
DevOps-At-Scale Documentation

Release 1.0

NetApp

Jun 03, 2019

Contents

1	Contents	3
1.1	Prerequisites	3
1.2	Installation	3
1.2.1	Installing Using Helm Package Manager	3
1.2.2	Additional Configuration	6
1.3	General Usage	6
1.3.1	Pipeline Creation	6
1.3.2	Integrate GitLab with Jenkins for automatic build triggers	7
1.3.3	Workspace Creation	10
1.3.4	Merge Workspace Creation	10
1.4	Uninstalling	14
1.5	Support	14
1.6	License	14
1.7	References	15
1.7.1	Installation and setup of Kubernetes cluster using Ansible	15
1.7.2	Installation and setup of Trident on Kubernetes using Ansible	16
1.8	Release Notes	17
1.8.1	Release 1.1: Known Issues	17

DevOps-at-Scale is a Kubernetes based open source solution which provides:

- Centralized management of the entire Software Development Tools Ecosystem
- Centralized management of developer workspaces
- Fully containerized tools environment (all deployed as Kubernetes services run within pods)
- Simplified creation of CI/CD pipelines for source code repositories
- Quick and storage efficient developer workspace creation using ontap technology
- Easy workspace access for developers via Theia IDE or NFS mounts
- One click installation via Helm Package Manager

1.1 Prerequisites

- 1 running instance of Data ONTAP cluster
- Kubernetes cluster RBAC cluster
- NetApp Trident Installation with Kubernetes
- Helm Package manager

Note: Please see <https://kubernetes.io/docs/setup/> for kubernetes installation instructions. Please check Trident documentation for supported Kubernetes version.

Note: Please ensure your Kubernetes cluster, ONTAP cluster, and Trident can communicate with each other and reside in secure network(s)

Note: Please visit [References](#) on how to use Ansible to automate Kubernetes cluster installation and setup

Note: Please visit [References](#) on how to use Ansible to automate Trident installation in a Kubernetes cluster

1.2 Installation

1.2.1 Installing Using Helm Package Manager

1. Download source code from github

```
git clone https://github.com/NetApp/devops-at-scale
```

2. Go to the “devops-at-scale” directory

```
cd ./devops-at-scale
```

3. Enter storage details and installation options by modifying values.yaml

```
cat values.yaml
global:
  # "LoadBalancer" or "NodePort"
  ServiceType: NodePort
  scm:
    # "gitlab" or "bitbucket"
    type: "gitlab"
  registry:
    # "artifactory" or "docker-registry"
    type: "artifactory"
  persistence:
    ontap:
      # If set to "true", ontap volumes for various services (E.g. gitlab/
      ↪artifactory/couchdb) will be automatically created
      automaticVolumeCreation: true
      # ontap data lif IP address
      dataIP: ""
      # ontap SVM name
      svm: ""
      # ontap aggregate
      aggregate: ""
```

4. Install helm chart using following command :

```
helm install --name devops-at-scale .
```

Note: If helm is not already installed , visit <https://helm.sh/> for installation instructions

5. Wait for pods to reach the “Running” state:

```
>kubectl get pods | grep devops-at-scale
```

NAME	READY	STATUS	RESTARTS	AGE
devops-at-scale-couchdb-58f48c5b8d-vw9mb	1/1	Running	0	3m
devops-at-scale-docker-registry-7969844c9f-phshp	1/1	Running	0	3m
devops-at-scale-gitlab-6c6dc79b77-j4dww	1/1	Running	0	3m
devops-at-scale-jenkins-74d87d6fd5-th29g	1/1	Running	0	3m

(continues on next page)

(continued from previous page)

devops-at-scale-webservice-5bbcd8f88c-rjrp4	1/1	Running	0	↵
↵ 3m				

Note: It may take up to 10 minutes for all the pods to come up.

6. After the pods are ready, retrieve the webservice URL:

```
>kubectl get svc
```

NAME	TYPE	CLUSTER-IP	AGE
↵EXTERNAL-IP	PORT(S)		
devops-at-scale-couchdb	NodePort	10.108.249.65	5m
↵<none>	5984:14339/TCP		
devops-at-scale-docker-registry	NodePort	10.97.110.240	5m
↵<none>	5000:24646/TCP		
devops-at-scale-gitlab	NodePort	10.102.216.157	5m
↵<none>	80:30593/TCP, 22:8639/TCP, 443:18600/TCP		
devops-at-scale-jenkins	NodePort	10.99.97.28	5m
↵<none>	8080:12899/TCP		
devops-at-scale-jenkins-agent	ClusterIP	10.100.249.190	5m
↵<none>	50000/TCP		
devops-at-scale-webservice	NodePort	10.101.38.243	
↵<none>	5000:12054/TCP		

