Sandwich Cloud Documentation

Release 0.0.1

Ryan Belgrave

Jan 28, 2018

Navigation:

Sandwich Cloud is a cloud platform that is built upon VMware VCenter.

Sandwich Cloud aims to bring most of the power of the cloud (er someone else's computer) to your VMware VCenter installation. Unlike OpenStack that is massively complex, Sandwich Cloud tries to be simple and easy with a useable deployment ready in less than an hour.

Due to this simplicity, Sandwich Cloud does not aim to replicate all functionality of a true cloud, instead it focuses on just Virtualization.

To get started all you need is an existing VCenter deployment, some esxi hosts and a routable layer 3 network.

CHAPTER 1

Getting Started

1.1 Requirements

1.1.1 VCenter 6.5

You must have a ready to use VCenter server with the following resources:

- At least one Datastore
- At least one VM Compute Cluster
- At least one Host
- At least one non-management Port Group

The firewall on all hosts (within a cluster) that will be added to Sandwich Cloud needs to allow outgoing connections on the "VM serial port connected to vSPC" service.

All hosts **MUST** be able to connect to your workstation on port 13370. This is used for the Sandwich Cloud Metadata Service. Without this instances will not be able to get networking configuration.

1.1.2 Networking

The mentioned Port Group in the VCenter requirements must be connected to a layer 3 network that is reachable from your workstation. It is recommended that no other devices be on this network aside from a router to prevent possible IP address overlaps.

1.1.3 Workstation

- Internet Access
- **Docker**

- If you are not using Docker for Mac or Docker for Windows the steps in "Setting up the CLI" may need to be modified to fit your environment.

1.2 Setting up the Environment

For this guide we will be using the Sandwich Cloud quick-start repository that will spin up Sandwich Cloud containers and it's dependencies using docker-compose.

So let's start by cloning the repository and changing our working directory to it.

```
git clone https://github.com/sandwichcloud/quick-start.git
cd quick-start
```

1.2.1 Configuration

Most of the configuration items are already set however we still need to configure a few things.

First copy the `.env-sample` file to `.env`

```
cp .env-sample .env
```

Under the `VCENTER` heading enter your connection parameters used to onnect to your VCenter server. It is recommended to use a user that has admin privileges, however any user with the correct permissions will work.

```
VCENTER_HOST=vcenter.exmple.com
VCENTER_PORT=443
VCENTER_USERNAME=administrator@vsphere.local
VCENTER_PASSWORD=password123
```

You also must give the `MENU_URL`. This is the telnet url to the metadata service for Sandwich Cloud. The host in this url should be your workstation's address and must be reachable by your compute hosts. The metadata service's port is set to 13370 and must be given in the url.

```
MENU_URL=telnet://192.168.0.32:13370
```

1.2.2 Launching

Now that everything is configured we can go ahead and launch all the services.

```
docker-compose up
```

You must wait until all services are started before continuing. You will see the following output when everything is done launching:

```
deli-manager_1              | [2018-01-11T01:20:58+0000][deli.manager.cli.commands.run.
↪RunManager][INFO] Creating CRDs
deli-menu_1                 | [2018-01-11T01:20:58+0000][deli.menu.vspc.server.
↪VSPCServer][INFO] Serving on ('0.0.0.0', 13370)
deli-manager_1              | [2018-01-11T01:21:08+0000][deli.manager.cli.commands.run.
↪RunManager][INFO] CRDs have been created
```

1.2.3 Create the database

The quick-start is configured to use Database Auth so you must run the database migrations.

```
docker-compose exec deli-counter deli_counter database upgrade
```

1.2.4 Creating the admin user

The admin user is not created by default so we must create it and generate a password.

You can do that by running the following command:

```
docker-compose exec deli-counter deli_counter database gen-admin
```

Make a note of the password as it will be used later.

1.3 First Steps

1.3.1 Setting up the CLI

To get started you first need to download the CLI. Download the latest release of the CLI from https://github.com/sandwichcloud/deli-cli/releases.

Once the CLI is downloaded you need to configure it to connect to the api server:

```
export DLEI_API_SERVER=http://localhost:8080
```

Now you can login to the API:

```
deli auth login -u admin
```

When prompted enter the password for the generated admin user. Once verified an API token will be generated and saved in `~/.sandwich/credentials`. This API token will automatically be used by the CLI to authenticate against the API.

