queues-
↵device-added event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.900062] init: Handling queues-
↵device-added event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.900301] init: network-interface-
↵security (network-interface/tun0) pre-start process (27032) exited normally
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.900403] init: network-interface-
↵security (network-interface/tun0) state changed from pre-start to spawned

```

```

Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.900465] init: network-interface-
↳security (network-interface/tun0) state changed from spawned to post-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.900527] init: network-interface-
↳security (network-interface/tun0) state changed from post-start to running
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.900591] init: network-interface_
↳(tun0) state changed from starting to security
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.900641] init: network-interface_
↳(tun0) state changed from security to pre-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.901534] init: network-interface_
↳(tun0) pre-start process (27033)
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.901884] init: Handling started_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.902189] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) goal changed from stop_
↳to start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.902361] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from_
↳waiting to starting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.902728] init: Handling starting_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: GID set to_
↳nogroup
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: UID set to nobody
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-01_prsm_dimsdemo1[26950]: Initialization_
↳Sequence Completed
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.902874] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from_
↳starting to security
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.903036] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from_
↳security to pre-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.903191] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from pre-
↳start to spawned
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.904568] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) main process (27035)
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.904606] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from_
↳spawned to post-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.904693] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from post-
↳start to running
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.904841] init: Handling started_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.905285] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) main process (27035)_
↳exited normally
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.905430] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) goal changed from start_
↳to stop
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.905509] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from_
↳running to stopping
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.905583] init: Handling stopping_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.905688] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from_
↳stopping to killed

```



```

Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.905752] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from_
↳killed to post-stop
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.905809] init: startpar-bridge_
↳(network-interface-security-network-interface/tun0-started) state changed from post-
↳stop to waiting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.906042] init: Handling stopped_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907410] init: network-interface_
↳(tun0) pre-start process (27033) exited normally
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907464] init: network-interface_
↳(tun0) state changed from pre-start to spawned
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907497] init: network-interface_
↳(tun0) state changed from spawned to post-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907531] init: network-interface_
↳(tun0) state changed from post-start to running
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907616] init: Handling started_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907693] init: startpar-bridge_
↳(network-interface-tun0-started) goal changed from stop to start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907727] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from waiting to starting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907816] init: Handling starting_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907870] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from starting to security
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907897] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from security to pre-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.907927] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from pre-start to spawned
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.908460] init: startpar-bridge_
↳(network-interface-tun0-started) main process (27039)
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.908481] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from spawned to post-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.908526] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from post-start to running
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.908606] init: Handling started_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.908945] init: startpar-bridge_
↳(network-interface-tun0-started) main process (27039) exited normally
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.909008] init: startpar-bridge_
↳(network-interface-tun0-started) goal changed from start to stop
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.909044] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from running to stopping
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.909082] init: Handling stopping_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.909120] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from stopping to killed
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.909151] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from killed to post-stop
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.909183] init: startpar-bridge_
↳(network-interface-tun0-started) state changed from post-stop to waiting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58063.909293] init: Handling stopped_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: SENT CONTROL_
↳[server]: 'PUSH_REQUEST' (status=1)
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: PUSH: Received_
↳control message: ...

```

```

Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: OPTIONS IMPORT:
↳timers and/or timeouts modified
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: OPTIONS IMPORT:
↳--ifconfig/up options modified
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: OPTIONS IMPORT:
↳route options modified
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: OPTIONS IMPORT:
↳route-related options modified
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: ROUTE_GATEWAY
↳192.168.0.1/255.255.255.0 IFACE=wlan0 HWADDR=d0:53:49:d7:9e:bd
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: TUN/TAP device
↳tun88 opened
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: TUN/TAP TX
↳queue length set to 100
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: do_ifconfig, tt-
↳>ipv6=0, tt->did_ifconfig_ipv6_setup=0
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: /sbin/ip link
↳set dev tun88 up mtu 1500
Jun  4 20:07:19 dimsdemo1.node.consul NetworkManager[1055]: SCPlugin-Ifupdown:
↳devices added (path: /sys/devices/virtual/net/tun88, iface: tun88)
Jun  4 20:07:19 dimsdemo1.node.consul NetworkManager[1055]: SCPlugin-Ifupdown:
↳device added (path: /sys/devices/virtual/net/tun88, iface: tun88): no ifupdown
↳configuration found.
Jun  4 20:07:19 dimsdemo1.node.consul NetworkManager[1055]: <warn> /sys/devices/
↳virtual/net/tun88: couldn't determine device driver; ignoring...
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: /sbin/ip addr
↳add dev tun88 10.88.88.2/24 broadcast 10.88.88.255
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: /sbin/ip route
↳add 192.168.88.0/24 via 10.88.88.1
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.341486] init: Handling net-
↳device-added event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.341622] init: network-interface
↳(tun88) goal changed from stop to start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.341655] init: network-interface
↳(tun88) state changed from waiting to starting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.341714] init: Handling starting
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.341838] init: network-interface-
↳security (network-interface/tun88) goal changed from stop to start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.341869] init: network-interface-
↳security (network-interface/tun88) state changed from waiting to starting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.341905] init: Handling starting
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.341945] init: network-interface-
↳security (network-interface/tun88) state changed from starting to security
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.341976] init: network-interface-
↳security (network-interface/tun88) state changed from security to pre-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.342560] init: network-interface-
↳security (network-interface/tun88) pre-start process (27060)
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.342787] init: Handling queues-
↳device-added event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.342956] init: Handling queues-
↳device-added event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.343091] init: network-interface-
↳security (network-interface/tun88) pre-start process (27060) exited normally
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.343149] init: network-interface-
↳security (network-interface/tun88) state changed from pre-start to spawned
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.343187] init: network-interface-
↳security (network-interface/tun88) state changed from spawned to post-start

```



```

Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.343217] init: network-interface-
↳security (network-interface/tun88) state changed from post-start to running
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.343275] init: network-interface_
↳(tun88) state changed from starting to security
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.343310] init: network-interface_
↳(tun88) state changed from security to pre-start
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: /sbin/ip route_
↳add 199.168.91.0/24 via 10.88.88.1
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: /sbin/ip route_
↳add 38.111.193.0/24 via 10.88.88.1
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: GID set to_
↳nogroup
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: UID set to_
↳nobody
Jun  4 20:07:19 dimsdemo1.node.consul ovpn-02_uwapl_dimsdemo1[26964]: Initialization_
↳Sequence Completed
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.343904] init: network-interface_
↳(tun88) pre-start process (27062)
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344021] init: Handling started_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344112] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) goal changed from stop_
↳to start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344155] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from_
↳waiting to starting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344310] init: Handling starting_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344352] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from_
↳starting to security
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344387] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from_
↳security to pre-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344418] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from pre-
↳start to spawned
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344889] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) main process (27064)
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344908] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from_
↳spawned to post-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.344956] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from_
↳post-start to running
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.345036] init: Handling started_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.345420] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) main process (27064)_
↳exited normally
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.345490] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) goal changed from_
↳start to stop
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.345534] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from_
↳running to stopping
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.345573] init: Handling stopping_
↳event

```

```

Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.345641] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from_
↳stopping to killed
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.345680] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from_
↳killed to post-stop
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.345709] init: startpar-bridge_
↳(network-interface-security-network-interface/tun88-started) state changed from_
↳post-stop to waiting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.345834] init: Handling stopped_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347178] init: network-interface_
↳(tun88) pre-start process (27062) exited normally
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347251] init: network-interface_
↳(tun88) state changed from pre-start to spawned
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347299] init: network-interface_
↳(tun88) state changed from spawned to post-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347333] init: network-interface_
↳(tun88) state changed from post-start to running
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347414] init: Handling started_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347488] init: startpar-bridge_
↳(network-interface-tun88-started) goal changed from stop to start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347525] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from waiting to starting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347619] init: Handling starting_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347660] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from starting to security
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347691] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from security to pre-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.347719] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from pre-start to spawned
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348254] init: startpar-bridge_
↳(network-interface-tun88-started) main process (27069)
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348277] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from spawned to post-start
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348328] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from post-start to running
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348422] init: Handling started_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348731] init: startpar-bridge_
↳(network-interface-tun88-started) main process (27069) exited normally
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348796] init: startpar-bridge_
↳(network-interface-tun88-started) goal changed from start to stop
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348841] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from running to stopping
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348874] init: Handling stopping_
↳event
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348913] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from stopping to killed
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348934] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from killed to post-stop
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.348953] init: startpar-bridge_
↳(network-interface-tun88-started) state changed from post-stop to waiting
Jun  4 20:07:19 dimsdemo1.node.consul kernel: [58064.349059] init: Handling stopped_
↳event
Jun  4 20:07:36 dimsdemo1.node.consul DITTRICH: Done

```





---

## Diagnosing System Problems and Outages

---

### 12.1 Using `dimsscli`

This chapter covers using `dimsscli` as a distributed shell for diagnosing problems throughout a DIMS deployment.

Ansible has two primary CLI programs, `ansible` and `ansible-playbook`. Both of these programs are passed a set of hosts on which they are to operate using an [Inventory](#).

---

**Note:** Read about Ansible and how it is used by the DIMS project in Section [ansibleplaybooks:ansiblefundamentals](#) of [ansibleplaybooks:ansibleplaybooks](#).

---

```
[dimesenv] dittrich@dimsdemo1:~/dims/git/python-dimsscli (develop*) $ cat complete_  
→inventory  
[all]  
floyd2-p.prisem.washington.edu  
foswiki-int.prisem.washington.edu  
git.prisem.washington.edu  
hub.prisem.washington.edu  
jenkins-int.prisem.washington.edu  
jira-int.prisem.washington.edu  
lapp-int.prisem.washington.edu  
lapp.prisem.washington.edu  
linda-vm1.prisem.washington.edu  
rabbitmq.prisem.washington.edu  
sso.prisem.washington.edu  
time.prisem.washington.edu  
u12-dev-svr-1.prisem.washington.edu  
u12-dev-ws-1.prisem.washington.edu  
wellington.prisem.washington.edu
```

Using this inventory, the modules `command` and `shell` can be used to run commands as needed to diagnose all of these hosts at once.

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimscli (develop*) $ dimscli ansible_
↪command --program "uptime" --inventory complete_inventory --remote-port 8422 --
↪remote-user dittrich
```

Host	Status	Results
rabbitmq.prisem.washington.edu	GOOD	22:07:53 up 33 days, 4:32, 1 user, load average: 0.07, 0.13, 0.09
wellington.prisem.washington.edu	GOOD	22:07:57 up 159 days, 12:16, 1 user, load average: 1.16, 0.86, 0.58
linda-vm1.prisem.washington.edu	GOOD	22:07:54 up 159 days, 12:03, 1 user, load average: 0.00, 0.01, 0.05
git.prisem.washington.edu	GOOD	22:07:54 up 159 days, 12:03, 2 users, load average: 0.00, 0.01, 0.05
time.prisem.washington.edu	GOOD	22:07:55 up 33 days, 4:33, 2 users, load average: 0.01, 0.07, 0.12
jenkins-int.prisem.washington.edu	GOOD	22:07:55 up 159 days, 12:03, 1 user, load average: 0.00, 0.01, 0.05
ul2-dev-ws-1.prisem.washington.edu	GOOD	22:07:56 up 159 days, 12:03, 1 user, load average: 0.00, 0.02, 0.05
sso.prisem.washington.edu	GOOD	22:07:56 up 159 days, 12:03, 1 user, load average: 0.00, 0.01, 0.05
lapp-int.prisem.washington.edu	GOOD	22:07:54 up 159 days, 12:04, 2 users, load average: 0.00, 0.01, 0.05
foswiki-int.prisem.washington.edu	GOOD	22:07:55 up 159 days, 12:04, 1 user, load average: 0.00, 0.01, 0.05
ul2-dev-svr-1.prisem.washington.edu	GOOD	22:07:59 up 155 days, 14:56, 1 user, load average: 0.05, 0.08, 0.06
hub.prisem.washington.edu	GOOD	06:07:53 up 141 days, 12:19, 1 user, load average: 0.08, 0.03, 0.05
floyd2-p.prisem.washington.edu	GOOD	22:07:53 up 33 days, 4:32, 1 user, load average: 0.00, 0.01, 0.05
jira-int.prisem.washington.edu	GOOD	22:07:54 up 159 days, 12:03, 2 users, load average: 0.00, 0.01, 0.05
lapp.prisem.washington.edu	GOOD	22:07:54 up 159 days, 12:04, 2 users, load average: 0.00, 0.01, 0.05

```
1 To: dims-devops@uw.ops-trust.net
2 From: Jenkins <dims@eclipse.prisem.washington.edu>
3 Subject: [dims devops] [Jenkins] [FAILURE] jenkins-update-cifbulk-server-develop-16
4 Date: Thu Jan 14 20:35:21 PST 2016
5 Message-ID: <20160115043521.C7D5E1C004F@jenkins>
```

```
6
7 Started by an SCM change
8 [EnvInject] - Loading node environment variables.
9 Building in workspace /var/lib/jenkins/jobs/update-cifbulk-server-develop/workspace
10
11 Deleting project workspace... done
```

```
12
13 [ssh-agent] Using credentials ansible (Ansible user ssh key - root)
14 [ssh-agent] Looking for ssh-agent implementation...
15 [ssh-agent] Java/JNR ssh-agent
16 [ssh-agent] Started.
```

```

17     ...
18
19
20 TASK: [cifbulk-server | Make config change available and restart if updating_
    ↪existing] ***
21 <rabbitmq.prisem.washington.edu> REMOTE_MODULE command . /opt/dims/envs/dimsenv/bin/
    ↪activate && supervisorctl -c /etc/supervisord.conf reread #USE_SHELL
22 failed: [rabbitmq.prisem.washington.edu] => (item=reread) => {"changed": true, "cmd
    ↪": ". /opt/dims/envs/dimsenv/bin/activate && supervisorctl -c /etc/supervisord.conf_
    ↪reread", "delta": "0:00:00.229614", "end": "2016-01-14 20:34:49.409784", "item":
    ↪"reread", "rc": 2, "start": "2016-01-14 20:34:49.180170"}
23 stderr: Error: could not find config file /etc/supervisord.conf
24 For help, use /usr/bin/supervisorctl -h
25 <rabbitmq.prisem.washington.edu> REMOTE_MODULE command . /opt/dims/envs/dimsenv/bin/
    ↪activate && supervisorctl -c /etc/supervisord.conf update #USE_SHELL
26 failed: [rabbitmq.prisem.washington.edu] => (item=update) => {"changed": true, "cmd
    ↪": ". /opt/dims/envs/dimsenv/bin/activate && supervisorctl -c /etc/supervisord.conf_
    ↪update", "delta": "0:00:00.235882", "end": "2016-01-14 20:34:50.097224", "item":
    ↪"update", "rc": 2, "start": "2016-01-14 20:34:49.861342"}
27 stderr: Error: could not find config file /etc/supervisord.conf
28 For help, use /usr/bin/supervisorctl -h
29
30 FATAL: all hosts have already failed -- aborting
31
32 PLAY RECAP *****
33         to retry, use: --limit @/var/lib/jenkins/cifbulk-server-configure.retry
34
35 rabbitmq.prisem.washington.edu : ok=11    changed=4    unreachable=0    failed=1
36
37 Build step 'Execute shell' marked build as failure
38 [ssh-agent] Stopped.
39 Warning: you have no plugins providing access control for builds, so falling back to_
    ↪legacy behavior of permitting any downstream builds to be triggered
40 Finished: FAILURE
41 --
42 [[ UW/DIMS ]]: All message content remains the property of the author
43 and must not be forwarded or redistributed without explicit permission.