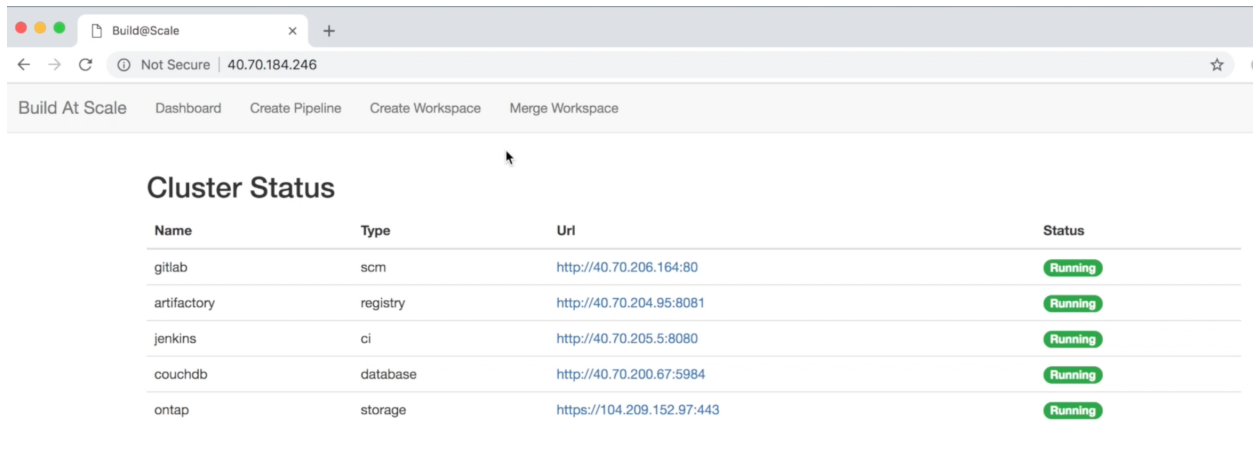
```
export NODE_IP=$(kubectl get nodes -o jsonpath="{.items[0].status.
↵addresses[0].address}")
export SERVICE_PORT=$(kubectl get -o jsonpath="{.spec.ports[0].nodePort}"
↵services {{.Release.Name}}-webservice)
export SERVICE_URL=$NODE_IP:$SERVICE_PORT
```

Note: Take note of the port of web service. The web service will be available at \$SERVICE_URL:<devops-at-scale-webservice-port>

7. Using a Web Browser, open the “devops-at-scale-webservice” URL ([http://<\\$SERVICE_URL>:<devops-at-scale-webservice-port>](http://<$SERVICE_URL>:<devops-at-scale-webservice-port>)) to visit the DevOps-At-Scale Frontend Management Console

Note: GitLab service can be accessed using credentials ‘root:root_devopsatscale’ initially

Note: All other services can be accessed using credentials ‘admin:admin’ initially



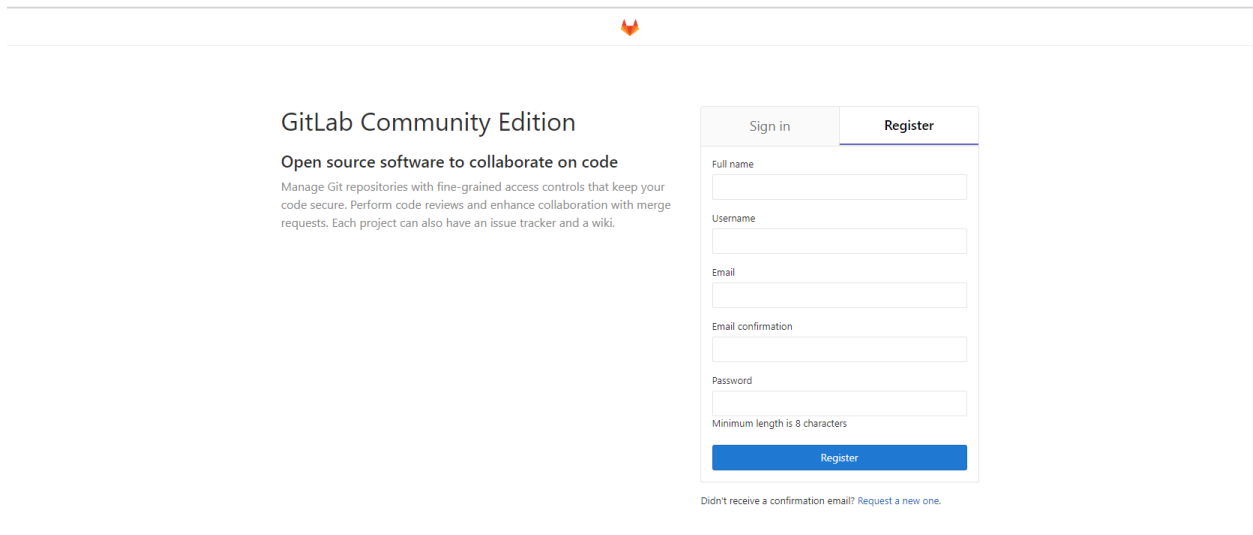
Name	Type	Url	Status
gitlab	scm	http://40.70.206.164:80	Running
artifactory	registry	http://40.70.204.95:8081	Running
jenkins	ci	http://40.70.205.5:8080	Running
couchdb	database	http://40.70.200.67:5984	Running
ontap	storage	https://104.209.152.97:443	Running

1.2.2 Additional Configuration

Create Initial GitLab User (Optional)

An initial account has to be created on Gitlab before starting to use it. To create an account on Gitlab, visit the following URL and sign up.

```
http://<<$SERVICE_URL>>:<<Gitlab_port>>
```



GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in Register

Full name

Username

Email

Email confirmation

Password

Minimum length is 8 characters

Register

Didn't receive a confirmation email? [Request a new one.](#)

1.3 General Usage

1.3.1 Pipeline Creation

DevOps-at-Scale pipelines can be created via pipeline creation page:

```
http://<<$SERVICE_URL>>:<<devops-at-scale-webservice-port>>/frontend/pipeline/
↩ create
```