1.3.2 Creating a Region

In Sandwich Cloud, a Region is linked to a VCenter Datacenter as well as an Image Datastore.

The Image Datastore is just a VCenter Datastore that we designate as our Image storage. This Datastore must be connected to all VCenter Compute Clusters that you want to make available to Sandwich Cloud.

You can create a region by running the following:

```
deli region create --datacenter ${DATACENTER_NAME} --image-datastore ${DATASTORE_NAME}
↪ region1
```

Make a note of the region id as this will be used later.

Now that we have a region we need to enable scheduling to it. If we don't VMs cannot be launched in this region.

```
deli region update --schedulable ${REGION_ID}
```

1.3.3 Creating a Network

Networks are unique to Regions and require a VCenter Port Group. This Port Group must be connected to all VCenter Compute Clusters that you want to make available.

You also must already have a routable address space available for that Port Group. It is recommended that there are no other devices other than a router present on this address space.

To get started please have handy the CIDR and the default gateway for the address space. You also must decide on an address pool within the CIDR. This typically ranges from the first usable address to the last usable address, the gateway will be automatically excluded from this range.

We will be using Google's DNS servers to make things simple, however feel free to substitute your own. At least one DNS server must be given.

Once you have those run the following command:

```
deli network create --port-group ${PORT_GROUP} --region-id ${REGION_ID} \
--cidr ${CIDR} --gateway ${GATEWAY} --dns-server 8.8.8.8 --dns-server 8.8.4.4 \
--pool-start ${POOL_START} --pool-end ${POOL_END} my-network
```

Make a note of the network id as this will be used later.

1.3.4 Creating a Zone

Zones, unique to Regions, are linked to VCenter Compute Clusters. They also, similar to regions, require their own Datastore that we call the VM Datastore. The VM Datastore must be shared across all Hosts within the VM Compute Cluster and will store all active VMs and Volumes within the Zone.

To create a zone run the following:

```
deli zone create --region-id ${REGION_ID} --vm-cluster ${VM_COMPUTE_CLUSTER} \
--vm-datastore ${VM_DATASTORE} region1-a
```

1.3.5 Creating a Project

Now that we have added compute resources we can logically separate these resources into Projects. Projects can be specific to applications, teams, or departments, it is really up to you.

```
deli project create my-project
```

Now that you have your project you must configure your CLI to be scoped to that Project.

```
deli auth scope ${PROJECT_ID}
```

Scoping to a project simply takes your auth token and generates a new one that has permissions to modify resources within the Project. However don't worry, your original token is still available and can still be used to interact with non-project based resources as well as scoping to other Projects.

By default projects are not allowed to create any resources, you can fix this by modifying the quota for the project.

```
deli project quota modify --vcpu=12 --ram=8182 --disk=100
```

If you don't care about quota you can set vcpu, ram, and disk to -1 and the project will be able to use unlimited resources. However, setting project quotas is recommended as it is a good way to limit resource usage in your Datacenter.

1.3.6 Importing an Image

Before we launch an instance we first must have an Image to launch from.

An Image is simply a VM Template with a preinstalled OS configured in a certain way to be compatible with Sandwich Cloud.

Official Images can be downloaded from https://github.com/sandwichcloud/images/releases. If you do not wish to download a pre-build image feel free to build one yourself by using the packer scripts in https://github.com/sandwichcloud/images.

Once the Image is downloaded, un-compress it, upload it into the Image Datastore and create a VM with the only hard drive set to the downloaded VMDK. The VM should have the following hardware:

- 1 CPU
- 512MB Memory
- 1 Hard Disk (set to the downloaded vmdk)
- 1 SCSI Controller (set to VMware Paravirtual)
- 1 Network adapter

Do not add any other hardware to the VM as it may create issues with operation.

Do not boot up the VM as it will introduce unwanted log files into the image. Make sure the image is placed inside the Datacenter and Image Datastore you specified when you created the region.

Once the image is imported to VCenter convert it to a template then you can import it to Sandwich Cloud.

```
deli image import --region-id ${REGION_ID} my-new-image $TEMPLATE_NAME
```

Make a note of the image id as this will be used later.

Note: To read more about images and learn how to create your own see: *Images*

1.3.7 Creating a Flavor

Flavors define instances types or sizing of instances. Flavors control the vcpus, ram, and root disk size of instances.