```

```

[dimsenv] dittrich@dimsdemo1:~/dims/git/ansible-playbooks (develop*) $ grep -r_
    ↪supervisord.conf
roles/supervisor-install/tasks/main.yml:  template: "src=supervisord.conf.j2 dest={{_
    ↪dims_supervisord_conf }} owner=root group=root"
roles/supervisor-install/tasks/main.yml:  file: path=/etc/dims-supervisord.conf_
    ↪state=absent
roles/supervisor-install/templates/supervisor.j2:DAEMON_OPTS="-c {{ dims_supervisord_
    ↪conf }} $DAEMON_OPTS"
roles/cifbulk-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} &&_
    ↪supervisorctl -c {{ dims_supervisord_conf }} {{ item }}"
roles/cifbulk-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} &&_
    ↪supervisorctl -c {{ dims_supervisord_conf }} start {{ name_base }}:"
roles/prisem-scripts-deploy/tasks/main.yml:  shell: ". {{ dimsenv_activate }} &&_
    ↪supervisorctl -c {{ dims_supervisord_conf }} restart {{ item }}:"
roles/anon-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -
    ↪c {{ dims_supervisord_conf }} {{ item }}"
roles/anon-server/tasks/main.yml:  shell: ". {{ dimsenv_activate }} && supervisorctl -
    ↪c {{ dims_supervisord_conf }} start {{ name_base }}:"
roles/consul-install/tasks/main.yml:  shell: ". {{ dimsenv_activate }} &&_
    ↪supervisorctl -c {{ dims_supervisord_conf }} remove {{ consul_basename }}"

```

```
roles/consul-install/tasks/main.yml: shell: ". {{ dimsenv_activate }} &&
↳supervisorctl -c {{ dims_supervisord_conf }} {{ item }}"
roles/consul-install/tasks/main.yml: shell: ". {{ dimsenv_activate }} &&
↳supervisorctl -c {{ dims_supervisord_conf }} start {{ consul_basename }}:"
roles/crosscor-server/tasks/main.yml: shell: ". {{ dimsenv_activate }} &&
↳supervisorctl -c {{ dims_supervisord_conf }} {{ item }}"
roles/crosscor-server/tasks/main.yml: shell: ". {{ dimsenv_activate }} &&
↳supervisorctl -c {{ dims_supervisord_conf }} start {{ name_base }}:"
group_vars/all:dims_supervisord_conf: '/etc/supervisord.conf'
```

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimsccli (develop*) $ dimsccli ansible_
↳shell --program "find /etc -name supervisord.conf" --inventory complete_inventory --
↳remote-port 8422 --remote-u
```

```
ser dittrich
```

Host	Status	Results
rabbitmq.prisem.washington.edu	GOOD	/etc/supervisor/supervisord.conf
wellington.prisem.washington.edu	GOOD	
hub.prisem.washington.edu	GOOD	
git.prisem.washington.edu	GOOD	/etc/supervisor/supervisord.conf
ul2-dev-ws-1.prisem.washington.edu	GOOD	
sso.prisem.washington.edu	GOOD	
jenkins-int.prisem.washington.edu	GOOD	/etc/supervisor/supervisord.conf
foswiki-int.prisem.washington.edu	GOOD	
lapp-int.prisem.washington.edu	GOOD	
ul2-dev-svr-1.prisem.washington.edu	GOOD	/etc/supervisor/supervisord.conf
linda-vm1.prisem.washington.edu	GOOD	
lapp.prisem.washington.edu	GOOD	
floyd2-p.prisem.washington.edu	GOOD	
jira-int.prisem.washington.edu	GOOD	/etc/supervisor/supervisord.conf
time.prisem.washington.edu	GOOD	

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimsccli (develop*) $ dimsccli ansible_
↳shell --program "find /etc -name '*supervisor*'" --inventory complete_inventory --
↳remote-port 8422 --remote-use
```

```
r dittrich
```

Host	Status	Results
rabbitmq.prisem.washington.edu	GOOD	/etc/rc0.d/K20supervisor
		/etc/rc3.d/S20supervisor
		/etc/rc1.d/K20supervisor
		/etc/default/supervisor
		/etc/rc2.d/S20supervisor
		/etc/rc6.d/K20supervisor
		/etc/supervisor



		/etc/supervisor/supervisord.conf.	
↪20140214204135		/etc/supervisor/supervisord.conf.	
↪20140214200547		/etc/supervisor/supervisord.conf.	
↪20140616162335		/etc/supervisor/supervisord.conf.	
↪20140814132409		/etc/supervisor/supervisord.conf.	
↪20140616162451		/etc/supervisor/supervisord.conf.	
↪20140616162248		/etc/supervisor/supervisord.conf.	
↪20140131230939		/etc/supervisor/supervisord.conf.	
↪20140222154901		/etc/supervisor/supervisord.conf.	
↪20140214194415		/etc/supervisor/supervisord.conf.	
↪20140222155042		/etc/supervisor/supervisord.conf.	
↪20150208174308		/etc/supervisor/supervisord.conf.	
↪20140814132717		/etc/supervisor/supervisord.conf.	
↪20140215134451		/etc/supervisor/supervisord.conf.	
↪20150208174742		/etc/supervisor/supervisord.conf.	
↪20140911193305		/etc/supervisor/supervisord.conf.	
↪20140219200951		/etc/supervisor/supervisord.conf.	
↪20140911202633		/etc/supervisor/supervisord.conf	⌋
↪		/etc/supervisor/supervisord.conf.	
↪20140222154751		/etc/supervisor/supervisord.conf.	
↪20150208174403		/etc/supervisor/supervisord.conf.	
↪20140814132351		/etc/supervisor/supervisord.conf.	
↪20140814132759		/etc/rc4.d/S20supervisor	⌋
↪		/etc/init.d/supervisor	⌋
↪		/etc/rc5.d/S20supervisor	⌋
↪			⌋
wellington.prisem.washington.edu	GOOD		⌋
↪		/etc/rc0.d/K20supervisor	⌋
↪		/etc/rc3.d/S20supervisor	⌋
↪		/etc/rc1.d/K20supervisor	⌋
↪			

			/etc/rc2.d/S20supervisor	
↪			/etc/rc6.d/K20supervisor	
↪			/etc/supervisor	
↪			/etc/rc4.d/S20supervisor	
↪			/etc/dims-supervisord.conf	
↪			/etc/init.d/supervisor	
↪			/etc/rc5.d/S20supervisor	
↪			/etc/rc0.d/K20supervisor	
git.prisem.washington.edu		GOOD	/etc/rc3.d/S20supervisor	
↪			/etc/rc1.d/K20supervisor	
↪			/etc/default/supervisor	
↪			/etc/rc2.d/S20supervisor	
↪			/etc/rc6.d/K20supervisor	
↪			/etc/supervisor	
↪			/etc/supervisor/supervisord.conf	
↪			/etc/rc4.d/S20supervisor	
↪			/etc/init.d/supervisor	
↪			/etc/rc5.d/S20supervisor	
↪				
time.prisem.washington.edu		GOOD		
↪			/etc/rc0.d/K20supervisor	
↪			/etc/rc3.d/S20supervisor	
↪			/etc/rc1.d/K20supervisor	
↪			/etc/default/supervisor	
↪			/etc/rc2.d/S20supervisor	
↪			/etc/rc6.d/K20supervisor	
↪			/etc/supervisor	
↪			/etc/supervisor/supervisord.conf	
↪			/etc/rc4.d/S20supervisor	
↪			/etc/init.d/supervisor	

			/etc/rc5.d/S20supervisor	U
↪				
	ul2-dev-ws-1.prisem.washington.edu	GOOD		U
↪				
	sso.prisem.washington.edu	GOOD		U
↪				
	lapp-int.prisem.washington.edu	GOOD		U
↪				
	foswiki-int.prisem.washington.edu	GOOD		U
↪				
	ul2-dev-svr-1.prisem.washington.edu	GOOD	/etc/rc2.d/S20supervisor	U
↪				
			/etc/rc4.d/S20supervisor	U
↪				
			/etc/init.d/supervisor	U
↪				
			/etc/rc5.d/S20supervisor	U
↪				
			/etc/rc3.d/S20supervisor	U
↪				
			/etc/supervisor	U
↪				
			/etc/supervisor/supervisord.conf	U
↪				
			/etc/rc6.d/K20supervisor	U
↪				
			/etc/rc1.d/K20supervisor	U
↪				
			/etc/rc0.d/K20supervisor	U
↪				
	hub.prisem.washington.edu	GOOD		U
↪				
	floyd2-p.prisem.washington.edu	GOOD		U
↪				
	jira-int.prisem.washington.edu	GOOD	/etc/rc0.d/K20supervisor	U
↪				
			/etc/rc3.d/S20supervisor	U
↪				
			/etc/rc1.d/K20supervisor	U
↪				
			/etc/default/supervisor	U
↪				
			/etc/rc2.d/S20supervisor	U
↪				
			/etc/rc6.d/K20supervisor	U
↪				
			/etc/supervisor	U
↪				
			/etc/supervisor/supervisord.conf	U
↪				
			/etc/rc4.d/S20supervisor	U
↪				
			/etc/init.d/supervisor	U
↪				
			/etc/rc5.d/S20supervisor	U
↪				
	lapp.prisem.washington.edu	GOOD		U
↪				

```
+-----+-----+-----+
↪-----+
```

While the concept of putting a list of host names into a file with a label is simple to understand, it is not very flexible or scalable. Ansible supports a concept called a [Dynamic Inventory](#). Rather than passing a hosts file using `-i` or `--inventory`, you can pass a Python script that produces a special JSON object.

What is not very widely known is that you can also trigger creation of a dynamic inventory within `ansible` or `ansible-playbook` by passing a *list* for the `-i` or `--inventory` option. Rather than creating a temporary file with `[all]` at the top, followed by a list of three host names, then passing that file with `-i` or `--inventory`, just pass a comma-separated list instead:

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimsccli (develop*) $ dimsccli ansible_
↪shell --program "find /etc -name supervisord.conf" --inventory rabbitmq.prisem.
↪washington.edu,time.prisem.washi
ngton.edu,ul2-dev-svr-1.prisem.washington.edu --remote-port 8422 --remote-user_
↪dittrich
```

Host	Status	Results
rabbitmq.prisem.washington.edu	GOOD	/etc/supervisor/supervisord.conf
time.prisem.washington.edu	GOOD	
ul2-dev-svr-1.prisem.washington.edu	GOOD	/etc/supervisor/supervisord.conf

There is a subtle trick for passing just a single host, and that is to pass the name with a trailing comma (`,`), as seen here:

```
[dimsenv] dittrich@dimsdemo1:~/dims/git/python-dimsccli (develop*) $ dimsccli ansible_
↪shell --program "find /etc -name supervisord.conf" --inventory rabbitmq.prisem.
↪washington.edu, --remote-port 84
22 --remote-user dittrich
```

Host	Status	Results
rabbitmq.prisem.washington.edu	GOOD	/etc/supervisor/supervisord.conf

...

## 12.2 Debugging Vagrant

Vagrant has a mechanism for enabling debugging output to determine what it is doing. That mechanism is to set an environment variable `VAGRANT_LOG=debug` before running `vagrant`.

```
$ vagrant halt
$ VAGRANT_LOG=debug vagrant up --no-provision > /tmp/debug.log.1 2>&1
```

The debugging log looks like the following:

```
INFO global: Vagrant version: 1.8.6
INFO global: Ruby version: 2.2.5
INFO global: RubyGems version: 2.4.5.1
INFO global: VAGRANT_LOG="debug"
INFO global: VAGRANT_OLD_ENV_TMPDIR="/tmp"
```

```

INFO global: VAGRANT_OLD_ENV_COMMAND=""
INFO global: VAGRANT_OLD_ENV_LANG="en_US.UTF-8"
INFO global: VAGRANT_OLD_ENV_UNDEFINED="__undefined__"
INFO global: VAGRANT_OLD_ENV_TERM="screen-256color"
INFO global: VAGRANT_OLD_ENV_VAGRANT_LOG="debug"

. . .

INFO global: VAGRANT_INTERNAL_BUNDLERIZED="1"
INFO global: Plugins:
INFO global:   - bundler = 1.12.5
INFO global:   - unf_ext = 0.0.7.2
INFO global:   - unf = 0.1.4
INFO global:   - domain_name = 0.5.20161129
INFO global:   - http-cookie = 1.0.3
INFO global:   - i18n = 0.7.0
INFO global:   - log4r = 1.1.10
INFO global:   - micromachine = 2.0.0
INFO global:   - mime-types-data = 3.2016.0521
INFO global:   - mime-types = 3.1
INFO global:   - net-ssh = 3.0.2
INFO global:   - net-scp = 1.1.2
INFO global:   - netrc = 0.11.0
INFO global:   - rest-client = 2.0.0
INFO global:   - vagrant-scp = 0.5.7
INFO global:   - vagrant-share = 1.1.6
INFO global:   - vagrant-triggers = 0.5.3
INFO global:   - vagrant-vbguest = 0.13.0

. . .

INFO vagrant: `vagrant` invoked: ["up"]
DEBUG vagrant: Creating Vagrant environment
INFO environment: Environment initialized (#<Vagrant::Environment:0x00000002618e68>)
INFO environment:   - cwd: /vm/run/blue14
INFO environment: Home path: /home/ansible/.vagrant.d
DEBUG environment: Effective local data path: /vm/run/blue14/.vagrant
INFO environment: Local data path: /vm/run/blue14/.vagrant
DEBUG environment: Creating: /vm/run/blue14/.vagrant
INFO environment: Running hook: environment_plugins_loaded
INFO runner: Preparing hooks for middleware sequence...
INFO runner: 3 hooks defined.
INFO runner: Running action: environment_plugins_loaded #
↳<Vagrant::Action::Builder:0x000000025278b0>

. .