Build At Scale	Dashboard	Create Pipeline	Create Workspace	Merge Workspace
SCM URL	<input type="text" value="http://build-at-scale-gitlab/hello/MySqlApp.git"/>			
SCM Branch	<input type="text" value="master"/>			
Export Policy	<input type="text" value="default"/>			
<input type="button" value="Create Pipeline"/>				

Parameter	Value	Description
SCM-URL		URL of the source code repository
SCM-Branch		SCM branch off which the pipeline should run
Export-policy	default	Export-policy that should be used for the pipeline volume

Once the pipeline creation is successful, a Jenkins project with pre-populated build parameters is setup

The screenshot shows the Jenkins web interface. The top navigation bar includes links like 'Getting Started', 'Most Visited', 'Help', 'Ngage', 'GoodReads', 'NetApp Help', and 'Build@Scale'. The main header displays the Jenkins logo and a search bar. The left sidebar contains navigation options: 'Back to Dashboard', 'Status', 'Changes', 'Build with Parameters', 'Delete Pipeline', 'Configure', 'Full Stage View', 'Rename', 'Pipeline Syntax', 'Polling Log', 'Build History', and RSS feeds. The main content area is titled 'Pipeline pipeline-trident-v1.0' and lists build parameters with their values:

- BUILDVOL**: trident_default_pipeline_trident_v1_0_pvc_03326
- BUILDVOLCLAIM**: pipeline-trident-v1.0-pvc
- SOURCE_CODE_URL**: http://[redacted].git
- SOURCE_CODE_BRANCH**: v1.0
- CONTAINER_REGISTRY**: devops-at-scale-artifactory
- SCM_VOLUME**: trident_default_devops_at_scale_gitlab_pvc_e1e5c
- SCM_VOLUME_CLAIM**: devops-at-scale-gitlab-pvc
- WEB_SERVICE_URL**: http://10.193.113.152:30567
- JENKINS_SLAVE_IMAGE**: jenkins/nlp-slave:3.10-1
- RUN_CLEAN_BUILD**: ☐

A 'Build' button is located at the bottom of the parameter list.

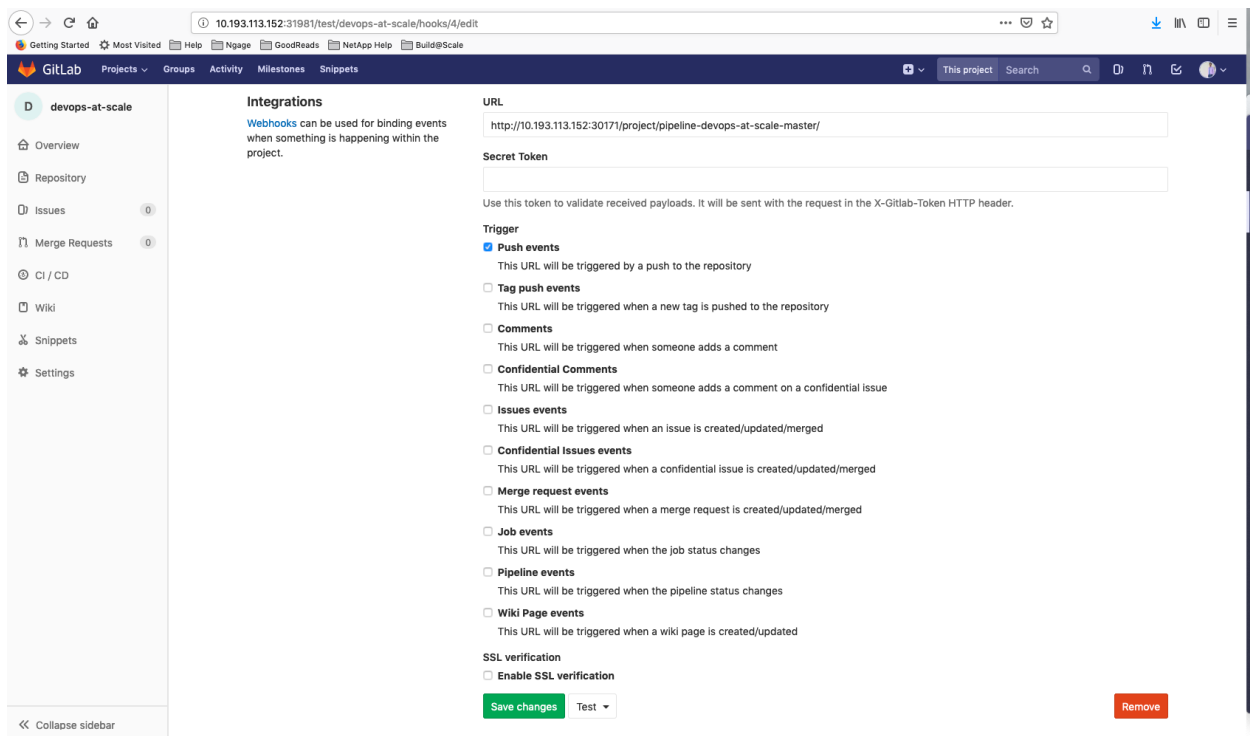
1.3.2 Integrate GitLab with Jenkins for automatic build triggers

1. From the webservice dashboard, copy the Jenkins URL for the pipeline created
2. Open GitLab from the webservice dashboard ([http://<<\\$SERVICE_URL>>:<<devops-at-scale-webservice-port>>](http://<<$SERVICE_URL>>:<<devops-at-scale-webservice-port>>))
3. Login using root/root_devopsatscale
4. In the GitLab project, goto Settings -> Integrations and paste the Jenkins project URL from step (1) and create the webhook