By default there are no flavors defined so you must create one.

```
deli flavor create --vcpus 2 --ram 2048 --disk 20 my-flavor
```

Make a note of the flavor id as this will be used later.

1.3.8 Creating a SSH Keypair

You are almost ready to launch an instance but we are missing one piece, an SSH key. For this guide we will be generating a new SSH key, however feel free to use the `import` command to import your own.

```
deli keypair generate my-keypair
```

Make a note of the keypair id as this will be used later.

1.3.9 Launching an Instance

Now you can finally launch the instance!

```
deli instance create --region-id ${REGION_ID} --network-id ${NETWORK_ID} \
--flavor-id ${FLAVOR_ID} --image-id ${IMAGE_ID} --keypair-id ${KEYPAIR_ID} my-instance
```

The instance will now be launching in VCenter. You can get the state of the instance by running the following command:

```
deli instance inspect ${INSTANCE_ID}
```

Once the instance state has changed to 'Created' it has now booted. To get the IP address of the instance inspect the instance and find the `network_port_id`, then inspect the network port to grab it's IP address:

```
deli network port inspect ${NETWORK_PORT_ID}
```

Now you can SSH into the instance!

```
ssh cloud-user@${IP_ADDRESS} -i ~/.ssh/id_my-keypair
```

CHAPTER 2

Command Line Client

2.1 Command List

All commands take the following arguments:

`--api-server` The Sandwich Cloud API Server URL | Environment Variable: DELI_API_SERVER

`-d | --debug` Turn on debug logging

`--raw` See the raw API server output

2.1.1 auth

auth login

Login to the Sandwich Cloud API

```
deli auth login
  --method <method>
  --username <username>
  --password <password>
```

`--method <method>` Method to use for auth. If not given uses the API default.

`-u | --username <username>` The username to auth with.

`-p | --paswword <password>` User password to auth with. If not given, will prompt for input

---

Note: Some auth methods may not require a username or password.

---

auth info

Show information about the current auth tokens

```
deli auth info <type>
```

<type> The type of token to view information from (scoped or unscoped)

auth scope

Scope the current auth token to a project

```
deli auth scope <project-id>
```

<project-id> The project id to scope to

2.1.2 flavor

flavor create

Create a flavor

```
deli flavor create
  --vcpus <vcpus>
  --ram <ram>
  --disk <disk>
  <name>
```

--vcpus <vcpus> Number of VCPUs for the flavor

--ram <ram> Amount of ram in megabytes for the flavor

--disk <disk> Size in gigabytes of the root disk for the flavor

<name> The flavor name

flavor delete

Delete a flavor

```
deli flavor delete <flavor-id>
```

<flavor-id> The flavor ID

flavor inspect

Inspect a flavor

```
deli flavor inspect <flavor-id>
```

<flavor-id> The flavor ID

flavor list

List flavors

```
deli flavor list
```

2.1.3 image

image delete

Delete an image

```
deli image delete <image-id>
```

<image-id> The image ID

image import

Import an image

```
deli image import
  --region-id <region-id>
  <name>
  <file-name>
```

--region-id <region-id> The region to import the image into

<name> The image name

<file-name> The image's VMware Template name.

image inspect

Inspect an image

```
deli image inspect <image-id>
```

<image-id> The image ID

image list

List images

```
deli image list
  --visibility <visibility>
```

--visibility <visibility> The visibility to filter by (PUBLIC, PRIVATE>)

image lock

image unlock

image visibility

Change the visibility of an image

```
deli image visibility
  --[no-]public
  <image-id>
```

--[no-]public Enable or disable public visibility of an image

<image-id> The image ID

image member

image member add

Add a project as a member of the image

```
deli image member add
  <image-id>
  <project-id>
```

<image-id> The image ID

<project-id> The project ID

image member remove

Remove a project from an image

```
deli image member remove
  <image-id>
  <project-id>
```

<image-id> The image ID

<project-id> The project ID

image member list

List an image's members

```
deli image member list <image-id>
```

<image-id> Optional. The image ID

2.1.4 instance

instance create

Create an instance

```
deli instance create
  --region-id <region-id>
  --zone-id <zone-id>
  --image-id <image-id>
  --service-account-id <service-account-id>
  --network-id <network-id>
  --flavor-id <flavor-id>
  --disk <size>
  --keypair-id <keypair-id>
  --tag <key>=<value>
  <name>
```