DEBUG meta: Finding driver for VirtualBox version: 5.1.10
INFO meta: Using VirtualBox driver:↳
↳VagrantPlugins::ProviderVirtualBox::Driver::Version_5_1
INFO base: VBoxManage path: VBoxManage
INFO subprocess: Starting process: ["/usr/bin/VBoxManage", "showvminfo", "d1f7ffcb-
↳3fab-4878-a77d-5fdb8d2f7fae"]
INFO subprocess: Command not in installer, restoring original environment...
DEBUG subprocess: Selecting on IO
DEBUG subprocess: stdout: Name: blue14_default_1482088614789_39851
Groups: /
Guest OS: Ubuntu (64-bit)

```

```
UUID:                d1f7ffcb-3fab-4878-a77d-5fdb8d2f7fae
Config file:         /home/ansible/VirtualBox VMs/blue14_default_1482088614789_39851/
↳blue14_default_1482088614789_39851.vbox
Snapshot folder:    /home/ansible/VirtualBox VMs/blue14_default_1482088614789_39851/
↳Snapshots
Log folder:         /home/ansible/VirtualBox VMs/blue14_default_1482088614789_39851/Logs
Hardware UUID:      d1f7ffcb-3fab-4878-a77d-5fdb8d2f7fae
Memory size:        3072MB
Page Fusion:        off
VRAM size:          32MB
CPU exec cap:       100%

. . .

Effective Paravirt. Provider: KVM
State:              powered off (since 2016-10-30T20:11:22.000000000)
Monitor count:      1
3D Acceleration:    off
2D Video Acceleration: off
Teleporter Enabled: off

. . .
```

For this debugging scenario, we are trying to add the ability to toggle whether Vagrant brings up the Virtualbox VM with or without a GUI (i.e., “headless” or not). The line we are concerned about here is the following line, which shows the `startvm` line used to run the Virtualbox VM:

```
INFO subprocess: Starting process: ["/usr/bin/VBoxManage", "startvm", "89e0e942-3b3b-
↳4f0a-b0e4-6d0bb51fef04", "--type", "headless"]
```

The default for Vagrant is to start VMs in headless mode. To instead boot with a GUI, the Vagrantfile should contain a provisioner block with the following setting:

```
config.vm.provider "virtualbox" do |v|
  v.gui = true
end
```

---

**Note:** It is important to note that the Vagrantfile is Ruby code, and that the above sets a Ruby boolean to the value `true`, which is *not necessarily* the same as the string `"true"`.

---

Rather than requiring that the user edit the Vagrantfile, it would be more convenient to support passing an environment variable into the child process.

Using the following code snippets, we can inherit an environment variable (which is a string) and turn it into a boolean using a string comparison operation in a ternary logical expression.

```
# Set GUI to boolean false if environment variable GUI == 'true'
GUI = ENV['GUI'].nil? ? false : (ENV['GUI'] == 'true')

. . .

# Conditionally control whether startvm uses "--type gui"
# or "--type headless" using GUI (set earlier)
config.vm.provider "virtualbox" do |v|
  v.gui = GUI
end
```

```
. . .
```

Now we can test the setting of the environment variable on a `vagrant` command line, again with debug logging enabled and redirected into a second log file.

```
$ vagrant halt
==> default: Attempting graceful shutdown of VM...
$ vagrant destroy --force
==> default: Destroying VM and associated drives...
$ GUI=true VAGRANT_LOG=debug vagrant up --no-provision > /tmp/debug.log.2 2>&1
```

Now looking for the specific string in the output of both files, we can compare the results and see that we have the desired effect:

```
$ grep 'Starting process.*startvm' /tmp/debug.log.{1,2}
/tmp/debug.log.1: INFO subprocess: Starting process: ["/usr/bin/VBoxManage", "startvm
↪", "89e0e942-3b3b-4f0a-b0e4-6d0bb51fef04", "--type", "headless"]
/tmp/debug.log.2: INFO subprocess: Starting process: ["/usr/bin/VBoxManage", "startvm
↪", "3921e4e9-fdb4-4191-90b3-f7415ec0b37d", "--type", "gui"]
```

## 12.3 Other Tools for Diagnosing System Problems

### 12.3.1 smartmontools

Hardware makes up the physical layer of the DIMS system. Developers are currently using Dell Precision M4800 laptops to develop the software layers of DIMS.

These laptops have had multiple issues, specifically including not sleeping properly and heating up to extreme temperatures, heating up to extreme temperatures when not sitting on solid, very well ventilated surfaces, and these specific problems have led to malfunctions with the hard drives. At least one laptop has completely stopped being able to boot. Multiple other laptops have struggled during the boot up process and have had other problems that may indicate a near-term hard drive failure.

In an effort to turn a black box into less of a black box and to try to see ahead of time if there are any indicators that may be pointing to a failure before a failure, we are now employing the use of a tool called `smartmontools`. This package comes with two tools – `smartctl` and `smartd` – which control and monitor storage systems using the Self-Monitoring, Analysis and Reporting Technology System (SMART) built in to a lot of modern hard drives, including the ones on the developer laptops. When using this tool as a daemon, it can give advanced warning of disk degradation and failure. (For more information, see [smartmontools home](#).)

The package will be added to the list of base packages installed on all DIMS systems, and the rest of this section will be devoted to a brief introduction for how to use the tool.

---

**Note:** These instructions were taken from [ubuntu smartmontools docs](#). If it differs on other Linux flavors (particularly Debian Jessie), new instructions will be added.

---

You will be using the `smartctl` utility to manually monitor your drives. First, you need to double check that your hard drive is SMART-enabled.

```
1 [dimsenv] mboggess@dimsdev2:it/dims-adminguide/docs/source (develop*) $ sudo smartctl_
↪ -i /dev/sda
2 smartctl 6.2 2013-07-26 r3841 [x86_64-linux-4.4.0-42-generic] (local build)
```

```

3 Copyright (C) 2002-13, Bruce Allen, Christian Franke, www.smartmontools.org
4
5 === START OF INFORMATION SECTION ===
6 Model Family:      Seagate Laptop SSHD
7 Device Model:      ST1000LM014-1EJ164
8 Serial Number:     W771CY1P
9 LU WWN Device Id:  5 000c50 089fc94f9
10 Firmware Version:  DEMB
11 User Capacity:     1,000,204,886,016 bytes [1.00 TB]
12 Sector Sizes:      512 bytes logical, 4096 bytes physical
13 Rotation Rate:     5400 rpm
14 Device is:         In smartctl database [for details use: -P show]
15 ATA Version is:    ACS-2, ACS-3 T13/2161-D revision 3b
16 SATA Version is:   SATA 3.1, 6.0 Gb/s (current: 6.0 Gb/s)
17 Local Time is:     Fri Oct 14 11:08:25 2016 EDT
18 SMART support is:  Available - device has SMART capability.
19 SMART support is:  Enabled

```

This output gives you information about the hard drive, including if SMART is support and enabled.

In the event that somehow SMART is available but not enabled, run

```
sudo smartctl -s on /dev/sda
```

There are several different types of tests you can run via smartctl. A full list is documented in the help/usage output which you can obtain by running

```
[dimsenv] mboggess@dimsdev2:it/dims-adminguide/docs/source (develop*) $ smartctl -h
```

To find an estimate of the time it will take to complete the various tests, run

```

1 [dimsenv] mboggess@dimsdev2:it/dims-adminguide/docs/source (develop*) $ sudo smartctl_
  ↪-c /dev/sda
2 smartctl 6.2 2013-07-26 r3841 [x86_64-linux-4.4.0-42-generic] (local build)
3 Copyright (C) 2002-13, Bruce Allen, Christian Franke, www.smartmontools.org
4
5 === START OF READ SMART DATA SECTION ===
6 General SMART Values:
7 Offline data collection status:  (0x00) Offline data collection activity
8                                     was never started.
9                                     Auto Offline Data Collection: Disabled.
10 Self-test execution status:      (   0) The previous self-test routine completed
11                                     without error or no self-test has ever
12                                     been run.
13 Total time to complete Offline
14 data collection:                  ( 139) seconds.
15 Offline data collection
16 capabilities:                     (0x73) SMART execute Offline immediate.
17                                     Auto Offline data collection on/off support.
18                                     Suspend Offline collection upon new
19                                     command.
20                                     No Offline surface scan supported.
21                                     Self-test supported.
22                                     Conveyance Self-test supported.
23                                     Selective Self-test supported.
24 SMART capabilities:              (0x0003) Saves SMART data before entering
25                                     power-saving mode.
26                                     Supports SMART auto save timer.

```



```

27 Error logging capability:      (0x01) Error logging supported.
28                               General Purpose Logging supported.
29 Short self-test routine
30 recommended polling time:      (  2) minutes.
31 Extended self-test routine
32 recommended polling time:      ( 191) minutes.
33 Conveyance self-test routine
34 recommended polling time:      (   3) minutes.
35 SCT capabilities:              (0x10b5) SCT Status supported.
36                               SCT Feature Control supported.
37                               SCT Data Table supported.

```

As you can see, the long test is rather long—191 minutes!

To run the long test, run

```

[dimsenv] mboggess@dimsdev2:it/dims-adminguide/docs/source (develop*) $ sudo smartctl
↳ -t long /dev/sda
smartctl 6.2 2013-07-26 r3841 [x86_64-linux-4.4.0-42-generic] (local build)
Copyright (C) 2002-13, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===
Sending command: "Execute SMART Extended self-test routine immediately in off-line
↳ mode".
Drive command "Execute SMART Extended self-test routine immediately in off-line mode"
↳ successful.
Testing has begun.
Please wait 191 minutes for test to complete.
Test will complete after Fri Oct 14 15:00:32 2016

Use smartctl -X to abort test.

```

To abort the test:

```

[dimsenv] mboggess@dimsdev2:it/dims-adminguide/docs/source (develop*) $ sudo smartctl
↳ -X /dev/sda
smartctl 6.2 2013-07-26 r3841 [x86_64-linux-4.4.0-42-generic] (local build)
Copyright (C) 2002-13, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===
Sending command: "Abort SMART off-line mode self-test routine".
Self-testing aborted!

```

To get test results, for a SATA drive, run

```

[dimsenv] mboggess@dimsdev2:it/dims-adminguide/docs/source (develop*) $ sudo smartctl
↳ -a -d ata /dev/sda

```

To get test results, for an IDE drive, run

```

[dimsenv] mboggess@dimsdev2:it/dims-adminguide/docs/source (develop*) $ sudo smartctl
↳ -a /dev/sda

```

Additionally, you can run smartmontools as a daemon, but for now, that will be left for an admin to research and develop on their own. In the future, this has potential to be turned into an Ansible role. Documentation from Ubuntu on how to use smartmontools as a daemon can be found in the [daemon subsection](#) of the Ubuntu smartmontools documentation.



---

## Managing CoreOS with Systemd and Other Tools

---

This chapter covers using `systemctl` commands and other debugging commands and services for diagnosing problems on a CoreOS system.

CoreOS uses `systemd` as both a system and service manager and as an init system. The tool `systemctl` has many commands which allow a user to look at and control the state of `systemd`.

This is by no means an exhaustive list or description of the potential of any of the tools described here, merely an overview of tools and their most useful services. See the links provided within this chapter for more information. For more debugging information relevant to DIMS, see `dimsdockerfiles:debuggingcoreos`.

### 13.1 State of systemd

There are a few ways to check on the state of `systemd`, as a whole system.

1. Check all running units and their state on a node at once.

```

1 core@core-01 ~ $ systemctl
2 UNIT                                LOAD    ACTIVE    SUB                DESCRIPTIO
3 boot.automount                     loaded active    waiting           Boot parti
4 sys-devices-pci0000:00-0000:00:01.1-ata1-host0-target0:0:0-0:0:0-0:0-
5 sys-devices-pci0000:00-0000:00:01.1-ata1-host0-target0:0:0-0:0:0-0:0-
6 sys-devices-pci0000:00-0000:00:01.1-ata1-host0-target0:0:0-0:0:0-0:0-
7 sys-devices-pci0000:00-0000:00:01.1-ata1-host0-target0:0:0-0:0:0-0:0-
8 sys-devices-pci0000:00-0000:00:01.1-ata1-host0-target0:0:0-0:0:0-0:0-
9 sys-devices-pci0000:00-0000:00:01.1-ata1-host0-target0:0:0-0:0:0-0:0-
10 sys-devices-pci0000:00-0000:00:01.1-ata1-host0-target0:0:0-0:0:0-0:0-
11 sys-devices-pci0000:00-0000:00:01.1-ata1-host0-target0:0:0-0:0:0-0:0-
12 sys-devices-pci0000:00-0000:00:03.0-virtio0-net-eth0.device loaded
13 sys-devices-pci0000:00-0000:00:08.0-virtio1-net-eth1.device loaded
14 sys-devices-platform-serial8250-tty-ttyS0.device loaded active
15 sys-devices-platform-serial8250-tty-ttyS1.device loaded active
16 sys-devices-platform-serial8250-tty-ttyS2.device loaded active
17 sys-devices-platform-serial8250-tty-ttyS3.device loaded active

```

