5. DriveTrain in MCP

DriveTrain in MCP

Chapter Details	
Chapter Goal	Deploy and use a Drivetrain cluster
Chapter Sections	*6.1. Prepare virtual lab* *5.2. Deploy Drivetrain* *5.3. Use Drivetrain*

5.1. Prepare virtual lab

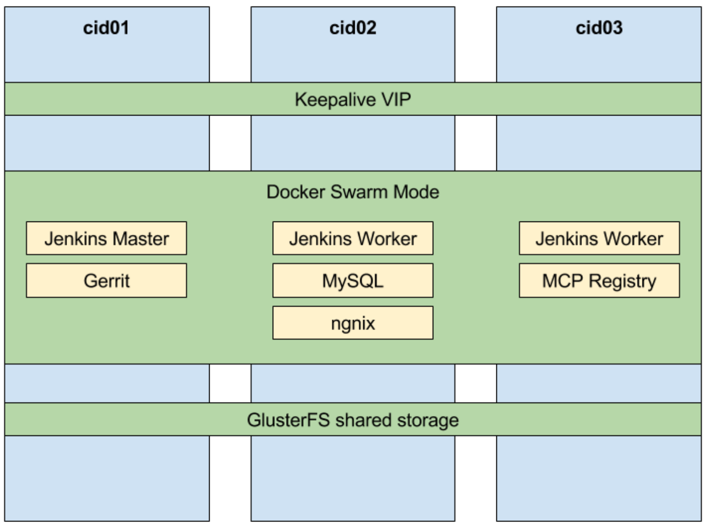
5.1.1. Environment Overview

MCP Drivetrain consists of:

- Jenkins
- Gerrit
- Aptly

Those are integrated together with reclass model.

MCP deployment guide recommends that DriveTrain deployment is done on 3 virtual nodes, where each service (Jenkins, Aptly and Gerrit) runs inside Docker container and those containers are managed by Docker swarm mode. Each of those components uses GlusterFS as a distributed storage for storing service-specific data reliably in distributed fashion. Diagram below shows the recommended deployment model below:



5.1.2. Configure networking

Step 1 Download the model and scripts:

```
stack@lab:~$ git clone --recurse-submodules git@bitbucket.org:mirantis-training/mcp100-
```

Step 2 Install prerequisites:

```
stack@lab:~$ sudo apt-get update

stack@lab:~$ sudo apt-get install -y git mkisofs curl virtinst cpu-checker qemu-kvm
```

Step 3 Create virtual networks:

```
stack@lab:~$ mcp100-drivetrain/provision-networks.sh
```

5.1.3. Prepare cfg01

Step 1 Create a model:

```
stack@lab:~$ cd mcp100-drivetrain/

stack@lab:~/mcp100-drivetrain$ sudo su

root@lab:/home/stack/mcp100-drivetrain# mkdir /root/model
root@lab:/home/stack/mcp100-drivetrain# cp -rT /home/stack/mcp100-drivetrain/ /root/moc
```

Step 2 Download images:

```
root@lab:/home/stack/mcp100-drivetrain# wget http://repos/cfg01-day01-2018.8.0.qcow2 -C
/pool/images/cfg01.drivetrain-ha.local.img
```

Step 3 Prepare directory for configuration ISO. Set MCP version and clone pipeline repositories:

```
root@lab:/home/stack/mcp100-drivetrain# export version="2018.11.0"

root@lab:/home/stack/mcp100-drivetrain# git clone https://github.com/Mirantis/mk-pipeli

Cloning into '/root/mk-pipelines'...
remote: Counting objects: 4064, done.
remote: Compressing objects: 100% (86/86), done.
remote: Total 4064 (delta 66), reused 86 (delta 36), pack-reused 3942
Receiving objects: 100% (4064/4064), 1.02 MiB | 0 bytes/s, done.
Resolving deltas: 100% (2612/2612), done.
Checking connectivity... done.

root@lab:/home/stack/mcp100-drivetrain# git clone https://github.com/Mirantis/pipeline-

Cloning into '/root/pipeline-library'...
remote: Counting objects: 6401, done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 6401 (delta 26), reused 63 (delta 16), pack-reused 6323
Receiving objects: 100% (6401/6401), 927.30 KiB | 0 bytes/s, done.
Resolving deltas: 100% (2924/2924), done.
Checking connectivity... done.
```

Step 4 Checkout specific version of pipelines:

```
root@lab:/home/stack/mcp100-drivetrain# pushd /root/mk-pipelines && git checkout tags/$
~/mk-pipelines ~ /home/stack/mcp100-drivetrain

Note: checking out 'tags/2018.3.1'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at 5dac8d8... cloud-update pipeline rename Merges param
/home/stack

root@lab:/home/stack/mcp100-drivetrain# pushd /root/pipeline-library && git checkout ta
~/pipeline-library /home/stack

Note: checking out 'tags/2018.3.1'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at 658b110... Merge "Change default version of salt-models-testing docker i
/home/stack
```