Pipelines

Pipeline	SCM	CI	Last CI Build	Action
pipeline-trident-master	http://devops-at-scale-gitlab/akshayp/trident	http://10.193.113.152:30171/job/pipeline-trident-master/	SUCCESS	Delete
pipeline-trident-v1.0	http://devops-at-scale-gitlab-ff4cc9f8-kssrt/akshayp/trident	http://10.193.113.152:30171/job/pipeline-trident-v1.0/	FAILURE	Delete
pipeline-devops-at-scale-master	http://devops-at-scale-gitlab/test/devops-at-scale	http://10.193.113.152:30171/job/pipeline-devops-at-scale-master/	SUCCESS	Delete

Note: When pasting the Jenkins URL, replace `/job/<jenkins-project-name>` with `/project/<jenkins-project-name>`

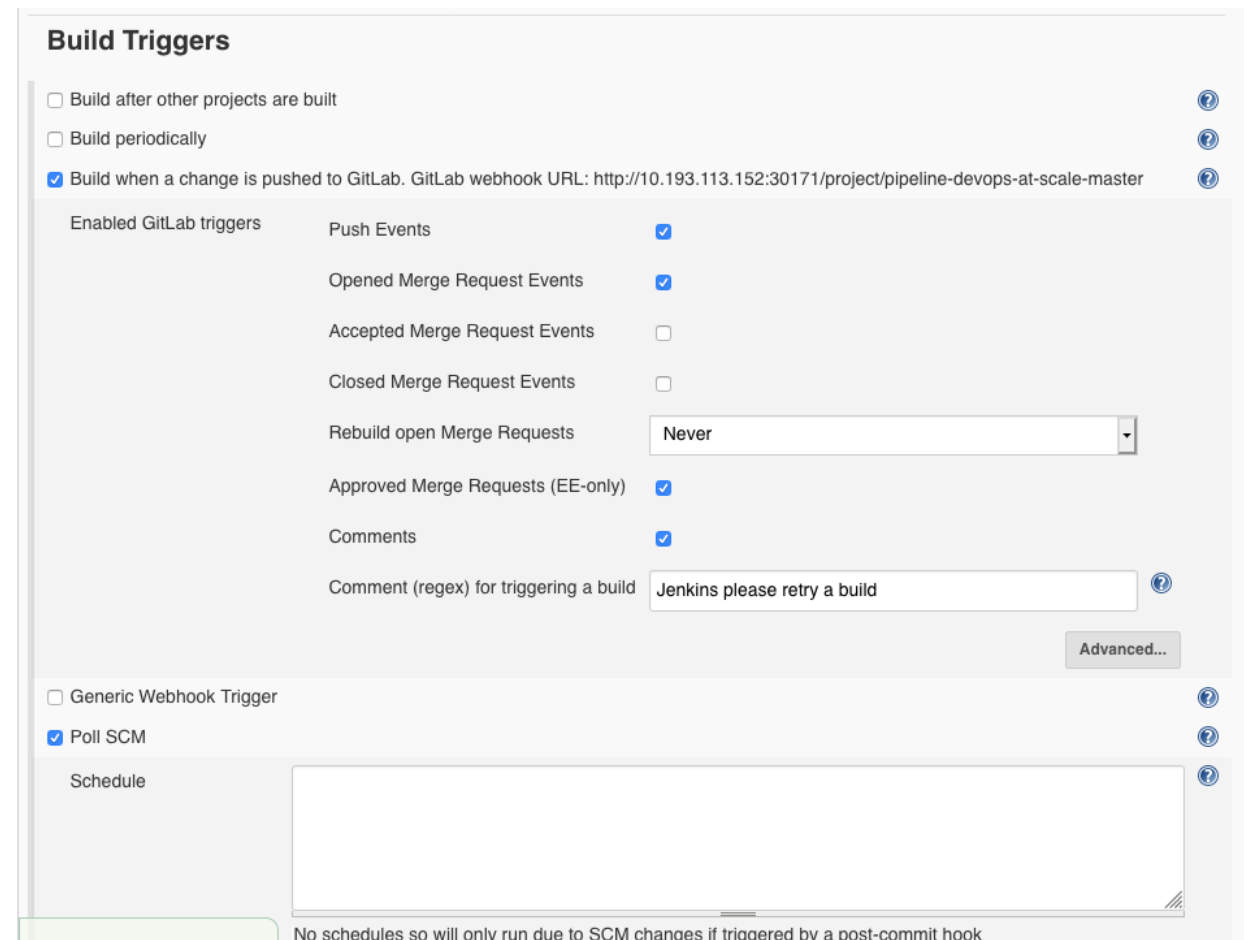
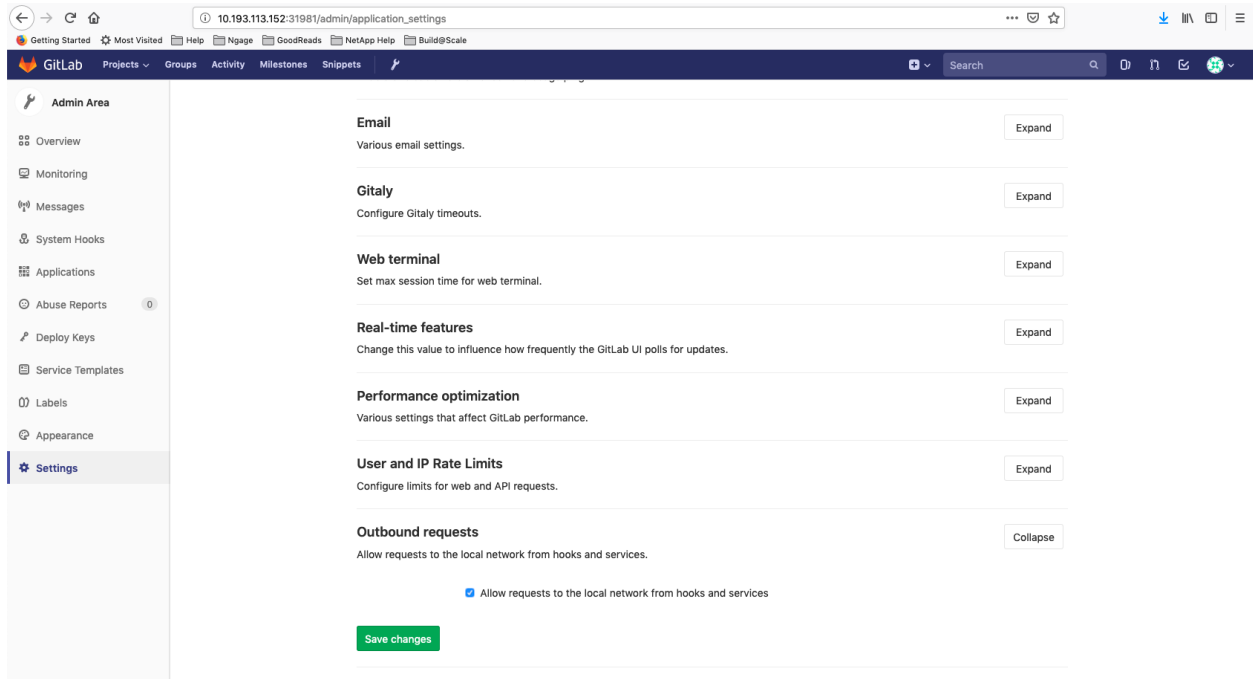