--region-id <region-id> The region to launch the instance in

--zone-id <zone-id> Optional. The zone to launch the instance in

--image-id <image-id> The image to launch the instance from

--service-account-id <service-account-id> Optional. The service account to attach to the instance

--network-id <network-id> The network to attach the instance to

--flavor-id <flavor-id> The flavor of the instance to launch

--disk <size> Optional. The size of the root disk to create, this overrides the flavor

--keypair-id <keypair-id> Optional. A keypair to add to the instance. Can be set multiple times for multiple keypairs

--tag <key>=<value> Optional. A metadata tag to add to the instance. Can be set multiple times for multiple tags

instance delete

Delete an instance

```
deli instance delete <instance-id>
```

<instance-id> The instance ID

instance inspect

Inspect an instance

```
deli instance inspect <instance-id>
```

<instance-id> The instance ID

instance list

List instances

```
deli instance list
  --image-id <image-id>
```

--image-id <image-id> Optional. The image ID to filter instances by

instance image

Create an image from an instance

```
deli instance image
  --name <name>
  <instance-id>
```

--name <name> The image name

<instance-id> The instance ID

instance restart

Restart an instance

```
deli instance restart
  --hard
  --timeout <timeout>
  <instance-id>
```

--hard Optional. Hard stop the instance

--timeout <timeout> Optional. Time in seconds until the instance is hard stopped. Default: 60

<instance-id> The instance ID

instance start

Start an instance

```
deli instance start <instance-id>
```

<instance-id> The instance ID

instance stop

Stop an instance

```
deli instance restart
  --hard
  --timeout <timeout>
  <instance-id>
```

--hard Optional. Hard stop the instance

--timeout <timeout> Optional. Time in seconds until the instance is hard stopped. Default: 60

<instance-id> The instance ID

2.1.5 keypair

keypair delete

Delete a keypair

```
deli keypair delete <keypair-id>
```

<keypair-id> The keypair ID

keypair generate

Generate a keypair

```
deli keypair generate
  --key-dir <key-dir>
  <name>
```

--key-dir <key-dir> Optional. Directory to save the keypair to. Default: ~/.ssh

<name> They keypair name

keypair import

Import a keypair

```
deli keypair import
  <name>
  <public-key-file>
```

<name> The keypair name

<public-key-file> The public key file for the keypair

keypair inspect

Inspect a keypair

```
deli keypair inspect <keypair-id>
```

<keypair-id> The keypair ID

keypair list

List keypairs

```
deli keypair list
```

2.1.6 metadata

metadata meta-data

View instance meta-data

```
deli metadata meta-data
```

metadata network-data

View instance network-data

```
deli metadata network-data
```

metadata user-data

View instance user-data

```
deli metadata user-data
```

2.1.7 network

network create

Create a network

```
deli network create
  --region-id <region-id>
  --port-group <port-group>
  --cidr <cidr>
  --gateway <gateway>
  --dns-server <dns-server>
  --pool-start <pool-start>
  --pool-end <pool-end>
  <name>
```

--region-id <region-id> The region to create the network in

--port-group <port-group> The port group for the network

--cidr <cidr> The network CIDR

--gateway <gateway> The network gateway

--dns-server <dns-server> DNS Server for the network. Can be repeated multiple times.