```

18 sys-devices-virtual-net-docker0.device loaded active plugged
19 sys-devices-virtual-net-vethcbb3671.device loaded active plugge
20 sys-devices-virtual-tty-ttyprintk.device loaded active plugged
21 sys-subsystem-net-devices-docker0.device loaded active plugged
22 sys-subsystem-net-devices-eth0.device loaded active plugged
23 sys-subsystem-net-devices-eth1.device loaded active plugged
24 sys-subsystem-net-devices-vethcbb3671.device loaded active plug
25 -.mount loaded active mounted /
26 boot.mount loaded active mounted Boot parti
27 dev-hugepages.mount loaded active mounted Huge Pages
28 dev-mqueue.mount loaded active mounted POSIX Mess
29 media.mount loaded active mounted External M
30 sys-kernel-debug.mount loaded active mounted Debug File
31 tmp.mount loaded active mounted Temporary
32 usr-share-oem.mount loaded active mounted /usr/share
33 usr.mount loaded active mounted /usr
34 coreos-cloudinit-vagrant-user.path loaded active running c
35 motdgen.path loaded active waiting Watch for
36 systemd-ask-password-console.path loaded active waiting Di
37 systemd-ask-password-wall.path loaded active waiting Forwa
38 user-cloudinit@var-lib-coreos\x2dinstall-user_data.path loaded acti
39 user-configdrive.path loaded active waiting Watch for
40 docker-201c7bd05ea49b654aa8b02a92dbb739a06dd3e8a4cc7813dcddc15aa4282
41 docker-5f41c7d23012a856462d3a7876d7165715164d2b2c6edf3f94449c21d594
42 docker-8323ab8192308e5a65102dffb109466c6a7c7f43ff28f356ea154a668b5f
43 app-overlay.service loaded activating auto-restart App overla
44 audit-rules.service loaded active exited Load Secur
45 consul.service loaded active running Consul boo
46 coreos-setup-environment.service loaded active exited Mod
47 data-overlay.service loaded activating auto-restart Data overl
48 dbus.service loaded active running D-Bus Syst
49 docker.service loaded active running Docker App
50 etcd2.service loaded active running etcd2
51 fleet.service loaded active running fleet daem
52 getty@tty1.service loaded active running Getty on t
53 kmod-static-nodes.service loaded active exited Create lis
54 locksmithd.service loaded active running Cluster re
55 settimezone.service loaded active exited Set the ti
56 sshd-keygen.service loaded active exited Generate s
57 sshd@2-10.0.2.15:22-10.0.2.2:33932.service loaded active runnin
58 swarm-agent.service loaded active running Swarm agen
59 swarm-manager.service loaded active running Swarm mana
60 system-cloudinit@usr-share-oem-cloud\x2dconfig.yml.service loaded a
61 system-cloudinit@var-tmp-hostname.yml.service loaded active exi
62 system-cloudinit@var-tmp-networks.yml.service loaded active exi
63 systemd-journal-flush.service loaded active exited Flush
64 systemd-journald.service loaded active running Journal Se
65 systemd-logind.service loaded active running Login Serv
66 systemd-networkd.service loaded active running Network Se
67 systemd-random-seed.service loaded active exited Load/Sav
68 systemd-resolved.service loaded active running Network Na
69 systemd-sysctl.service loaded active exited Apply Kern
70 systemd-timesyncd.service loaded active running Network Ti
71 systemd-tmpfiles-setup-dev.service loaded active exited C
72 ...skipping...
73 systemd-udev-trigger.service loaded active exited udev Co
74 systemd-udevd.service loaded active running udev Kerne
75 systemd-update-utmp.service loaded active exited Update U

```

```

76 systemd-vconsole-setup.service loaded active exited Setup
77 update-engine.service loaded active running Update Eng
78 user-cloudinit@var-lib-coreos\x2dvagrant-vagrantfile\x2duser\x2ddat
79 -.slice loaded active active Root Slice
80 system-addon\x2dconfig.slice loaded active active system-
81 system-addon\x2drun.slice loaded active active system-add
82 system-getty.slice loaded active active system-get
83 system-sshd.slice loaded active active system-ssh
84 system-system\x2dcloudinit.slice loaded active active sys
85 system-user\x2dcloudinit.slice loaded active active syste
86 system.slice loaded active active System Sli
87 user.slice loaded active active User and S
88 dbus.socket loaded active running D-Bus Syst
89 docker-tcp.socket loaded active running Docker Soc
90 docker.socket loaded active running Docker Soc
91 fleet.socket loaded active running Fleet API
92 rkt-metadata.socket loaded active listening rkt metada
93 sshd.socket loaded active listening OpenSSH Se
94 systemd-initctl.socket loaded active listening /dev/initc
95 systemd-journald-audit.socket loaded active running Journa
96 systemd-journald-dev-log.socket loaded active running Jour
97 systemd-journald.socket loaded active running Journal So
98 systemd-networkd.socket loaded active running networkd r
99 systemd-udev-control.socket loaded active running udev Co
100 systemd-udev-kernel.socket loaded active running udev Ker
101 basic.target loaded active active Basic Syst
102 cryptsetup.target loaded active active Encrypted
103 getty.target loaded active active Login Prom
104 local-fs-pre.target loaded active active Local File
105 local-fs.target loaded active active Local File
106 multi-user.target loaded active active Multi-User
107 network.target loaded active active Network
108 paths.target loaded active active Paths
109 remote-fs.target loaded active active Remote Fil
110 slices.target loaded active active Slices
111 sockets.target loaded active active Sockets
112 swap.target loaded active active Swap
113 sysinit.target loaded active active System Ini
114 system-config.target loaded active active Load syste
115 time-sync.target loaded active active System Tim
116 timers.target loaded active active Timers
117 user-config.target loaded active active Load user-
118 logrotate.timer loaded active waiting Daily Log
119 rkt-gc.timer loaded active waiting Periodic G
120 systemd-tmpfiles-clean.timer loaded active waiting Daily C
121
122 LOAD = Reflects whether the unit definition was properly loaded.
123 ACTIVE = The high-level unit activation state, i.e. generalization
124 SUB = The low-level unit activation state, values depend on unit
125
126 119 loaded units listed. Pass --all to see loaded but inactive unit
127 To show all installed unit files use 'systemctl list-unit-files'.

```

This shows all loaded units and their state, as well as a brief description of the units.

- For a slightly more organized look at the state of a node, along with a list of failed unites, queued jobs, and a process tree based on CGroup:

```

1 [dimesnv] mbogges@dimsdev2:~/core-local () $ vagrant ssh core-03
2 VM name: core-03 - IP: 172.17.8.103
3 Last login: Tue Jan 26 15:49:34 2016 from 10.0.2.2
4 CoreOS beta (877.1.0)
5 core@core-03 ~ $ systemctl status
6 ● core-03
7   State: starting
8   Jobs: 4 queued
9   Failed: 0 units
10  Since: Wed 2016-01-27 12:40:52 EST; 1min 0s ago
11  CGroup: /
12          +-1 /usr/lib/systemd/systemd --switched-root --system --
13          +-system.slice
14              +-dbus.service
15                  | +-509 /usr/bin/dbus-daemon --system --address=system
16          +-update-engine.service
17                  | +-502 /usr/sbin/update_engine -foreground -logtostde
18          +-system-sshd.slice
19                  | +-sshd@2-10.0.2.15:22-10.0.2.2:58499.service
20                  |   +-869 sshd: core [priv]
21                  |   +-871 sshd: core@pts/0
22                  |   +-872 -bash
23                  |   +-878 systemctl status
24                  |   +-879 systemctl status
25          +-systemd-journald.service
26                  | +-387 /usr/lib/systemd/systemd-journald
27          +-systemd-resolved.service
28                  | +-543 /usr/lib/systemd/systemd-resolved
29          +-systemd-timesyncd.service
30                  | +-476 /usr/lib/systemd/systemd-timesyncd
31          +-systemd-logind.service
32                  | +-505 /usr/lib/systemd/systemd-logind
33          +-systemd-networkd.service
34                  | +-837 /usr/lib/systemd/systemd-networkd
35          +-system-getty.slice
36                  | +-getty@tty1.service
37                  |   +-507 /sbin/agetty --noclear tty1 linux
38          +-system-user\x2dcloudinit.slice
39                  | +-user-cloudinit@var-lib-coreos\x2dvagrant-vagrantfi
40                  |   +-658 /usr/bin/coreos-cloudinit --from-file=/var/l
41          +-systemd-udev.service
42                  | +-414 /usr/lib/systemd/systemd-udev
43          +-locksmithd.service
44                  | +-504 /usr/lib/locksmith/locksmithd
45          +-docker.service
46                  +-547 docker daemon --dns 172.18.0.1 --dns 8.8.8.8 -
47                  +-control
48                  +-742 /usr/bin/systemctl stop docker

```

This shows the status of the node (line 7), how many jobs are queued (line 8), and any failed units (line 9). It also shows which services have started, and what command they are running at the time this status “snapshot” was taken.

```

1 core@core-01 ~ $ systemctl status
2 ● core-01
3   State: running
4   Jobs: 2 queued
5   Failed: 0 units

```

```

6      Since: Wed 2016-01-27 12:40:13 EST; 3min 28s ago
7      CGroup: /
8          +-1 /usr/lib/systemd/systemd --switched-root --system --
9          +-system.slice
10             +-docker-5f41c7d23012a856462d3a7876d7165715164d2b2c6ed
11             | +-1475 /swarm join --addr=172.17.8.101:2376 consul:/
12             +-dbus.service
13             | +-508 /usr/bin/dbus-daemon --system --address=system
14             +-update-engine.service
15             | +-517 /usr/sbin/update_engine -foreground -logtostde
16             +-system-sshd.slice
17             | +-sshd@2-10.0.2.15:22-10.0.2.2:33932.service
18             |   +- 860 sshd: core [priv]
19             |   +- 862 sshd: core@pts/0
20             |   +- 863 -bash
21             |   +-1499 systemctl status
22             |   +-1500 systemctl status
23             +-docker-201c7bd05ea49b654aa8b02a92dbb739a06dd3e8a4cc7
24             | +-1461 /swarm manage -H tcp://172.17.8.101:8333 cons
25             +-swarm-agent.service
26             | +-1437 /bin/bash /home/core/runswarmagent.sh 172.17.
27             | +-1449 /usr/bin/docker run --name swarm-agent --net=
28             +-systemd-journald.service
29             | +-398 /usr/lib/systemd/systemd-journald
30             +-fleet.service
31             | +-918 /usr/bin/fleetd
32             +-systemd-resolved.service
33             | +-554 /usr/lib/systemd/systemd-resolved
34             +-systemd-timesyncd.service
35             | +-476 /usr/lib/systemd/systemd-timesyncd
36             +-swarm-manager.service
37             | +-1405 /bin/bash /home/core/runswarmmanager.sh 172.1
38             | +-1421 /usr/bin/docker run --name swarm-manager --ne
39             +-systemd-logind.service
40             | +-505 /usr/lib/systemd/systemd-logind
41             +-systemd-networkd.service
42             | +-829 /usr/lib/systemd/systemd-networkd
43             +-system-getty.slice
44             | +-getty@tty1.service
45             |   +-498 /sbin/agetty --noclear tty1 linux
46             +-systemd-udev.service
47             | +-425 /usr/lib/systemd/systemd-udev
48             +-consul.service
49             | +-940 /bin/sh -c NUM_SERVERS=$(fleetctl list-machine
50             | +-973 /usr/bin/docker run --name=consul-core-01 -v /
51             +-docker-8323ab8192308e5a65102dfbf109466c6a7c7f43ff28f
52             | +-1371 /bin/consul agent -config-dir=/config -node c
53             +-locksmithd.service
54             | +-1125 /usr/lib/locksmith/locksmithd
55             +-docker.service
56             | +- 877 docker daemon --dns 172.18.0.1 --dns 8.8.8.8
57             | +-1004 docker-proxy -proto tcp -host-ip 172.17.8.101
58             | +-1011 docker-proxy -proto tcp -host-ip 172.17.8.101
59             | +-1027 docker-proxy -proto tcp -host-ip 172.17.8.101
60             | +-1036 docker-proxy -proto tcp -host-ip 172.17.8.101
61             | +-1057 docker-proxy -proto udp -host-ip 172.17.8.101
62             | +-1071 docker-proxy -proto tcp -host-ip 172.17.8.101
63             | +-1089 docker-proxy -proto udp -host-ip 172.17.8.101

```

```

64 | +-1108 docker-proxy -proto tcp -host-ip 172.17.8.101
65 | +-1117 docker-proxy -proto udp -host-ip 172.18.0.1 -
66 +-etcd2.service
67 +-912 /usr/bin/etcd2 -name core-01 -initial-advertis
68 core@core-01 ~ $ docker ps
69 CONTAINER ID          IMAGE                COMMAND                  CR
70 EATED                STATUS              PORTS
71
72
73 NAMES
74 5f41c7d23012          swarm:latest        "/swarm join --addr=1"   Ab
75 out a minute ago    Up About a minute
76
77
78 swarm-agent
79 201c7bd05ea4          swarm:latest        "/swarm manage -H tcp"   Ab
80 out a minute ago    Up About a minute
81
82
83 swarm-manager
84 8323ab819230          progridium/consul    "/bin/start -node cor"   2
85 minutes ago         Up 2 minutes        172.17.8.101:8300->8300
86 -8302/tcp, 172.17.8.101:8400->8400/tcp, 172.17.8.101:8500->8500/tcp
87 , 172.18.0.1:53->53/udp, 172.17.8.101:8600->8600/tcp, 172.17.8.101:
88 8301-8302->8301-8302/udp, 53/tcp    consul-core-01

```

This shows the status of another node in the cluster at a different point in the startup process. It still shows the status of the node, the number of jobs queued and failed units, but there are a lot more services in the process tree. Finally, at line 68, you see how to check on the status of active, running Docker containers.

**Note:** If `docker ps` seems to “hang”, this generally means there is one or more Docker containers trying to get started. Just be patient, and they should show up. To check that the Docker daemon is indeed running, try to run “`docker info`”. It might also hang until whatever activating container starts up, but as long as it doesn’t return immediately with “Cannot connect to the Docker daemon. Is the docker daemon running on this host?”, Docker is working, just be patient.

If `docker ps` doesn’t hang but shows up with just headings and no containers, but you are expecting there to be containers, run `docker ps -a`. This will show all docker containers, even ones that have failed for some reason.

3. `systemd` logs output to its [journal](#). The journal is queried by a tool called `journalctl`. To see all journal output of all `systemd` processes since the node was created, run

```
journalctl
```

This is a lot of output, so it won’t be shown here. Use this tool to see output of all the things in one gigantic set. Particularly useful if you’re trying to see how different services might be affecting each other.

4. To only see journal output for the last boot, run

```
journalctl -b
```

Same type of output as `journalctl`, but only since the last boot.



## 13.2 State of systemd units

All services run on a node with `systemd` are referred to as units. You can check the state of these units individually.

1. Check the status of a unit and get the tail of its log output.