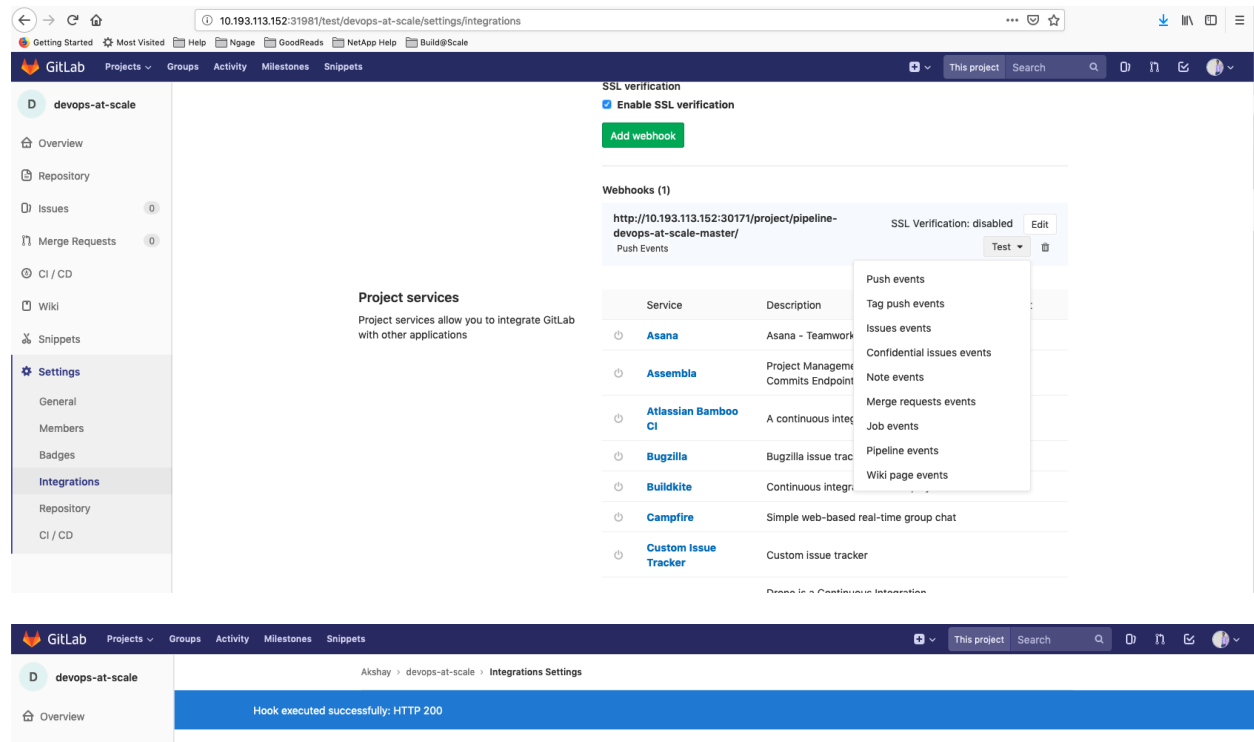
5. In global Gitlab settings, allow outbound requests from local network