--pool-start <pool-start> The address for the start of the IP pool

--pool-end <pool-end> The address for the end of the IP pool

<name> The network name

network delete

Delete a network

```
deli network delete <network-id>
```

<network-id> The network ID

network inspect

Inspect a network

```
deli network inspect <network-id>
```

<network-id> The network ID

network list

List networks

```
deli network list
```

network port

network port delete

Delete a network port delete

```
deli network port delete <network-port-id>
```

<network-port-id> The network port ID

network port inspect

Inspect a network port

```
deli network port inspect <network-port-id>
```

<network-port-id> The network port ID

network port list

List network ports

```
deli network port list
```

2.1.8 policy

policy inspect

Inspect a policy

```
deli policy inspect <name>
```

<name> The policy name

policy list

List policies

```
deli policy list
```

2.1.9 project

project create

Create a project

```
deli project create <name>
```

<name> The project name

project delete

Delete a project

```
deli project delete <project-id>
```

<project-id> The project ID

project inspect

Inspect a project

```
deli project inspect <project-id>
```

<project-id> The project ID

project list

List projects

```
deli project list
  --all
```

--all Optional. Include projects you are not a member of

project member

project member add

Add a member to a project

```
deli project member add <username> <driver>
```

<username> The username of the user to add

<driver> The drive of the user to add

project member inspect

Inspect a project member

```
deli project member inspect <member-id>
```

<member-id> The project member ID

project member list

List project members

```
deli project member list
```

project member remove

Remove a member from a project

```
deli project member remove <member-id>
```

<member-id> The project member ID

project member update

Update a project member's roles

```
deli project member update
  --role-id <role-id>
  <member-id>
```

--role-id <role-id> The role to give the member. Can be repeated multiple times.

<member-id> The project member ID

project quota

project quota inspect

Inspect a project's quota

```
deli project quota inspect
```

project quota modify

Modify a project's quota

```
deli project quota modify
  --vcpu <vcpu>
  --ram <ram>
  --disk <size>
```

--vcpu <vcpu> The number of vcpus the project is allowed to use

--ram <ram> The amount of ram (in MB) the project is allowed to use

--disk <size> The amount of disk (in GB) the project is allowed to use

2.1.10 region

region create

Create a region

```
deli region create
  --datacenter <datacenter>
  --image-datastore <datastore>
  --image-folder <image-folder>
  <name>
```

--datacenter <datacenter> The VMware Datacenter for the region

--image-datastore <datastore> The VMware Datastore to keep images in

--image-folder <image-folder> Optional. The VMware VM & Tempaltes folder to keep images in

<name> The region name

region delete

Delete a region

```
deli region delete <region-id>
```

<region-id> The region ID

region inspect

Inspect a region

```
deli region inspect <region-id>
```

<region-id> The region ID

region list

List regions

```
deli region list
```

region update

Update a region

```
deli region update
  --[no-]schedulable
  <region-id>
```

--[no-]schedulable Enable or disable the ability to schedule workloads in the region

<region-id> The region ID

2.1.11 global-role

global-role create

Create a global role

```
deli global-role create
  --policy <policy-name>
  <name>
```

--policy <policy-name> Policy to add to the role. Can be repeated multiple times for multiple policies.

<name> The global role name

global-role delete

Delete a global role

```
deli global-role delete <global-role-id>
```

<global-role-id> The global role ID

global-role inspect

Inspect a global role

```
deli global-role inspect <global-role-id>
```

<global-role-id> The global role ID

global-role list

List global roles

```
deli global-role list
```

global-role update

Update a global role

```
deli global-role update
  --policy <policy-name>
  <global-role-id>
```

--policy <policy-name> Policy to add to the role. Can be repeated multiple times for multiple policies.

<global-role-id> The global role ID

2.1.12 project-role

project-role create

Create a project role

```
deli project-role create
  --policy <policy-name>
  <name>
```

--policy <policy-name> Policy to add to the role. Can be repeated multiple times for multiple policies.

<name> The project role name

project-role delete

Delete a project role

```
deli project-role delete <project-role-id>
```

<project-role-id> The project role ID

project-role inspect

Inspect a project role

```
deli project-role inspect <project-role-id>
```

<project-role-id> The project role ID

project-role list

List project roles

```
deli project-role list
```

project-role update

Update a project role

```
deli project-role update
  --policy <policy-name>
  <project-role-id>
```

--policy <policy-name> Policy to add to the role. Can be repeated multiple times for multiple policies.

<project-role-id> The project role ID

2.1.13 service-account

service-account create

service-account delete

service-account list

service-account update

2.1.14 volume

volume create

Create a volume

```
deli volume create
  --zone-id <zone-id>
  --size <size>
  <name>
```

--zone-id <zone-id> The zone tp create the volume in

--size <size> The size of the volume in gigabytes

<name> The volume name

volume delete

Delete a volume

```
deli volume delete <volume-id>
```

<volume-id> The volume ID

volume inspect

Inspect a volume

```
deli volume inspect <volume-id>
```

<volume-id> The volume ID

volume list

List volumes

```
deli volume list
```

volume attach

Attach a volume to an instance

```
deli volume attach <volume-id> <instance-id>
```