```

1 core@core-01 ~ $ systemctl status consul.service -l
2 ● consul.service - Consul bootstrap
3   Loaded: loaded (/run/systemd/system/consul.service; disabled; ve
4 ndor preset: disabled)
5   Active: active (running) since Wed 2016-01-27 12:41:56 EST; 37mi
6 n ago
7   Process: 941 ExecStartPost=/bin/sh -c /usr/bin/etcdctl set "/serv
8 ices/consul/bootstrap/servers/$COREOS_PUBLIC_IPV4" "$COREOS_PUBLIC_
9 IPV4" (code=exited, status=0/SUCCESS)
10  Process: 932 ExecStartPre=/bin/sh -c /usr/bin/etcdctl mk /service
11 s/consul/bootstrap/host $COREOS_PUBLIC_IPV4 || sleep 10 (code=exite
12 d, status=0/SUCCESS)
13  Process: 926 ExecStartPre=/usr/bin/docker rm consul-%H (code=exit
14 ed, status=0/SUCCESS)
15  Process: 921 ExecStartPre=/usr/bin/docker kill consul-%H (code=ex
16 ited, status=1/FAILURE)
17  Main PID: 940 (sh)
18  Memory: 28.0M
19  CPU: 117ms
20  CGroup: /system.slice/consul.service
21          +-940 /bin/sh -c NUM_SERVERS=$(fleetctl list-machines |
22 grep -v "MACHINE" |wc -l)      && EXPECT=$(if [ $NUM_SERVERS -lt 3
23 ] ; then echo 1; else echo 3; fi)      && JOIN_IP=$(etcdctl ls /s
24 ervices/consul/bootstrap/servers      | grep -v $COREOS_PUBLIC_
25 IPV4      | cut -d '/' -f 6      | head -n 1)      && JOIN
26 =$(if [ "$JOIN_IP" != "" ] ; then sleep 10; echo "-join $JOIN_IP";
27 else echo "-bootstrap-expect $EXPECT"; fi)      && /usr/bin/docker
28 run --name=consul-core-01 -v /mnt:/data      -p 172.17.8.101
29 :8300:8300      -p 172.17.8.101:8301:8301      -p 172.1
30 7.8.101:8301:8301/udp      -p 172.17.8.101:8302:8302
31 -p 172.17.8.101:8302:8302/udp      -p 172.17.8.101:8400:84
32 00      -p 172.17.8.101:8500:8500      -p 172.17.8.101:
33 8600:8600      -p 172.18.0.1:53:53/udp      progridium/co
34 nsul -node core-01 -server -dc=local -advertise 172.17.8.101 $JOIN
35          +-973 /usr/bin/docker run --name=consul-core-01 -v /mnt:
36 /data -p 172.17.8.101:8300:8300 -p 172.17.8.101:8301:8301 -p 172.17
37 .8.101:8301:8301/udp -p 172.17.8.101:8302:8302 -p 172.17.8.101:8302
38 :8302/udp -p 172.17.8.101:8400:8400 -p 172.17.8.101:8500:8500 -p 17
39 2.17.8.101:8600:8600 -p 172.18.0.1:53:53/udp progridium/consul -node
40 core-01 -server -dc=local -advertise 172.17.8.101 -bootstrap-expect
41 1
42
43 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [WARN] raft: R
44 ejecting vote from 172.17.8.103:8300 since our last term is greater
45 (43, 1)
46 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [WARN] raft: H
47 eartbeat timeout reached, starting election
48 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: N
49 ode at 172.17.8.101:8300 [Candidate] entering Candidate state
50 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: E
51 lection won. Tally: 2
52 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: N

```

```

53 ode at 172.17.8.101:8300 [Leader] entering Leader state
54 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] consul:
55   cluster leadership acquired
56 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] consul:
57   New leader elected: core-01
58 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [WARN] raft: A
59 ppendEntries to 172.17.8.103:8300 rejected, sending older logs (nex
60 t: 479)
61 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: p
62 ipelining replication to peer 172.17.8.102:8300
63 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: p
64 ipelining replication to peer 172.17.8.103:8300

```

The `-l` is important as the output will be truncated without it.

This command also shows a multitude of things. It gives you a unit's state as well as from what unit file location a unit is run. Unit files can be placed in multiple locations, and they are run according to a hierarchy, but the file shown by here (line 3) is the one that `systemd` actually runs.

This command also shows the status of any commands used in the stopping or starting of a service (i.e., all the `ExecStart*` or `ExecStop*` directives in a unit file). See lines 9, 12, 14, 16. This is particularly useful if you have `Exec*` directives that could be the cause of a unit failure.

The command run from the `ExecStart` directive is shown, starting at line 20.

Finally, this command gives essentially the tail of the service's journal output. As you can see at line 57, a Consul leader was elected!

## 2. To see the unit file `systemd` runs, run

```

1 core@core-01 ~ $ systemctl cat consul.service
2 # /run/systemd/system/consul.service
3 [Unit]
4 Description=Consul bootstrap
5 Requires=docker.service fleet.service
6 After=docker.service fleet.service
7
8 [Service]
9 EnvironmentFile=/etc/environment
10 TimeoutStartSec=0
11 ExecStartPre=/usr/bin/docker kill consul-%H
12 ExecStartPre=/usr/bin/docker rm consul-%H
13 ExecStartPre=/bin/sh -c "/usr/bin/etcdctl mk /services/consul/boots
14 ExecStart=/bin/sh -c "NUM_SERVERS=$(fleetctl list-machines | grep -
15   && EXPECT=$(if [ $NUM_SERVERS -lt 3 ] ; then echo 1; else echo
16   && JOIN_IP=$(etcdctl ls /services/consul/bootstrap/servers \
17     | grep -v $COREOS_PUBLIC_IPV4 \
18     | cut -d '/' -f 6 \
19     | head -n 1) \
20   && JOIN=$(if [ \"$JOIN_IP\" != \"\" ] ; then sleep 10; echo \"
21   && /usr/bin/docker run --name=consul-%H -v /mnt:/data \
22     -p ${COREOS_PUBLIC_IPV4}:8300:8300 \
23     -p ${COREOS_PUBLIC_IPV4}:8301:8301 \
24     -p ${COREOS_PUBLIC_IPV4}:8301:8301/udp \
25     -p ${COREOS_PUBLIC_IPV4}:8302:8302 \
26     -p ${COREOS_PUBLIC_IPV4}:8302:8302/udp \
27     -p ${COREOS_PUBLIC_IPV4}:8400:8400 \
28     -p ${COREOS_PUBLIC_IPV4}:8500:8500 \
29     -p ${COREOS_PUBLIC_IPV4}:8600:8600 \

```

```

30     -p 172.18.0.1:53:53/udp \
31     progrium/consul -node %H -server -dc=local -advertise ${C
32 ExecStartPost=/bin/sh -c "/usr/bin/etcdctl set \"/services/consul/b
33 ExecStop=/bin/sh -c "/usr/bin/etcdctl rm \"/services/consul/bootstr
34 ExecStop=/bin/sh -c "/usr/bin/etcdctl rm /services/consul/bootstrap
35 ExecStop=/usr/bin/docker stop consul-%H
36 Restart=always
37 RestartSec=10s
38 LimitNOFILE=40000
39
40 [Install]
41 WantedBy=multi-user.target

```

This command shows the service’s unit file directives. It also shows at the top (line 2) the location of the file. In this unit file, there are directives under three headings, “Unit”, “Service”, and “Install”. To learn more about what can go in each of these sections of a unit file, see freedesktop.org’s page on [systemd unit files](#).

3. To make changes to a unit file, run

```
systemctl edit consul.service
```

This will actually create a brand new file to which you can add or override directives to the unit definition. For slightly more information, see DigitalOcean’s [How to Use Systemctl to Manage Systemd Services and Units](#).

4. You can also edit the actual unit file, rather than just creating an override file by running

```
systemctl edit --full consul.service
```

5. `systemd` unit files have many [directives](#) used to configure the units. Some of these are set or have defaults that you may not be aware of. To see a list of the directives for a given unit and what these directives are set to, run

```

1 core@core-01 ~ $ systemctl show consul.service
2 Type=simple
3 Restart=always
4 NotifyAccess=none
5 RestartUsec=10s
6 TimeoutStartUsec=0
7 TimeoutStopUsec=1min 30s
8 WatchdogUsec=0
9 WatchdogTimestamp=Wed 2016-01-27 12:41:56 EST
10 WatchdogTimestampMonotonic=102810100
11 StartLimitInterval=10000000
12 StartLimitBurst=5
13 StartLimitAction=none
14 FailureAction=none
15 PermissionsStartOnly=no
16 RootDirectoryStartOnly=no
17 RemainAfterExit=no
18 GuessMainPID=yes
19 MainPID=940
20 ControlPID=0
21 FileDescriptorStoreMax=0
22 StatusErrno=0
23 Result=success
24 ExecMainStartTimestamp=Wed 2016-01-27 12:41:56 EST
25 ExecMainStartTimestampMonotonic=102810054
26 ExecMainExitTimestampMonotonic=0
27 ExecMainPID=940
28 ExecMainCode=0
29 ExecMainStatus=0

```

```

30 ExecStartPre={ path=/usr/bin/docker ; argv[]=/usr/bin/docker kill c
31 ExecStartPre={ path=/usr/bin/docker ; argv[]=/usr/bin/docker rm con
32 ExecStartPre={ path=/bin/sh ; argv[]=/bin/sh -c /usr/bin/etcdctl mk
33 ExecStart={ path=/bin/sh ; argv[]=/bin/sh -c NUM_SERVERS=$(fleetctl
34 ExecStartPost={ path=/bin/sh ; argv[]=/bin/sh -c /usr/bin/etcdctl s
35 ExecStop={ path=/bin/sh ; argv[]=/bin/sh -c /usr/bin/etcdctl rm "/s
36 ExecStop={ path=/bin/sh ; argv[]=/bin/sh -c /usr/bin/etcdctl rm /se
37 ExecStop={ path=/usr/bin/docker ; argv[]=/usr/bin/docker stop consu
38 Slice=system.slice
39 ControlGroup=/system.slice/consul.service
40 MemoryCurrent=29401088
41 CPUUsageNSec=141291138
42 Delegate=no
43 CPUAccounting=no
44 CPUShares=18446744073709551615
45 StartupCPUShares=18446744073709551615
46 CPUQuotaPerSecUSec=infinity
47 BlockIOAccounting=no
48 BlockIOWeight=18446744073709551615
49 StartupBlockIOWeight=18446744073709551615
50 MemoryAccounting=no
51 MemoryLimit=18446744073709551615
52 DevicePolicy=auto
53 EnvironmentFile=/etc/environment (ignore_errors=no)
54 UMask=0022
55 LimitCPU=18446744073709551615
56 LimitFSIZE=18446744073709551615
57 LimitDATA=18446744073709551615
58 LimitSTACK=18446744073709551615
59 LimitCORE=18446744073709551615
60 LimitRSS=18446744073709551615
61 LimitNOFILE=40000
62 LimitAS=18446744073709551615
63 LimitNPROC=3873
64 LimitMEMLOCK=65536
65 LimitLOCKS=18446744073709551615
66 LimitSIGPENDING=3873
67 LimitMSGQUEUE=819200
68 LimitNICE=0
69 LimitRTPRIO=0
70 LimitRTTIME=18446744073709551615
71 OOMScoreAdjust=0
72 Nice=0
73 IOScheduling=0
74 CPUSchedulingPolicy=0
75 CPUSchedulingPriority=0
76 TimerSlackNSec=50000
77 CPUSchedulingResetOnFork=no
78 NonBlocking=no
79 StandardInput=null
80 StandardOutput=journal
81 StandardError=inherit
82 TTYReset=no
83 TTYVHangup=no
84 TTYVTDisallocate=no
85 SyslogPriority=30
86 SyslogLevelPrefix=yes
87 SecureBits=0

```

```

88 CapabilityBoundingSet=18446744073709551615
89 MountFlags=0
90 PrivateTmp=no
91 PrivateNetwork=no
92 PrivateDevices=no
93 ProtectHome=no
94 ProtectSystem=no
95 SameProcessGroup=no
96 UtmpMode=init
97 IgnoreSIGPIPE=yes
98 NoNewPrivileges=no
99 SystemCallErrorNumber=0
100 RuntimeDirectoryMode=0755
101 KillMode=control-group
102 KillSignal=15
103 SendSIGKILL=yes
104 SendSIGHUP=no
105 Id=consul.service
106 Names=consul.service
107 Requires=basic.target docker.service fleet.service
108 Wants=system.slice
109 RequiredBy=swarm-manager.service
110 Conflicts=shutdown.target
111 Before=shutdown.target swarm-manager.service
112 After=system.slice systemd-journald.socket fleet.service docker.ser
113 Description=Consul bootstrap
114 LoadState=loaded
115 ActiveState=active
116 SubState=running
117 FragmentPath=/run/systemd/system/consul.service
118 UnitFileState=disabled
119 UnitFilePreset=disabled
120 InactiveExitTimestamp=Wed 2016-01-27 12:41:55 EST
121 InactiveExitTimestampMonotonic=102215240
122 ActiveEnterTimestamp=Wed 2016-01-27 12:41:56 EST
123 ActiveEnterTimestampMonotonic=102891180
124 ActiveExitTimestampMonotonic=0
125 InactiveEnterTimestampMonotonic=0
126 CanStart=yes
127 CanStop=yes
128 CanReload=no
129 CanIsolate=no
130 StopWhenUnneeded=no
131 RefuseManualStart=no
132 RefuseManualStop=no
133 AllowIsolate=no
134 DefaultDependencies=yes
135 OnFailureJobMode=replace
136 IgnoreOnIsolate=no
137 IgnoreOnSnapshot=no
138 NeedDaemonReload=no
139 JobTimeoutUSec=0
140 JobTimeoutAction=none
141 ConditionResult=yes
142 AssertResult=yes
143 ConditionTimestamp=Wed 2016-01-27 12:41:55 EST
144 ConditionTimestampMonotonic=102214129
145 AssertTimestamp=Wed 2016-01-27 12:41:55 EST

```

```
146 AssertTimestampMonotonic=102214129
147 Transient=no
```

6. To see all logs of a given unit since the node was created, run

```
journalctl -u consul.service
```

7. Watch the logs of a given unit since the last reboot, run

```
journalctl -b -u consul.service
```

8. Watch the tail of the logs of a unit.