Step 5 Download script *create_config_drive.sh*:

```
root@lab:/home/stack/mcp100-drivetrain# wget -O /root/create-config-drive \
https://raw.githubusercontent.com/Mirantis/mcp-common-scripts/${version}/config-drive/c

root@lab:/home/stack/mcp100-drivetrain# chmod +x /root/create-config-drive
```

The script will be used to create configuration ISO

Step 6 Download cloud-init script for **cfg01**:

```
root@lab:/home/stack/mcp100-drivetrain# wget -O /root/user_data.sh \
https://raw.githubusercontent.com/Mirantis/mcp-common-scripts/${version}/config-drive/m
```

Step 7 Edit file `/root/user_data.sh`, modify environment parameters as shown below:

```
write_files:
  - owner: root:root
    path: /etc/cloud/master_environment
    permissions: '0644'
    content: |
      [ -f /etc/cloud/master_environment_override ] && . /etc/cloud/master_environment_
      export SALT_MASTER_DEPLOY_IP=192.168.10.100
      export SALT_MASTER_MINION_ID=cfg01.drivetrain-ha.local
      export DEPLOY_NETWORK_GW=192.168.10.1
      export DEPLOY_NETWORK_NETMASK=255.255.255.0
      export DNS_SERVERS=172.19.0.6
```

```
        export DEPLOY_NETWORK_MTU=1500
        export MCP_VERSION=2018.11.0
```

Step 8 Create configuration ISO using downloaded script from previous steps:

```
root@lab:/home/stack/mcp100-drivetrain# /root/create-config-drive -u /root/user_data.sh
--model /root/model --mk-pipelines /root/mk-pipelines \
--pipeline-library /root/pipeline-library /pool/images/cfg01.drivetrain-ha.local.iso

adding user data from /root/user_data.sh
adding reclass model directory /root/model
adding mk_pipelines directory /root/mk-pipelines
adding pipeline_library directory /root/pipeline-library
generating configuration image at /var/lib/libvirt/images/cfg01/cfg01-config.local.iso
```

5.1.4. Create cfg01 VM

Step 1 Create VM:

```
root@lab:/home/stack/mcp100-drivetrain# virt-install --name cfg01.drivetrain-ha.local \
--disk path=/pool/images/cfg01.drivetrain-ha.local.img,bus=virtio,format=qcow2,cache=nc
--disk path=/pool/images/cfg01.drivetrain-ha.local.iso,device=cdrom \
--network network:br_pxe,model=virtio \
--network network:br_mgmt,model=virtio \
--network network:br_ctl,model=virtio \
--network network:br_ext,model=virtio \
--ram 8192 --vcpus=4 --accelerate \
--boot hd --vnc --noreboot --autostart

Starting install...
Creating domain...
Failed to connect to Mir: Failed to connect to server socket: No such file or directory
Unable to init server: Could not connect: Connection refused
Cannot open display:
Run 'virt-viewer --help' to see a full list of available command line options
Domain creation completed. You can restart your domain by running:
  virsh --connect qemu:///system start cfg01.drivetrain-ha.local
```

Step 2 Update networking:

```
root@lab:/home/stack/mcp100-drivetrain# virsh net-update br_pxe add ip-dhcp-host "\
<host mac='$(virsh domiflist cfg01.drivetrain-ha.local | grep br_pxe | awk '{print $5}'
" --live --config

Updated network br_pxe persistent config and live state
```

Step 3 Start cfg01 VM:

```
root@lab:/home/stack/mcp100-drivetrain# virsh start cfg01.drivetrain-ha.local

Domain cfg01.drivetrain-ha.local started
```

Step 4 Enter *cfg01* and wait until cloud-init is done. It should take around 7 min to complete:

```
root@lab:/home/stack/mcp100-drivetrain# virsh console cfg01.drivetrain-ha.local

Connected to domain cfg01.drivetrain-ha.local
Escape character is ^]
```

```
[  OK  ] Started LSB: Machine Check Exceptions (MCE) collector & decoder.
[  OK  ] Started Login Service.
         Starting Daily apt upgrade and clean activities...
         Starting Terminate Plymouth Boot Screen...
         Starting Hold until boot process finishes up...
[  OK  ] Started LSB: Start NTP daemon.
```

Step 5 Update SSH keys:

```
root@cfg01:~# salt-call state.sls openssh
```

Exit to host lab using *CTRL+]*

5.1.5. Create other VMs

Step 1 Spawn cluster VMs:

```
root@lab:/home/stack/mcp100-drivetrain# ./provision-cluster.sh

root@lab:/home/stack/mcp100-drivetrain#  virsh list

 Id    Name                           State
----------------------------------------------------
 14    cfg01.drivetrain-ha.local      running
 15    cid01.drivetrain-ha.local      running
 16    cid02.drivetrain-ha.local      running
 17    cid03.drivetrain-ha.local      running
```

Switch back to *stack* user:

```
root@lab:/home/stack/mcp100-drivetrain# exit

stack@lab:~/mcp100-drivetrain$
```

5.2. Deploy Drivetrain

The deployment of Drivetrain will require several salt states to be applied to cluster nodes.