6. Enable the build trigger from webhook in Jenkins. Navigate to the pipeline's Jenkins URL from the webservice dashboard and goto Configure -> Build Triggers

7. Webhook setup is complete. Test the webhook setup manually from GitLab (Project -> Settings -> Integrations -> Webhook -> Test -> Push Events)

This will validate whether the GitLab and Jenkins integration has been successful





8. All further pushes to the GitLab project will automatically trigger a build in Jenkins project corresponding to the pipeline

1.3.3 Workspace Creation

DevOps-at-Scale workspaces can be created via workspace creation page:

```
http://<<$SERVICE_URL>>:<<devops-at-scale-webservice-port>>/frontend/workspace/
↩create
```

Parameter	Value	Description
Pipeline		Select the pipeline
Username		Developer username
Workspace prefix		Enter a prefix which can be used to identify the workspace
Build		Select the build from which the workspace should be created

Once a workspace is created, you will be provided instructions on how to access your workspace via Theia Browser IDE or locally via NFS:

1.3.4 Merge Workspace Creation

DevOps-at-Scale merge workspaces can be created via workspace creation page:

```
http://<<$SERVICE_URL>>:<<devops-at-scale-webservice-port>>/frontend/workspace/
↩merge
```

The screenshot shows the Jenkins web interface for a pipeline named 'pipeline-devops-at-scale-master'. The left sidebar contains navigation links: Back to Dashboard, Status, Changes, Build with Parameters, Delete Pipeline, Configure, Full Stage View, Rename, Pipeline Syntax, and Git Polling Log. The main area displays the 'Stage View' with a table of stage execution times.

Pipeline pipeline-devops-at-scale-master

Recent Changes

Stage View

Average stage times: (Average full run time: ~38s)

	Create SCM Clone using Trident	Setup	Checkout	Build	Create CI Clone using Trident	Determine overall build status
May 31, 2019 6:44 PM	339ms	1s	5s	614ms	7s	59ms
May 31, 2019 4:41 PM	427ms					
May 31, 2019 4:29 PM	388ms	1s	4s	719ms	3s	55ms
May 31, 2019 4:29 PM	456ms	1s	5s	527ms	4s	60ms
May 30, 2019 21:54	86ms	1s	5s	596ms	13s	64ms

Build History

find

- #4 May 31, 2019 6:44 PM
Started by GitLab push by [redacted]
- #3 May 31, 2019 4:41 PM
Started by GitLab push by [redacted]
Commit: 97c634
- #2 May 31, 2019 4:29 PM
Started by GitLab push by [redacted]
Commit: 80c421b
- #1 May 31, 2019 4:54 AM
Commit: 820a535

BSS for all BSS for failures

The screenshot shows the 'Create Workspace' form in the 'Build At Scale' application. The form includes fields for Pipeline, Username, Workspace Name Prefix, and Build, each with a dropdown menu. A 'Creating Workspace...' button is at the bottom.