```
journalctl -fu consul.service
```

9. To see logs with explanation texts, run

```
1 core@core-01 ~ $ journalctl -b -x -u consul.service
2 -- Logs begin at Tue 2016-01-26 15:47:27 EST, end at Wed 2016-01-27 13:50:21 EST.
3 Jan 27 12:41:55 core-01 systemd[1]: Starting Consul bootstrap...
4 -- Subject: Unit consul.service has begun start-up
5 -- Defined-By: systemd
6 -- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel
7 --
8 -- Unit consul.service has begun starting up.
9 Jan 27 12:41:56 core-01 docker[921]: Error response from daemon: Cannot kill
10 Jan 27 12:41:56 core-01 docker[921]: Error: failed to kill containers: [consul-
11 Jan 27 12:41:56 core-01 docker[926]: consul-core-01
12 Jan 27 12:41:56 core-01 sh[932]: 172.17.8.101
13 Jan 27 12:41:56 core-01 sh[940]: Error retrieving list of active machines:
14 Jan 27 12:41:56 core-01 sh[941]: 172.17.8.101
15 Jan 27 12:41:56 core-01 systemd[1]: Started Consul bootstrap.
16 -- Subject: Unit consul.service has finished start-up
17 -- Defined-By: systemd
18 -- Support: http://lists.freedesktop.org/mailman/listinfo/systemd-devel
19 --
20 -- Unit consul.service has finished starting up.
21 --
22 -- The start-up result is done.
23 Jan 27 12:42:39 core-01 sh[940]: ==> WARNING: BootstrapExpect Mode is specified
24 Jan 27 12:42:39 core-01 sh[940]: ==> WARNING: Bootstrap mode enabled! Do not
25 Jan 27 12:42:39 core-01 sh[940]: ==> WARNING: It is highly recommended to set
26 Jan 27 12:42:39 core-01 sh[940]: ==> Starting raft data migration...
27 Jan 27 12:42:39 core-01 sh[940]: ==> Starting Consul agent...
28 Jan 27 12:42:39 core-01 sh[940]: ==> Starting Consul agent RPC...
29 Jan 27 12:42:39 core-01 sh[940]: Consul agent running!
30 Jan 27 12:42:39 core-01 sh[940]: Node name: 'core-01'
31 Jan 27 12:42:39 core-01 sh[940]: Datacenter: 'local'
32 Jan 27 12:42:39 core-01 sh[940]: Server: true (bootstrap: true)
33 Jan 27 12:42:39 core-01 sh[940]: Client Addr: 0.0.0.0 (HTTP: 8500, HTTPS: -1,
34 Jan 27 12:42:39 core-01 sh[940]: Cluster Addr: 172.17.8.101 (LAN: 8301, WAN: 8302)
35 Jan 27 12:42:39 core-01 sh[940]: Gossip encrypt: false, RPC-TLS: false, TLS-
Incoming: false
```

```

36 Jan 27 12:42:39 core-01 sh[940]: Atlas: <disabled>
37 Jan 27 12:42:39 core-01 sh[940]: ==> Log data will now stream in as it occurs:
38 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [INFO] serf:
    ↳EventMemberJoin: core-01 172.17.8.101
39 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [INFO] serf:
    ↳EventMemberJoin: core-01.local 172.17.8.101
40 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [INFO] raft: Node at 172.17.
    ↳8.101:8300 [Follower] entering Follower state
41 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [WARN] serf: Failed to re-
    ↳join any previously known node
42 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [WARN] serf: Failed to re-
    ↳join any previously known node
43 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [INFO] consul: adding server
    ↳core-01 (Addr: 172.17.8.101:8300) (DC: local)
44 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [INFO] consul: adding server
    ↳core-01.local (Addr: 172.17.8.101:8300) (DC: loc
45 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [ERR] agent: failed to sync
    ↳remote state: No cluster leader
46 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [ERR] http: Request /v1/kv/
    ↳docker/nodes/172.19.0.1:2376, error: No cluster le
47 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [ERR] http: Request /v1/kv/
    ↳docker/nodes/172.19.0.1:2376, error: No cluster le
48 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [INFO] serf:
    ↳EventMemberJoin: core-02 172.17.8.102
49 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [INFO] consul: adding server
    ↳core-02 (Addr: 172.17.8.102:8300) (DC: local)
50 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [ERR] http: Request /v1/kv/
    ↳docker/nodes/172.19.0.1:2376, error: No cluster le
51 Jan 27 12:42:39 core-01 sh[940]: 2016/01/27 17:42:39 [ERR] http: Request /v1/kv/
    ↳docker/nodes/172.19.0.1:2376, error: No cluster le
52 Jan 27 12:42:40 core-01 sh[940]: 2016/01/27 17:42:40 [WARN] raft: Heartbeat
    ↳timeout reached, starting election
53 Jan 27 12:42:40 core-01 sh[940]: 2016/01/27 17:42:40 [INFO] raft: Node at 172.17.
    ↳8.101:8300 [Candidate] entering Candidate state
54 Jan 27 12:42:40 core-01 sh[940]: 2016/01/27 17:42:40 [ERR] raft: Failed to make
    ↳RequestVote RPC to 172.17.8.103:8300: dial tcp 172
55 Jan 27 12:42:40 core-01 sh[940]: 2016/01/27 17:42:40 [INFO] raft: Election won.
    ↳Tally: 2
56 Jan 27 12:42:40 core-01 sh[940]: 2016/01/27 17:42:40 [INFO] raft: Node at 172.17.
    ↳8.101:8300 [Leader] entering Leader state
57 ...skipping...
58 Jan 27 12:42:41 core-01 sh[940]: 2016/01/27 17:42:41 [ERR] raft: Failed to
    ↳AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
59 Jan 27 12:42:41 core-01 sh[940]: 2016/01/27 17:42:41 [ERR] raft: Failed to
    ↳heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
60 Jan 27 12:42:41 core-01 sh[940]: 2016/01/27 17:42:41 [ERR] raft: Failed to
    ↳AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
61 Jan 27 12:42:41 core-01 sh[940]: 2016/01/27 17:42:41 [WARN] raft: Failed to
    ↳contact 172.17.8.103:8300 in 509.786599ms
62 Jan 27 12:42:41 core-01 sh[940]: 2016/01/27 17:42:41 [ERR] raft: Failed to
    ↳heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
63 Jan 27 12:42:41 core-01 sh[940]: 2016/01/27 17:42:41 [ERR] raft: Failed to
    ↳heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
64 Jan 27 12:42:41 core-01 sh[940]: 2016/01/27 17:42:41 [ERR] raft: Failed to
    ↳AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
65 Jan 27 12:42:41 core-01 sh[940]: 2016/01/27 17:42:41 [ERR] raft: Failed to
    ↳heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
66 Jan 27 12:42:41 core-01 sh[940]: 2016/01/27 17:42:41 [WARN] raft: Failed to
    ↳contact 172.17.8.103:8300 in 981.100031ms

```

```

67 Jan 27 12:42:42 core-01 sh[940]: 2016/01/27 17:42:42 [ERR] raft: Failed to_
    ↳ AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
68 Jan 27 12:42:42 core-01 sh[940]: 2016/01/27 17:42:42 [ERR] raft: Failed to_
    ↳ heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
69 Jan 27 12:42:42 core-01 sh[940]: 2016/01/27 17:42:42 [WARN] raft: Failed to_
    ↳ contact 172.17.8.103:8300 in 1.480625817s
70 Jan 27 12:42:42 core-01 sh[940]: 2016/01/27 17:42:42 [ERR] raft: Failed to_
    ↳ heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
71 Jan 27 12:42:42 core-01 sh[940]: 2016/01/27 17:42:42 [ERR] raft: Failed to_
    ↳ AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
72 Jan 27 12:42:43 core-01 sh[940]: 2016/01/27 17:42:43 [ERR] raft: Failed to_
    ↳ heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
73 Jan 27 12:42:44 core-01 sh[940]: 2016/01/27 17:42:44 [ERR] raft: Failed to_
    ↳ AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
74 Jan 27 12:42:44 core-01 sh[940]: 2016/01/27 17:42:44 [ERR] raft: Failed to_
    ↳ heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
75 Jan 27 12:42:46 core-01 sh[940]: 2016/01/27 17:42:46 [ERR] raft: Failed to_
    ↳ AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
76 Jan 27 12:42:47 core-01 sh[940]: 2016/01/27 17:42:47 [ERR] raft: Failed to_
    ↳ heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
77 Jan 27 12:42:51 core-01 sh[940]: 2016/01/27 17:42:51 [ERR] raft: Failed to_
    ↳ AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
78 Jan 27 12:42:52 core-01 sh[940]: 2016/01/27 17:42:52 [ERR] raft: Failed to_
    ↳ heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
79 Jan 27 12:43:02 core-01 sh[940]: 2016/01/27 17:43:02 [ERR] raft: Failed to_
    ↳ AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
80 Jan 27 12:43:05 core-01 sh[940]: 2016/01/27 17:43:05 [ERR] raft: Failed to_
    ↳ heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
81 Jan 27 12:43:14 core-01 sh[940]: 2016/01/27 17:43:14 [ERR] raft: Failed to_
    ↳ AppendEntries to 172.17.8.103:8300: dial tcp 172.17.8.1
82 Jan 27 12:43:17 core-01 sh[940]: 2016/01/27 17:43:17 [ERR] raft: Failed to_
    ↳ heartbeat to 172.17.8.103:8300: dial tcp 172.17.8.103:8
83 Jan 27 12:43:23 core-01 sh[940]: 2016/01/27 17:43:23 [INFO] serf:_
    ↳ EventMemberJoin: core-03 172.17.8.103
84 Jan 27 12:43:23 core-01 sh[940]: 2016/01/27 17:43:23 [INFO] consul: adding server_
    ↳ core-03 (Addr: 172.17.8.103:8300) (DC: local)
85 Jan 27 12:43:23 core-01 sh[940]: 2016/01/27 17:43:23 [INFO] consul: member 'core-
    ↳ 03' joined, marking health alive
86 Jan 27 12:43:24 core-01 sh[940]: 2016/01/27 17:43:24 [WARN] raft: AppendEntries_
    ↳ to 172.17.8.103:8300 rejected, sending older logs
87 Jan 27 12:43:24 core-01 sh[940]: 2016/01/27 17:43:24 [WARN] raft: Rejecting vote_
    ↳ from 172.17.8.103:8300 since we have a leader: 17
88 Jan 27 12:43:24 core-01 sh[940]: 2016/01/27 17:43:24 [WARN] raft: Failed to_
    ↳ contact 172.17.8.103:8300 in 500.297851ms
89 Jan 27 12:43:25 core-01 sh[940]: 2016/01/27 17:43:25 [WARN] raft: Failed to_
    ↳ contact 172.17.8.103:8300 in 938.153601ms
90 Jan 27 12:43:25 core-01 sh[940]: 2016/01/27 17:43:25 [WARN] raft: Rejecting vote_
    ↳ from 172.17.8.103:8300 since we have a leader: 17
91 Jan 27 12:43:25 core-01 sh[940]: 2016/01/27 17:43:25 [WARN] raft: Failed to_
    ↳ contact 172.17.8.103:8300 in 1.424666193s
92 Jan 27 12:43:27 core-01 sh[940]: 2016/01/27 17:43:27 [WARN] raft: Rejecting vote_
    ↳ from 172.17.8.103:8300 since we have a leader: 17
93 Jan 27 12:43:28 core-01 sh[940]: 2016/01/27 17:43:28 [WARN] raft: Rejecting vote_
    ↳ from 172.17.8.103:8300 since we have a leader: 17
94 Jan 27 12:43:30 core-01 sh[940]: 2016/01/27 17:43:30 [WARN] raft: Rejecting vote_
    ↳ from 172.17.8.103:8300 since we have a leader: 17
95 Jan 27 12:43:31 core-01 sh[940]: 2016/01/27 17:43:31 [WARN] raft: Rejecting vote_
    ↳ from 172.17.8.103:8300 since we have a leader: 17

```



```

96 Jan 27 12:43:33 core-01 sh[940]: 2016/01/27 17:43:33 [WARN] raft: Rejecting vote_
    ↳from 172.17.8.103:8300 since we have a leader: 17
97 Jan 27 12:43:34 core-01 sh[940]: 2016/01/27 17:43:34 [WARN] raft: Rejecting vote_
    ↳from 172.17.8.103:8300 since we have a leader: 17
98 Jan 27 12:43:34 core-01 sh[940]: 2016/01/27 17:43:34 [ERR] raft: peer 172.17.8.
    ↳103:8300 has newer term, stopping replication
99 Jan 27 12:43:34 core-01 sh[940]: 2016/01/27 17:43:34 [INFO] raft: Node at 172.17.
    ↳8.101:8300 [Follower] entering Follower state
100 Jan 27 12:43:34 core-01 sh[940]: 2016/01/27 17:43:34 [INFO] consul: cluster_
    ↳leadership lost
101 Jan 27 12:43:34 core-01 sh[940]: 2016/01/27 17:43:34 [INFO] raft: aborting_
    ↳pipeline replication to peer 172.17.8.102:8300
102 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [WARN] raft: Rejecting vote_
    ↳from 172.17.8.103:8300 since our last term is gre
103 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [WARN] raft: Heartbeat_
    ↳timeout reached, starting election
104 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: Node at 172.17.
    ↳8.101:8300 [Candidate] entering Candidate state
105 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: Election won._
    ↳Tally: 2
106 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: Node at 172.17.
    ↳8.101:8300 [Leader] entering Leader state
107 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] consul: cluster_
    ↳leadership acquired
108 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] consul: New leader_
    ↳elected: core-01
109 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [WARN] raft: AppendEntries_
    ↳to 172.17.8.103:8300 rejected, sending older logs
110 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: pipelining_
    ↳replication to peer 172.17.8.102:8300
111 Jan 27 12:43:35 core-01 sh[940]: 2016/01/27 17:43:35 [INFO] raft: pipelining_
    ↳replication to peer 172.17.8.103:8300
112 Jan 27 13:30:47 core-01 sh[940]: 2016/01/27 18:30:47 [INFO] agent.rpc: Accepted_
    ↳client: 127.0.0.1:44510

```

Line 2 says what the date/time range of possible logs exist, but as you can see in line 3, the first log in this set is not a Jan 26 date, as could be possible according to line 2, but a Jan 27 date, which is the last time this node was rebooted.

This service started up just fine, so there's no failures to point out, but this is where you'd find them and any possible explanation for those failures.

10. If the unit you are running is running a Docker container, all relevant and helpful information may not be available to you via `journalctl`. To see logs from the Docker container itself, run

```

1 core@core-01 ~ $ docker logs consul-core-01
2 ==> WARNING: BootstrapExpect Mode is specified as 1; this is the sa
3 me as Bootstrap mode.
4 ==> WARNING: Bootstrap mode enabled! Do not enable unless necessary
5 ==> WARNING: It is highly recommended to set GOMAXPROCS higher than
6 1
7 ==> Starting raft data migration...
8 ==> Starting Consul agent...
9 ==> Starting Consul agent RPC...
10 ==> Consul agent running!
11     Node name: 'core-01'
12     Datacenter: 'local'
13     Server: true (bootstrap: true)

```