<volume-id> The volume ID

<instance-id> The instance ID

volume clone

Clone a volume

```
deli volume clone <volume-id>
```

<volume-id> The volume ID

volume detach

Detach a volume from an instance

```
deli volume detach <volume-id>
```

<volume-id> The volume ID

volume grow

Increase the size of a volume

```
deli volume grow <volume-id> <size>
```

<volume-id> The volume ID

<size> The size of the volume in gigabytes

2.1.15 zone

zone create

Create a zone

```
deli zone create
  --region-id <region-id>
  --vm-cluster <vm-cluster>
  --vm-datastore <vm-datastore>
  --vm-folder <vm-folder>
  --core-provision-percent <core-provision-percent>
  --ram-provision-percent <ram-provision-percent>
```

--region-id <region-id> The region this zone belongs to

--vm-cluster <vm-cluster> The VMware cluster for this zone

--vm-datastore <vm-datastore> The VMware datastore for this zone

--vm-folder <vm-folder> Optional. The VMware VM & Templates folder to keep vms in

--core-provision-percent <core-provision-percent> Optional. The percentage of cores to provision from the VMware cluster. Default: 1600

--ram-provision-percent <ram-provision-percent> Optional. The percentage of ram to provision from the VMware cluster. Default: 150

zone delete

Delete a zone

```
deli zone delete <zone-id>
```

<zone-id> The zone ID

zone inspect

Inspect a zone

```
deli zone inspect <zone-id>
```

<zone-id> The zone ID

zone list

List zones

```
deli zone list
  --region-id <region-id>
```

--region-id <region-id> Optional. The region ID to filter by

zone update

Update a zone

```
deli zone update
  --[no-]schedulable
  <zone-id>
```

--[no-]schedulable Enable or disable the ability to schedule workloads in the zone

<zone-id> The zone ID

CHAPTER 3

Administration

3.1 Policies and Roles

3.1.1 Policies

Policies

`policies:get` Ability to get a policy

`policies:list` Ability to list policies

Roles

`roles:global:create` Ability to create a global role

`roles:project:create` Ability to create a project role

`roles:global:get` Ability to get a global role

`roles:project:get` Ability to get a project role

`roles:global:list` Ability to list global roles

`roles:project:list` Ability to list project roles

`roles:global:update` Ability to update a global role

`roles:project:update` Ability to update a project role

`roles:global:delete` Ability to delete a global role

`roles:project:create` Ability to delete a project role

Flavors

flavors:create Ability to create a flavor

flavors:get Ability to get a flavor

flavors:list Ability to list flavors

flavors:delete Ability to delete a flavor

Regions

regions:create Ability to create a region

regions:get Ability to get a region

regions:list Ability to list regions

regions:delete Ability to delete a region

regions:action:schedule Ability to change the schedule mode of the region

Zones

zones:create Ability to create a zone

zones:get Ability to get a zone

zones:list Ability to list zones

zones:delete Ability to delete a zone

zones:action:schedule Ability to change the schedule mode of the zone

Projects

projects:create Ability to create a project

projects:get Ability to get a project

projects:get:all Ability to get all projects

projects:list Ability to list projects

projects:list:all Ability to list all projects

projects:delete Ability to delete a project

projects:scope Ability to scope to a project

projects:scope:all Ability to scope to all projects

projects:members:add Ability to add a member to a project

projects:members:get Ability to get a member in a project

projects:members:list Ability to list members in a project

projects:members:modify Ability to modify a project member's roles

projects:members:remove Ability to remove a member from a project

projects:quota:get Ability to get a project's quota

projects:quota:modify Ability to modify a project's quota

Volumes

volumes:create Ability to create a volume

volumes:get Ability to get a volume

volumes:list Ability to list volumes

volumes:delete Ability to delete a volume

volumes:action:attach Ability to attach a volume to an instance

volumes:action:detach Ability to detach a volume from an instance

volumes:action:grow Ability to grow a volume

volumes:action:clone Ability to clone a volume

Images

images:create Ability to create an image

images:get Ability to get an image

images:list Ability to list images

images:delete Ability to delete an image

images:action:visibility Ability to change an image's visibility

images:action:visibility:public Ability to change an image's visibility to public