Build At Scale Dashboard Create Pipeline **Create Workspace** Merge Workspace

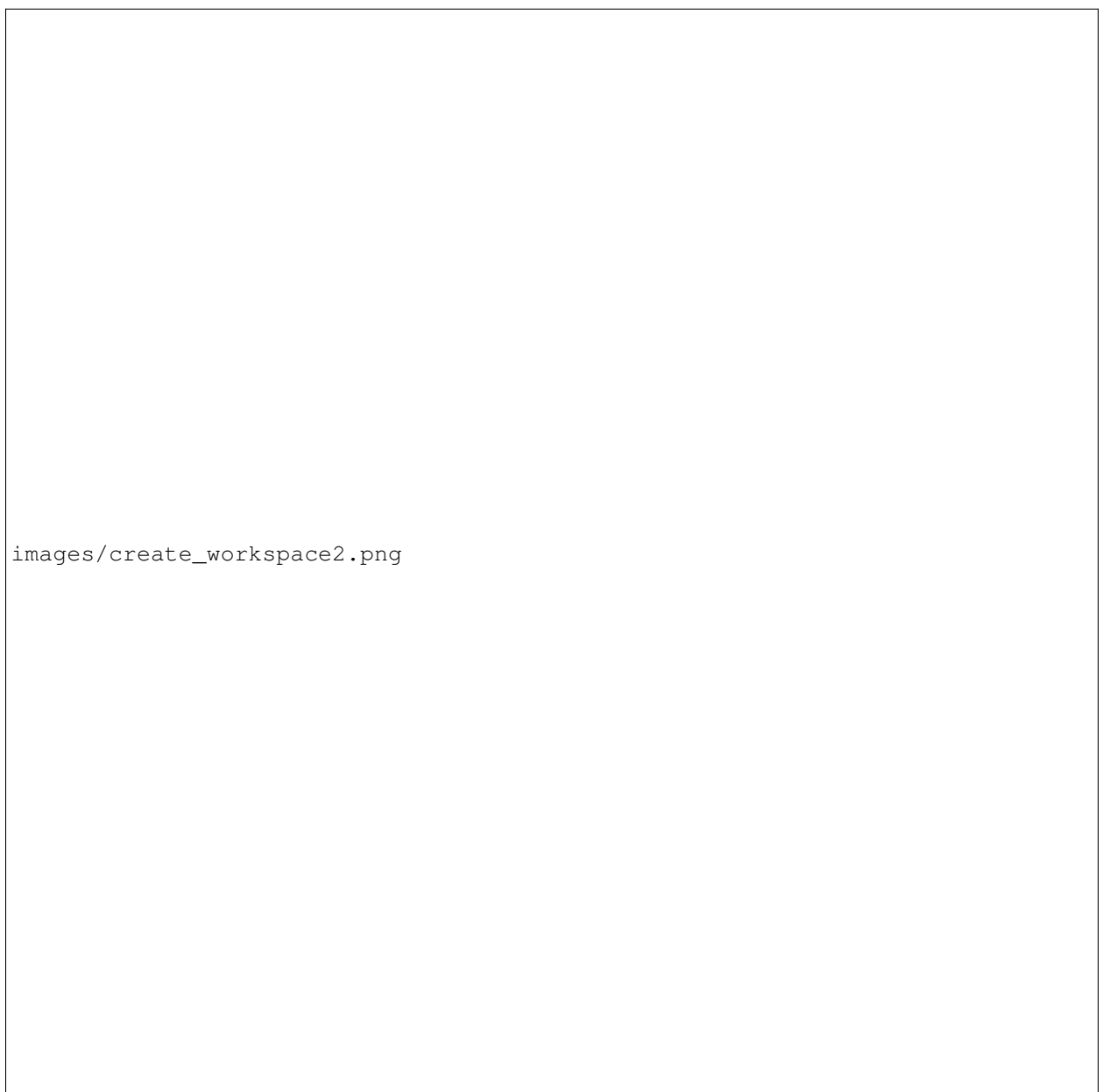
Pipeline insight-demo-master

Username default

Workspace Name Prefix default

Build cf7ea94_2_201810010116

Creating Workspace...



images/create_workspace2.png

[Build At Scale](#) [Dashboard](#) [Create Pipeline](#) [Create Workspace](#) [Merge Workspace](#)

Workspace Details

Workspace created successfully!

Please access your workspace using one of the following methods:

Option A: Theia Browser IDE

To access your workspace via Theia IDE , please visit the following url:

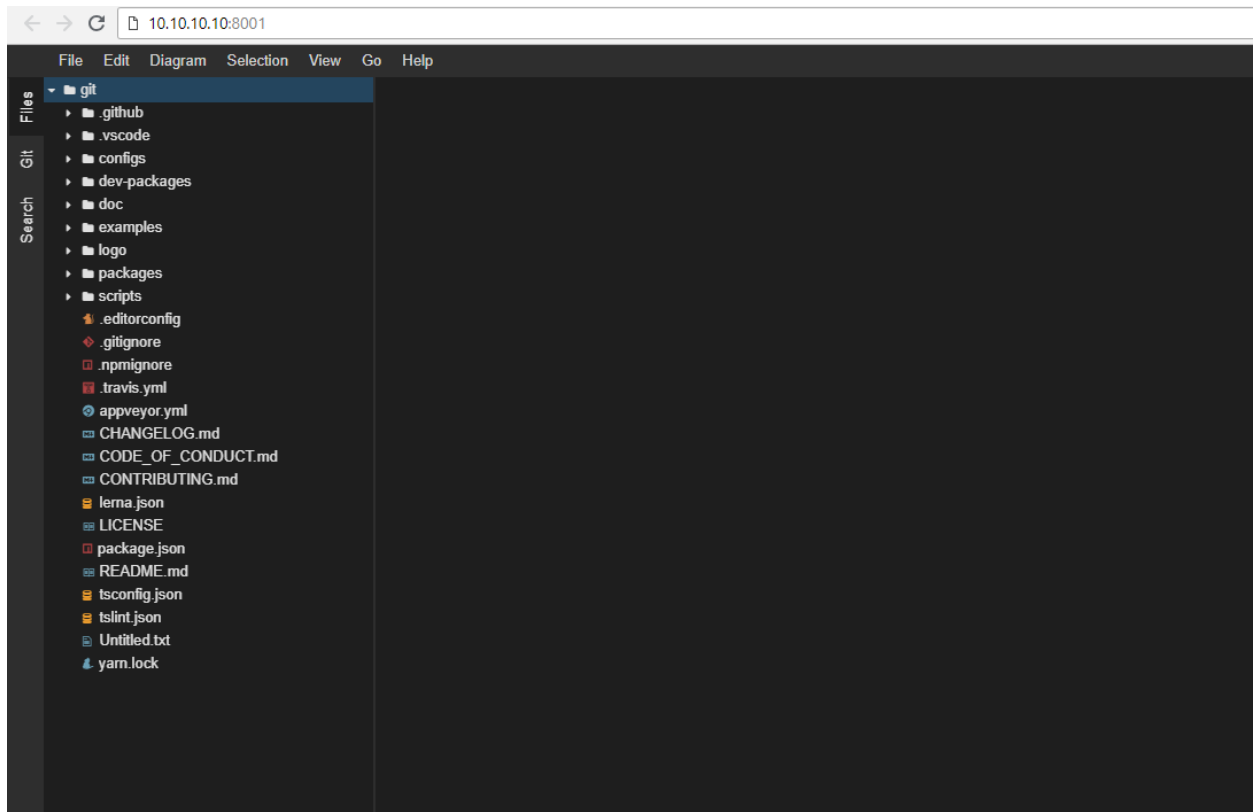
10.10.10.10:31936

Option B: Local NFS Mount

Your workspace can be mounted to your personal computer using NFS.

To mount the workspace locally , please run the following:

```
mount -t nfs 10.0.0.1:/project_master_workspacedmhc /mnt/ws-mount-point
```



Users can merge their workspace with the latest build when they feel their workspace is out of date.