```

14      Client Addr: 0.0.0.0 (HTTP: 8500, HTTPS: -1, DNS: 53, RPC: 8
15 400)
16      Cluster Addr: 172.17.8.101 (LAN: 8301, WAN: 8302)
17      Gossip encrypt: false, RPC-TLS: false, TLS-Incoming: false
18      Atlas: <disabled>
19
20 ==> Log data will now stream in as it occurs:
21
22      2016/01/27 17:42:39 [INFO] serf: EventMemberJoin: core-01 172.1
23 7.8.101
24      2016/01/27 17:42:39 [INFO] serf: EventMemberJoin: core-01.local
25 172.17.8.101
26      2016/01/27 17:42:39 [INFO] raft: Node at 172.17.8.101:8300 [Fol
27 lower] entering Follower state
28      2016/01/27 17:42:39 [WARN] serf: Failed to re-join any previous
29 ly known node
30      2016/01/27 17:42:39 [WARN] serf: Failed to re-join any previous
31 ly known node
32      2016/01/27 17:42:39 [INFO] consul: adding server core-01 (Addr:
33 172.17.8.101:8300) (DC: local)
34      2016/01/27 17:42:39 [INFO] consul: adding server core-01.local
35 (Addr: 172.17.8.101:8300) (DC: local)
36      2016/01/27 17:42:39 [ERR] agent: failed to sync remote state: N
37 o cluster leader
38      2016/01/27 17:42:39 [ERR] http: Request /v1/kv/docker/nodes/172
39 .19.0.1:2376, error: No cluster leader
40      2016/01/27 17:42:39 [ERR] http: Request /v1/kv/docker/nodes/172
41 .19.0.1:2376, error: No cluster leader
42      2016/01/27 17:42:39 [INFO] serf: EventMemberJoin: core-02 172.1
43 7.8.102
44      2016/01/27 17:42:39 [INFO] consul: adding server core-02 (Addr:
45 172.17.8.102:8300) (DC: local)
46      2016/01/27 17:42:39 [ERR] http: Request /v1/kv/docker/nodes/172
47 .19.0.1:2376, error: No cluster leader
48      2016/01/27 17:42:39 [ERR] http: Request /v1/kv/docker/nodes/172
49 .19.0.1:2376, error: No cluster leader
50      2016/01/27 17:42:40 [WARN] raft: Heartbeat timeout reached, sta
51 rting election
52      2016/01/27 17:42:40 [INFO] raft: Node at 172.17.8.101:8300 [Can
53 didate] entering Candidate state
54      2016/01/27 17:42:40 [ERR] raft: Failed to make RequestVote RPC
55 to 172.17.8.103:8300: dial tcp 172.17.8.103:8300: connection refuse
56 d
57      2016/01/27 17:42:40 [INFO] raft: Election won. Tally: 2
58      2016/01/27 17:42:40 [INFO] raft: Node at 172.17.8.101:8300 [Lea
59 der] entering Leader state
60      2016/01/27 17:42:40 [INFO] consul: cluster leadership acquired
61      2016/01/27 17:42:40 [INFO] consul: New leader elected: core-01
62      2016/01/27 17:42:40 [INFO] raft: Disabling EnableSingleNode (bo
63 otstrap)
64      2016/01/27 17:42:40 [ERR] raft: Failed to AppendEntries to 172.
65 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
66      2016/01/27 17:42:40 [INFO] raft: pipelining replication to peer
67 172.17.8.102:8300
68      2016/01/27 17:42:40 [ERR] raft: Failed to AppendEntries to 172.
69 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
70      2016/01/27 17:42:40 [INFO] consul: member 'core-03' reaped, der
71 egistering

```

```

72 2016/01/27 17:42:41 [ERR] raft: Failed to AppendEntries to 172.
73 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
74 2016/01/27 17:42:41 [ERR] raft: Failed to heartbeat to 172.17.8
75 .103:8300: dial tcp 172.17.8.103:8300: connection refused
76 2016/01/27 17:42:41 [ERR] raft: Failed to AppendEntries to 172.
77 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
78 2016/01/27 17:42:41 [ERR] raft: Failed to heartbeat to 172.17.8
79 .103:8300: dial tcp 172.17.8.103:8300: connection refused
80 2016/01/27 17:42:41 [ERR] raft: Failed to AppendEntries to 172.
81 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
82 2016/01/27 17:42:41 [ERR] raft: Failed to heartbeat to 172.17.8
83 .103:8300: dial tcp 172.17.8.103:8300: connection refused
84 2016/01/27 17:42:41 [ERR] raft: Failed to AppendEntries to 172.
85 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
86 2016/01/27 17:42:41 [WARN] raft: Failed to contact 172.17.8.103
87 :8300 in 509.786599ms
88 2016/01/27 17:42:41 [ERR] raft: Failed to heartbeat to 172.17.8
89 .103:8300: dial tcp 172.17.8.103:8300: connection refused
90 2016/01/27 17:42:41 [ERR] raft: Failed to heartbeat to 172.17.8
91 .103:8300: dial tcp 172.17.8.103:8300: connection refused
92 2016/01/27 17:42:41 [ERR] raft: Failed to AppendEntries to 172.
93 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
94 2016/01/27 17:42:41 [ERR] raft: Failed to heartbeat to 172.17.8
95 .103:8300: dial tcp 172.17.8.103:8300: connection refused
96 2016/01/27 17:42:41 [WARN] raft: Failed to contact 172.17.8.103
97 :8300 in 981.100031ms
98 2016/01/27 17:42:42 [ERR] raft: Failed to AppendEntries to 172.
99 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
100 2016/01/27 17:42:42 [ERR] raft: Failed to heartbeat to 172.17.8
101 .103:8300: dial tcp 172.17.8.103:8300: connection refused
102 2016/01/27 17:42:42 [WARN] raft: Failed to contact 172.17.8.103
103 :8300 in 1.480625817s
104 2016/01/27 17:42:42 [ERR] raft: Failed to heartbeat to 172.17.8
105 .103:8300: dial tcp 172.17.8.103:8300: connection refused
106 2016/01/27 17:42:42 [ERR] raft: Failed to AppendEntries to 172.
107 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
108 2016/01/27 17:42:43 [ERR] raft: Failed to heartbeat to 172.17.8
109 .103:8300: dial tcp 172.17.8.103:8300: connection refused
110 2016/01/27 17:42:44 [ERR] raft: Failed to AppendEntries to 172.
111 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
112 2016/01/27 17:42:44 [ERR] raft: Failed to heartbeat to 172.17.8
113 .103:8300: dial tcp 172.17.8.103:8300: connection refused
114 2016/01/27 17:42:46 [ERR] raft: Failed to AppendEntries to 172.
115 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
116 2016/01/27 17:42:47 [ERR] raft: Failed to heartbeat to 172.17.8
117 .103:8300: dial tcp 172.17.8.103:8300: connection refused
118 2016/01/27 17:42:51 [ERR] raft: Failed to AppendEntries to 172.
119 17.8.103:8300: dial tcp 172.17.8.103:8300: connection refused
120 2016/01/27 17:42:52 [ERR] raft: Failed to heartbeat to 172.17.8
121 .103:8300: dial tcp 172.17.8.103:8300: connection refused
122 2016/01/27 17:43:02 [ERR] raft: Failed to AppendEntries to 172.
123 17.8.103:8300: dial tcp 172.17.8.103:8300: no route to host
124 2016/01/27 17:43:05 [ERR] raft: Failed to heartbeat to 172.17.8
125 .103:8300: dial tcp 172.17.8.103:8300: no route to host
126 2016/01/27 17:43:14 [ERR] raft: Failed to AppendEntries to 172.
127 17.8.103:8300: dial tcp 172.17.8.103:8300: no route to host
128 2016/01/27 17:43:17 [ERR] raft: Failed to heartbeat to 172.17.8
129 .103:8300: dial tcp 172.17.8.103:8300: no route to host

```

```

130     2016/01/27 17:43:23 [INFO] serf: EventMemberJoin: core-03 172.1
131 7.8.103
132     2016/01/27 17:43:23 [INFO] consul: adding server core-03 (Addr:
133 172.17.8.103:8300) (DC: local)
134     2016/01/27 17:43:23 [INFO] consul: member 'core-03' joined, mar
135 king health alive
136     2016/01/27 17:43:24 [WARN] raft: AppendEntries to 172.17.8.103:
137 8300 rejected, sending older logs (next: 479)
138     2016/01/27 17:43:24 [WARN] raft: Rejecting vote from 172.17.8.1
139 03:8300 since we have a leader: 172.17.8.101:8300
140     2016/01/27 17:43:24 [WARN] raft: Failed to contact 172.17.8.103
141 :8300 in 500.297851ms
142     2016/01/27 17:43:25 [WARN] raft: Failed to contact 172.17.8.103
143 :8300 in 938.153601ms
144     2016/01/27 17:43:25 [WARN] raft: Rejecting vote from 172.17.8.1
145 03:8300 since we have a leader: 172.17.8.101:8300
146     2016/01/27 17:43:25 [WARN] raft: Failed to contact 172.17.8.103
147 :8300 in 1.424666193s
148     2016/01/27 17:43:27 [WARN] raft: Rejecting vote from 172.17.8.1
149 03:8300 since we have a leader: 172.17.8.101:8300
150     2016/01/27 17:43:28 [WARN] raft: Rejecting vote from 172.17.8.1
151 03:8300 since we have a leader: 172.17.8.101:8300
152     2016/01/27 17:43:30 [WARN] raft: Rejecting vote from 172.17.8.1
153 03:8300 since we have a leader: 172.17.8.101:8300
154     2016/01/27 17:43:31 [WARN] raft: Rejecting vote from 172.17.8.1
155 03:8300 since we have a leader: 172.17.8.101:8300
156     2016/01/27 17:43:33 [WARN] raft: Rejecting vote from 172.17.8.1
157 03:8300 since we have a leader: 172.17.8.101:8300
158     2016/01/27 17:43:34 [WARN] raft: Rejecting vote from 172.17.8.1
159 03:8300 since we have a leader: 172.17.8.101:8300
160     2016/01/27 17:43:34 [ERR] raft: peer 172.17.8.103:8300 has newe
161 r term, stopping replication
162     2016/01/27 17:43:34 [INFO] raft: Node at 172.17.8.101:8300 [Fol
163 lower] entering Follower state
164     2016/01/27 17:43:34 [INFO] consul: cluster leadership lost
165     2016/01/27 17:43:34 [INFO] raft: aborting pipeline replication
166 to peer 172.17.8.102:8300
167     2016/01/27 17:43:35 [WARN] raft: Rejecting vote from 172.17.8.1
168 03:8300 since our last term is greater (43, 1)
169     2016/01/27 17:43:35 [WARN] raft: Heartbeat timeout reached, sta
170 rting election
171     2016/01/27 17:43:35 [INFO] raft: Node at 172.17.8.101:8300 [Can
172 didate] entering Candidate state
173     2016/01/27 17:43:35 [INFO] raft: Election won. Tally: 2
174     2016/01/27 17:43:35 [INFO] raft: Node at 172.17.8.101:8300 [Lea
175 der] entering Leader state
176     2016/01/27 17:43:35 [INFO] consul: cluster leadership acquired
177     2016/01/27 17:43:35 [INFO] consul: New leader elected: core-01
178     2016/01/27 17:43:35 [WARN] raft: AppendEntries to 172.17.8.103:
179 8300 rejected, sending older logs (next: 479)
180     2016/01/27 17:43:35 [INFO] raft: pipelining replication to peer
181 172.17.8.102:8300
182     2016/01/27 17:43:35 [INFO] raft: pipelining replication to peer
183 172.17.8.103:8300
184     2016/01/27 18:30:47 [INFO] agent.rpc: Accepted client: 127.0.0.
185 1:44510

```

This is generally the same output what you can get from `journalctl`, but I think I have found other informa-

tion in the docker logs than `journalctl` by itself.

---

**Note:** The name of the `systemd` service and the name of the Docker container might NOT be the same. They *can* be the same. However, if, as in this example, you name your service “foo” so the service is “foo.service”, and you name your Docker container “foo-`$hostname`”, running `docker logs foo.service` or `docker logs foo` will not work. Don’t get upset with Docker when it tells you there’s no such container “foo.service” when you named a container “foo-`$hostname`”. :)

---

11. To follow the logs in real time, run

```
docker logs -f consul-core-01
```

## 13.3 Managing systemd units

1. You can start, stop, restart, and reload units with

```
sudo systemctl {start|stop|reload|restart} consul.service
```

You must run with `sudo`.

The “reload” option works for units which can reload their configurations without restarting.

2. When you make changes to a unit and are going to restart that unit, first you must let the system daemon know that changes are happening:

```
sudo systemctl daemon-reload
```

**Warning:** This may seem obvious, but it’s a good thing to remember: if a `systemd` unit is running a Docker container, if you restart the unit, this doesn’t necessarily mean the Docker container gets removed and you get a new container when the unit is restarted.



---

## Managing Virtualbox VMs

---

This chapter covers using Virtualbox command line tools, most importantly `VBoxManage`, to manage core DIMS virtual machines.

---

**Note:** See also the descriptions of `dimsasbuilt:wellington` and `dimsasbuilt:stirling` in `dimsasbuilt:dimsasbuilt`.

---

### 14.1 Remotely Managing Virtualbox

Virtualbox can be managed remotely using X11 (“X Window System”) clients like those in [virt tools](#). From a system running an X11 server, you can use SSH with:

- [How to forward X over SSH from Ubuntu machine?](#)
- [Use the virt-manager X11 GUI](#)
- [Use virt-install and connect by using a local VNC client](#)
- [virt-manager won't release the mouse when using ssh forwarding from OS X](#)

```
[root@wellington ~]# VBoxManage list runningvms
"vpn" {4f6ed378-8a9d-4c69-a380-2c194bc4eae0}
"fowiki" {8978f52d-1251-4fea-a3d7-8d9a0950bad1}
"lapp" {511b9f91-9323-476e-baf3-9bc64f97511e}
"jira" {c873db45-b81a-47fe-a5e3-6bdfe96b0dea}
"jenkins" {28e023eb-f4c4-40f5-b4e8-d37cfafde3be}
"linda-vm1" {df5fdc5e-d508-4007-9f5d-84a000a2b5c5}
"sso" {3916fa49-d251-4ced-9275-c8757aceaf66}
"u12-dev-ws-1" {9f58eca0-b3a6-451e-9b2b-f458c75d6869}
"u12-dev-svr-1" {cc1fef3-61f4-4d67-b767-1f4add8f760a}
"hub" {4b530a22-df34-4fd2-89df-2e0a5844b397}
```

```
[lparsons@wellington ~]$ vboxmanage list bridgedifs
Name:          em1
GUID:          00316d65-0000-4000-8000-f04da240a9e1
DHCP:          Disabled
IPAddress:     172.28.234.234
NetworkMask:   255.255.255.0
IPv6Address:   fe80:0000:0000:0000:f24d:a2ff:fe40:a9e1
IPv6NetworkMaskPrefixLength: 64
HardwareAddress: f0:4d:a2:40:a9:e1
MediumType:    Ethernet
Status:        Up
VBoxNetworkName: HostInterfaceNetworking-em1

Name:          em2
GUID:          00326d65-0000-4000-8000-f04da240a9e3
DHCP:          Disabled
IPAddress:     0.0.0.0
NetworkMask:   0.0.0.0
IPv6Address:
IPv6NetworkMaskPrefixLength: 0
HardwareAddress: f0:4d:a2:40:a9:e3
MediumType:    Ethernet
Status:        Down
VBoxNetworkName: HostInterfaceNetworking-em2