images:action:lock Ability to lock an image

images:action:unlock Ability to unlock an image

images:members:add Ability to add a member to an image

images:members:list Ability to list an image's members

images:members:delete Ability to delete a member from an image

Instances

instances:create Ability to create an instance

instances:get Ability to get an instance

instances:list Ability to list instances

instances:delete Ability to delete an instance

instances:action:stop Ability to stop an instance

instances:action:start Ability to start an instance

instances:action:restart Ability to restart an instance

instances:action:image Ability to create an image from an instance

Networks

networks:create Ability to create a network

networks:get Ability to get a network

networks:list Ability to list networks

networks:delete Ability to delete a network

Service Accounts

service_accounts:create Ability to create a service account

service_accounts:get Ability to get a service account

service_accounts:list Ability to list service accounts

service_accounts:update Ability to update a service account

service_accounts:delete Ability to delete a service account

Keypairs

keypairs:create Ability to create a keypair

keypairs:get Ability to get a keypair

keypairs:list Ability to list keypairs

keypairs:delete Ability to delete a keypair

Network Ports

network_ports:get Ability to get a network port

network_ports:list Ability to list network ports

network_ports:delete Ability to delete a network port

Database Users

`database:users:create`

`database:users:get`

`database:users:list`

`database:users:delete`

`database:users:password`

`database:users:roles:update`

3.1.2 Roles

Global Roles

Admin Role

The administrative role for Sandwich Cloud. This role has access to all API endpoints.

Project Roles

Project roles can only have policies that are for project based resources.

Default Member

This is the default role for all project members. This role has access to all scoped API endpoints.

Default Service Account

This is the default service account role for all project service accounts. This role has access to read-only scoped API endpoints.

3.2 Images

3.2.1 Official

3.2.2 Custom

3.3 VMware

3.3.1 Account Permissions

3.3.2 Firewall

All hosts in all Clusters connected to Sandwich Cloud Zones must allow the following firewall rule `VM serial port connected to vSPC`.

All hosts must also be able to communicate with the Sandwich Cloud Metadata Service.

3.3.3 VMotion

Regular

Feel free to VMotion VMs to different hosts to balance out your clusters.

> **Warning:** VMs must be kept within the same Compute Cluster. If they are not Sandwich Cloud may loose track of the VM and any action performed may result in unexpected behavior.

Storage

DO NOT Storage VMotion VMs. If you are migrating to a new datastore please see *Migrating Datastores*.

3.3.4 Maintenance

Migrating Datastores

Regions

Migrating a Region's Image Datastore requires a new Region to be created. This requires building out the new Region, importing images and redeploying VMs and Volumes.

1. Create a new Region connected to the new Image Datastore
2. Create new Zones in the new Region and enable scheduling.
3. Enable scheduling into the new Region.
4. Disable scheduling into the old Region.
5. Import images into the new Region.
6. Deploy replacement VMs into the new Region.
7. Decommission Zones in the old Region.

Zones

Migrating a Zone's VM Datastore requires a new Zone to be created. This requires redeploying VMs and Volumes into the new Region.

1. Create a new zone connected to the same Compute Cluster but with the new VM Datastore.
2. Enable scheduling into the new Zone.
3. Disable scheduling into the old Zone.
4. Deploy replacement VMs into the new Zone.
5. Delete the old Zone.

3.4 Security

3.4.1 Metadata Service

3.5 Authentication

3.5.1 Fernet Keys & Tokens

Sandwich Cloud uses Fernet tokens for authorization. These tokens require a key to be encrypted. The key must be a 32-byte base64 encoded string. Multiple Fernet Keys can be used to allow key rotation.

These tokens contain information about the user or service account and what roles they have access to. User tokens expire one day after they are generated while service account tokens expire 30 minutes after they are requested from the Metadata service.

3.5.2 Drivers

There are many ways to authenticate to Sandwich Cloud. Authentication is pluggable and is handled by drivers.

Database

The Database driver authenticates users against a database that is managed by Sandwich Cloud.

Github

The Github driver authenticates users against public GitHub or a private Github Enterprise installation.

Gitlab

Not implemented

LDAP

Not implemented

OpenID

Not implemented

3.6 Configuration

Configuration of all Sandwich Cloud services are managed by environment variables.

3.6.1 Counter - API Server

Kubernetes

KUBECONFIG Path to a Kubernetes client configuration file to communicate with a Kubernetes Cluster. If not given will default to using the in-cluster configuration.