This allows users to pull in the latest code and artifacts into their workspace , thus potentially providing incremental build time savings.

To merge workspaces, navigate to the Merge Workspace tab and fill in the following values :-

Build At Scale	Dashboard	Create Pipeline	Create Workspace	Merge Workspace
Username	<input type="text" value="default"/>			
Workspace Name Prefix	<input type="text" value="default"/>			
Source workspace name	<input type="text" value="default_orignalws"/>			
Build	<input type="text" value="9210aao"/>			
<input type="button" value="Create Merge Workspace"/>				

Parameter	Value	Description
Username		Developer username
Workspace Name Prefix		Enter a prefix which can be used to identify the workspace
Source Workspace name		Enter name of the source workspace to merge from
Build		Select the build which the workspace should be created off

1.4 Uninstalling

Build-at-Scale can be uninstalled using a single command

```
helm del --purge devops-at-scale
```

Note: Once all the services' PVCs are deleted, Trident deletes the associated PVs and ONTAP volumes

1.5 Support

Support for Build-at-Scale is handled via [Slack](#).

Please post your comments in the #devops-at-scale channel

Note: Support is done on a best effort basis

1.6 License

```
BSD 3-Clause License
```

```
Copyright (c) 2018-2019, NetApp, Inc.
```

(continues on next page)

(continued from previous page)

All rights reserved.

Redistribution and use in `source` and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of `source` code must retain the above copyright notice, `↪this` list of conditions and the following disclaimer.
- * 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.
- * 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.

1.7 References

1.7.1 Installation and setup of Kubernetes cluster using Ansible

Pre-requisites

1. If you do not have an Ansible setup. Please setup by following the instructions [here](#)
2. One or more VMs reachable from where Ansible playbooks are being run

Note: Ansible playbooks referred in the below steps are located in `devops-at-scale/devops-at-scale/ansible-playbooks/k8s_setup`

Usage

1. Download roles

```
ansible-galaxy install --roles-path roles -c geerlingguy.docker
ansible-galaxy install --roles-path roles -c geerlingguy.kubernetes
```

2. Create inventory file

```
$ cat inventory
[all]
scspa0633050001 kubernetes_role="master"
scspa0633051001 kubernetes_role="node"
```

If more than one node, tag them appropriately.

3. Install docker and kubernetes

```
ansible-playbook -i inventory -K --become-method=su --become k8s_setup_
↪cluster.yml
```

This will install kubectl, kubelet, and create a cluster with worker nodes.

1.7.2 Installation and setup of Trident on Kubernetes using Ansible

Pre-requisites

1. If you do not have an Ansible setup. Please setup by following the instructions from [Ansible Setup](#)
2. Kubernetes cluster. The inventory file identifies master and worker nodes.
3. ONTAP cluster

Note: Ansible playbooks referred in the below steps are located in devops-at-scale/devops-at-scale/ansible-playbooks/trident_setup

Qualify your Kubernetes cluster

```
ansible-playbook -i inventory kubectl_check.yml -K --become --become-
↪method=su --extra-vars=@vsim_vars.yml
(requires root access on K8S master node to run kubectl)
```

Preparation

1. The trident_prereqs.yml playbook will install pip, setuptools, and the openshift python package. This is required to run k8s Ansible module.

This playbook will then create a “trident” namespace.

```
ansible-playbook -i inventory trident_prereqs.yml -K --become --become-
↪method=su
```

Download installer and final checks

2. The trident.yml playbook will install the trident installer and set up a backend storage file to support trident etcd database:

```
ansible-playbook -i inventory trident.yml -K --become --become-method=su -
↪extra-vars=@vsim_vars.yml
```

(requires root access on K8S master node to run yum - and maybe k8s)

Trident installation

3. The next step will be to run the trident installer. In the kubernetes master node:

4. Check Trident is running

```
ansible-playbook -i inventory trident_check_pods.yml -K --become --become-  
↪method=su
```

As of today, you should see: 2/2 Running (1 pod is running 2 containers out of 2)

Trident configuration

5. The backend created in the preparation step is only used to support the Trident etcd persistent storage. New backend(s) need to be created to support production.

Add backend

Being lazy here, we can reuse the same backend

```
trident/trident-installer/tridentctl -n trident create backend -f trident/  
↪trident-installer/setup/backend.json
```

6. Add storage class in Kubernetes. Follow instructions [Trident documentation](#)

7. Test Trident installation by creating first volume and mounting it into an nginx pod. Follow instructions [Trident example](#)

1.8 Release Notes

1.8.1 Release 1.1: Known Issues

- For merge workspace, the new pod is mounting two volumes, one volume with the source workspace and the other volume with a copy of the selected build. The changes have to be merged manually by the developer
- The UID and GID for the workspace and service volumes are defaulted to 0, 0. We will provide customizable UID, GID values in 1.2
- Manual webhook setup for GitLab and Jenkins integration is required for every pipeline
- The solution is only tested with ONTAP using NFS volumes
- The CI pipeline build clones are required to be purged manually
- The number of active clones (build and workspace) is limited by ONTAP. Please check the ONTAP release and make sure the purge policies are in place
- In case of failure during pipeline or workspace creation, the Kubernetes PVCs may have to be purged manually
- For GitLab, the URL for git cloning is incorrect. Please use http://<SERVICE_URL>:<devops-at-scale-gitlab-port>/ during git clone.