Name:          em3
GUID:          00336d65-0000-4000-8000-f04da240a9e5
DHCP:          Disabled
IPAddress:     0.0.0.0
NetworkMask:   0.0.0.0
IPv6Address:
IPv6NetworkMaskPrefixLength: 0
HardwareAddress: f0:4d:a2:40:a9:e5
MediumType:    Ethernet
Status:        Down
VBoxNetworkName: HostInterfaceNetworking-em3

Name:          em4
GUID:          00346d65-0000-4000-8000-f04da240a9e7
DHCP:          Disabled
IPAddress:     10.11.11.1
NetworkMask:   255.255.255.0
IPv6Address:   fe80:0000:0000:0000:f24d:a2ff:fe40:a9e7
IPv6NetworkMaskPrefixLength: 64
HardwareAddress: f0:4d:a2:40:a9:e7
MediumType:    Ethernet
Status:        Up
VBoxNetworkName: HostInterfaceNetworking-em4
```

```
[lparsons@wellington ~]$ vboxmanage list hostonlyifs
Name:          vboxnet0
GUID:          786f6276-656e-4074-8000-0a0027000000
DHCP:          Disabled
IPAddress:     192.168.88.0
NetworkMask:   255.255.255.0
IPv6Address:   fe80:0000:0000:0000:0800:27ff:fe00:0000
IPv6NetworkMaskPrefixLength: 64
HardwareAddress: 0a:00:27:00:00:00
```



```

MediumType:      Ethernet
Status:          Up
VBoxNetworkName: HostInterfaceNetworking-vboxnet0

Name:            vboxnet1
GUID:            786f6276-656e-4174-8000-0a0027000001
DHCP:            Disabled
IPAddress:       192.168.57.1
NetworkMask:     255.255.255.0
IPv6Address:
IPv6NetworkMaskPrefixLength: 0
HardwareAddress: 0a:00:27:00:00:01
MediumType:      Ethernet
Status:          Down
VBoxNetworkName: HostInterfaceNetworking-vboxnet1

Name:            vboxnet2
GUID:            786f6276-656e-4274-8000-0a0027000002
DHCP:            Disabled
IPAddress:       192.168.58.1
NetworkMask:     255.255.255.0
IPv6Address:
IPv6NetworkMaskPrefixLength: 0
HardwareAddress: 0a:00:27:00:00:02
MediumType:      Ethernet
Status:          Down
VBoxNetworkName: HostInterfaceNetworking-vboxnet2

Name:            vboxnet3
GUID:            786f6276-656e-4374-8000-0a0027000003
DHCP:            Disabled
IPAddress:       172.17.8.1
NetworkMask:     255.255.255.0
IPv6Address:     fe80:0000:0000:0000:0800:27ff:fe00:0003
IPv6NetworkMaskPrefixLength: 64
HardwareAddress: 0a:00:27:00:00:03
MediumType:      Ethernet
Status:          Up
VBoxNetworkName: HostInterfaceNetworking-vboxnet3

```

```

[lparkers@wellington ~]$ sudo vboxmanage list dhcpservers
NetworkName:      HostInterfaceNetworking-vboxnet0
IP:               192.168.88.100
NetworkMask:      255.255.255.0
lowerIPAddress:   192.168.88.102
upperIPAddress:   192.168.88.254
Enabled:          Yes

NetworkName:      HostInterfaceNetworking-vboxnet2
IP:               0.0.0.0
NetworkMask:      0.0.0.0
lowerIPAddress:   0.0.0.0
upperIPAddress:   0.0.0.0
Enabled:          No

NetworkName:      HostInterfaceNetworking-vboxnet1
IP:               0.0.0.0
NetworkMask:      0.0.0.0

```

```
lowerIPAddress: 0.0.0.0  
upperIPAddress: 0.0.0.0  
Enabled:       No
```

<http://superuser.com/questions/375316/closing-gui-session-while-running-virtual-machine-virtual-box>

### 15.1 Add New Connection to Apache Directory Studio

---

**Note:** These instructions are based on contents from this original DIMS project [FosWiki Provision New Users](#) page.

---

---

**Note:** We are in the process of moving to a “split-horizon DNS” configuration using the subdomains `ops.develop` and/or `devops.develop` as opposed to the original monolithic domain `prisem.washington.edu` that was being overlayed with both routable and non-routable IP address mappings. As a result, some configuration using the original `prisem.washington.edu` domain remains, such as the **DN** entry information shown below.

---

If you have never connected to our LDAP before, you will need to add the connection to Apache Directory Studio (`apache-directory-studio`). You can see your saved connections in the Connections tab. To add a new connection, do the following:

1. On the LDAP menu, select **New Connection**. The **Network Parameter** dialog will display.
  - (a) Enter a name for the connection. Use `ldap.devops.develop`
  - (b) Enter hostname: `ldap.devops.develop`
  - (c) Port should be 389
  - (d) No encryption
2. You can click **Check Network Parameter** to check the connection
3. Click **Next**. The **Authentication** dialog will display.
  - (a) Leave **Authentication Method** as **Simple Authentication**
  - (b) Bind DN or user: `cn=admin,dc=prisem,dc=washington,dc=edu`
  - (c) Bind password: [See the [FosWiki Provision New Users](#) page for password.]
  - (d) Click the checkbox to save the password if it is not already checked.

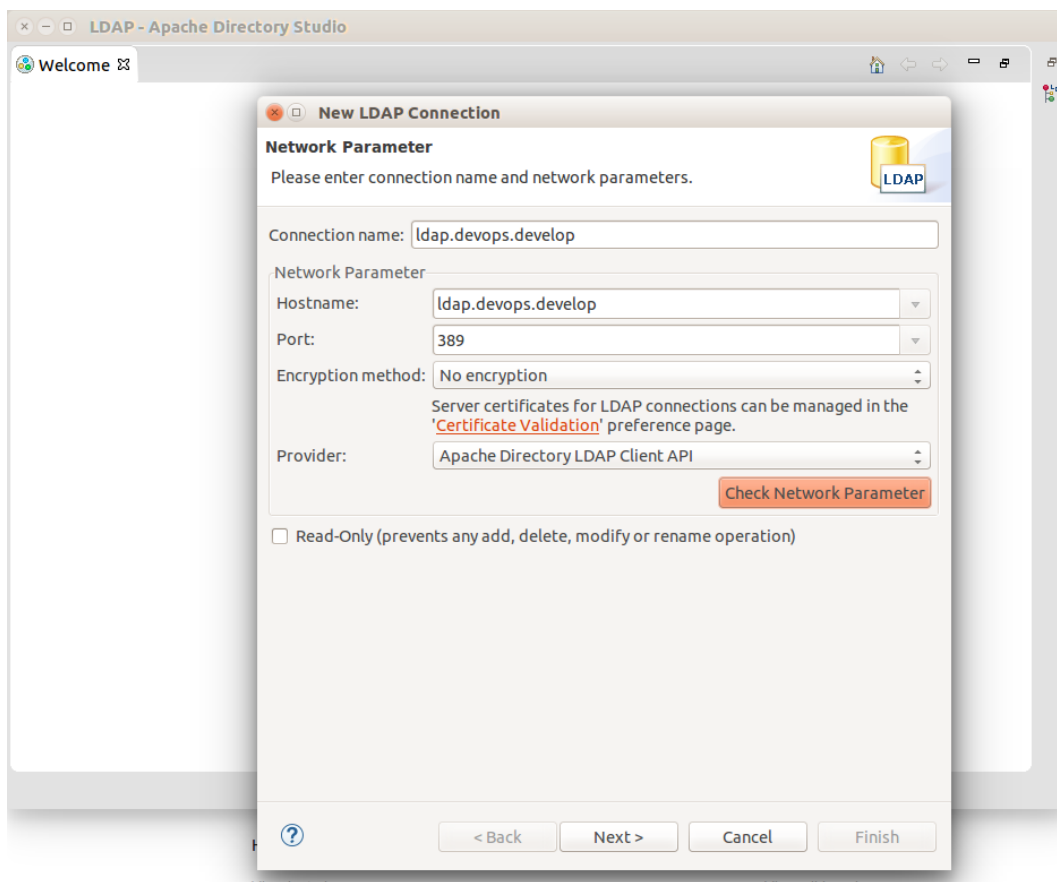


Fig. 15.1: Entering Network Parameters

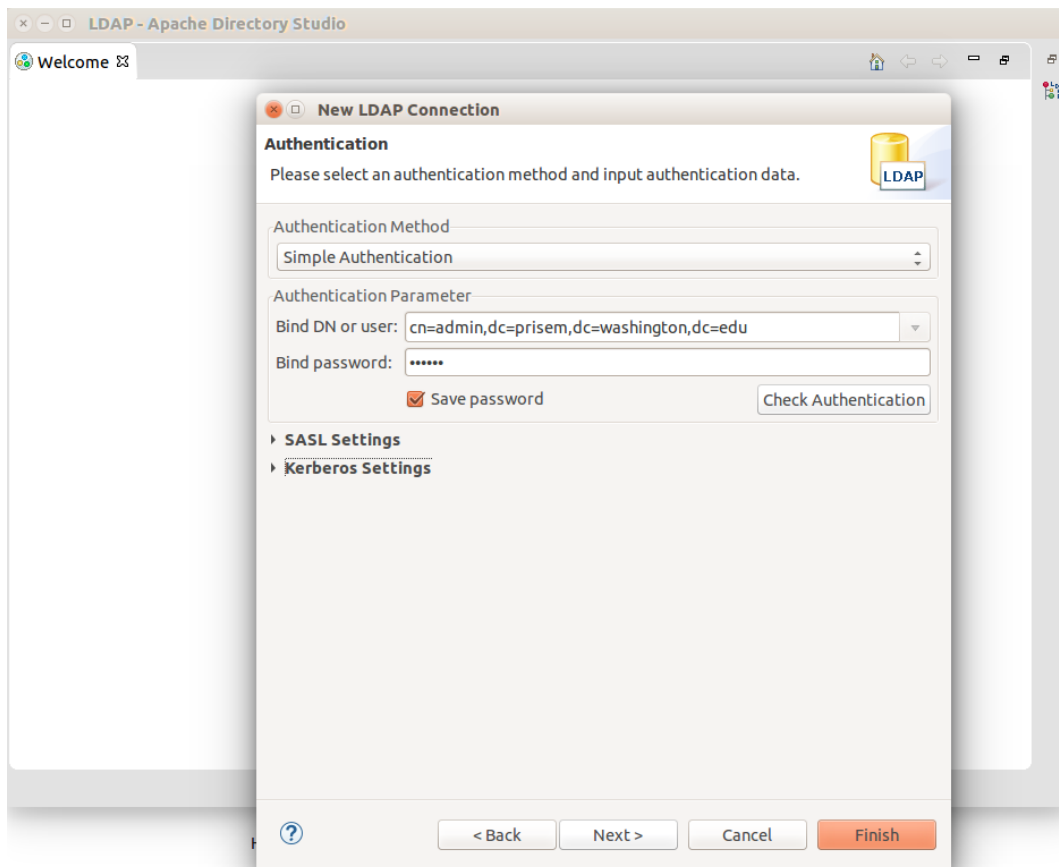


Fig. 15.2: LDAP Connection Authentication

- (e) Click the **Check Authentication** button to make sure you can authenticate.
4. Click **Finish**. The new connection will appear in the **Connections** list and will open. If you minimize the **Welcome** window, the **LDAP Browser** window will occupy the full application window and will remain visible as you operate on the connection.

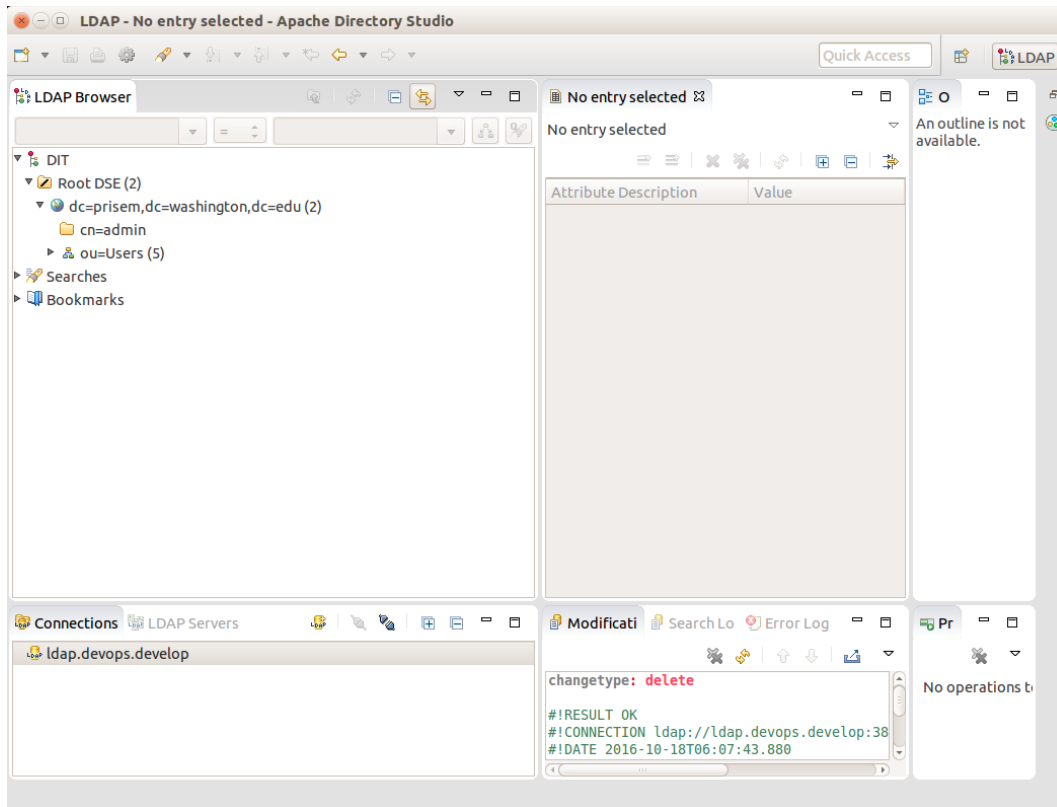


Fig. 15.3: Main LDAP Browser window

## CHAPTER 16

---

### Contact

---

*Section author: Dave Dittrich (@davedittrich) <dittrich @ u.washington.edu>*





# CHAPTER 17

## License

Copyright © 2014, 2016 University of Washington. All rights reserved.

```
Berkeley Three Clause License
=====
```

```
Copyright (c) 2014, 2015 University of Washington. All rights reserved.
```

```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
```

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

```
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```