KUBEMASTER The address of the Kubernetes API server. This will override any value that is set in the `KUBECONFIG`

Authentication

AUTH_FERNET_KEYS A url safe 32-bit base64 encoded string used to encrypt user tokens. Multiple keys can be listed to allow rotation (comma separated). The first key in the list is the primary key. To rotate keys simply generate a new key and put it in the front of the list, then after a while remove the old key from the list.

AUTH_DRIVERS A python module path to a class that implements an auth driver. Multiple auth drivers can be given as a comma separated string. The first driver in the list is considered the default auth driver that clients will default to. If no drivers are given it will default to the Database Driver.

Driver Paths:

- Github: deli.counter.auth.drivers.github.driver:GithubAuthDriver
- Database: deli.counter.auth.drivers.database.driver:DatabaseAuthDriver

Database Auth

These configuration items are only needed when using database authentication.

DATABASE_DRIVER The database driver to use to connect. Some drivers may require additional python libraries to work.

Valid Supported Values:

- postgresql
- mysql
- sqlite

DATABASE_DB The name of the database. If using sqlite this is usually the path to the sqlite file.

DATABASE_HOST The address of the database host

DATABASE_PORT The port used to connect to the database host

DATABASE_USERNAME The username used to connect to the database

DATABASE_PASSWORD The password used to connect to the database

Github Auth

These configuration items are only needed when using github authentication.

GITHUB_URL The Github API url. Defaults to the public Github API url.

GITHUB_CLIENT_ID The client ID used to authenticate to the Github API

GITHUB_CLIENT_SECRET The client secret used to authenticate to the Github API

GITHUB_ORG The github organization users must be part of. This organization is also used to check user teams.

GITHUB_TEAM_ROLES A static mapping of sandwich cloud global roles to github teams. These static mappings will override GITHUB_TEAM_ROLES_REFIX if a role is found.

Examples:

- **admin:sandwich-admin**
  - A Github team called `sandwich-admin` will be mapped to the global role called `admin`.
- **role1:sandwich-role1**
  - A Github team called `sandwich-role1` will be mapped to the global role called `role1`.

GITHUB_TEAM_ROLES_PREFIX Prefix to use when searching for Github teams. If no static mapping for a role is given this prefix will be used.

Example:

- **sandwich-**
  - For a Github team called `sandwich-role1` a global role with the name of `role1` will be given to the user.

3.6.2 Manager - Resource Controller

Kubernetes

KUBECONFIG Path to a Kubernetes client configuration file to communicate with a Kubernetes Cluster. If not given will default to using the in-cluster configuration.

KUBEMASTER The address of the Kubernetes API server. This will override any value that is set in the `KUBECONFIG`

VMware VCenter

VCENTER_HOST The address used to connect to the VMware Vcenter server

VCENTER_PORT The port used to connect to the VMware Vcenter server

VCENTER_USERNAME The username to authenticate with against the VMware Vcenter server

VCENTER_PASSWORD The password to authenticate with against the VMware Vcenter server

Menu

MENU_URL The telnet url of the Metadata Service.

For SSL connections use the following format: telnets://<host>:<port>#key[=value][&key[=value] ...]

- **thumbprint=value**
  - Specifies a certificate thumbprint against which the peer certificate thumbprint is compared. When you specify a thumbprint, certificate verification is automatically enabled. See the description of the verify parameter below.
- **peerName=value**
  - Specifies the peer name that will be used to validate the peer certificate. When you specify a peer name, certificate verification is automatically enabled. See the description of the verify parameter below.
- **verify**

– Forces certificate verification. The virtual machine will verify that the peer certificate subject matches the specified peerName and that it was signed by a certificate authority known to the ESXi host. Verification is automatically enabled if you specify a thumbprint or peerName.

3.6.3 Menu - Metadata Service

Kubernetes

KUBECONFIG Path to a Kubernetes client configuration file to communicate with a Kubernetes Cluster. If not given will default to using the in-cluster configuration.

KUBEMASTER The address of the Kubernetes API server. This will override any value that is set in the `KUBECONFIG`

Authentication

FERNET_KEY A url safe 32-bit base64 encoded string used to encrypt service account tokens. This must match a fernet key that the API Server uses.