5.2.1. Prepare Salt

Step 1 Login to *cfg01* using SSH:

```
stack@lab:~/mcp100-drivetrain$ ssh root@192.168.10.100

root@cfg01:~#
```

Step 2 Bootstrap cluster nodes:

```
root@cfg01:~# salt '*' saltutil.sync_all

root@cfg01:~# salt '*' state.sls linux.system.repo

root@cfg01:~# salt '*' state.sls salt.minion,linux,openssh,ntp
```

Step 3 Install GlusterFS:

```
root@cfg01:~# salt -C 'I@glusterfs:server' state.sls glusterfs.server.service

root@cfg01:~# salt -C 'I@glusterfs:server:role:primary' state.sls glusterfs.server.setu
```

Verify GlusterFS server:

```
root@cfg01:~# salt -C 'I@glusterfs:server' cmd.run 'gluster peer status; gluster volume
```

Ensure GlusterFS clusters is ready:

```
root@cfg01:~# salt -C 'I@glusterfs:client' state.sls glusterfs.client
```

Step 4 Install HAProxy:

```
root@cfg01:~# salt -C 'I@haproxy:proxy and I@docker:host' state.sls haproxy,keepalived
```

Step 5 Install Docker:

```
root@cfg01:~# salt -C 'I@docker:host' state.sls docker.host
```

Step 6 Enable swarm master and update salt mines:

```
root@cfg01:~# salt -C 'I@docker:swarm:role:master' state.sls docker.swarm

root@cfg01:~# salt -C 'I@docker:swarm' state.sls salt
root@cfg01:~# salt -C 'I@docker:swarm' mine.flush
root@cfg01:~# salt -C 'I@docker:swarm' mine.update
root@cfg01:~# salt -C 'I@docker:swarm' saltutil.sync_all
```

Step 7 Enable swarm workers:

```
root@cfg01:~# salt -C 'I@docker:swarm' state.sls docker.swarm
```

Update mines:

```
root@cfg01:~# salt -C 'I@docker:swarm' mine.update
```

Join swarm nodes:

```
root@cfg01:~# salt -C 'I@docker:swarm' state.sls docker.swarm
```

Verify swarm:

```
root@cfg01:~# salt -C 'I@docker:swarm:role:master' cmd.run 'docker node ls'
```

Step 8 Install Aptly:

```
root@cfg01:~# salt -C 'I@aptly:publisher' state.sls aptly.publisher

root@cfg01:~# salt -C 'I@docker:swarm:role:master' state.sls docker.client

Check that all replicas have been spawned:
```

```
root@cfg01:~# salt -C 'I@docker:swarm:role:master' cmd.run 'docker service ls'
```

Step 9 Update pillars:

```
root@cfg01:~# salt '*' saltutil.sync_all
```

Step 10 Install openldap:

```
root@cfg01:~# salt -C 'I@openldap:client' state.sls openldap
```

Step 11 Install gerrit:

```
root@cfg01:~# salt -C 'I@gerrit:client' state.sls gerrit
```

Step 12 Install jenkins:

```
root@cfg01:~# salt -C 'I@jenkins:client' state.sls jenkins
```

5.3. Use Drivetrain

5.3.1. Create ad-hoc formula with a static file

Step 1 Create dir:

```
root@cfg01:~# mkdir /srv/salt/env/prd/hello
```

Step 2 Create state `/srv/salt/env/prd/hello/init.sls` file:

```
/etc/hello.conf:
  file.managed:
    - source: salt://hello/files/hello.conf
    - user: root
    - group: root
    - mode: 644
```

Step 3 Create dir:

```
root@cfg01:~# mkdir /srv/salt/env/prd/hello/files
```

Step 4 Create a static file `/srv/salt/env/prd/hello/files/hello.conf`:

```
This is hello file version 1
```

Step 5 Run state:

```
root@cfg01:~# salt "cfg01*" state.sls hello
```

Step 6 Verify the result:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 1
```

5.3.2. Store the formula on git and install it from git using your cluster model

Step 1 Create git repo directory:

```
root@cfg01:~# mkdir /root/hello_repo
root@cfg01:~# cd /root/hello_repo
```

Step 2 Create empty git repo:

```
root@cfg01:~# git init
root@cfg01:~# git config --global user.email "student@mirantis.com"
```

Step 3 Copy formula:

```
root@cfg01:~# cp -r /srv/salt/env/prd/hello/ .
```

Step 4 Increment version in the config file `hello/files/hello.conf`:

```
This is hello file version 2
```

Step 5 Add and commit formula files to the git repo:

```
root@cfg01:~# git add -A
root@cfg01:~# git commit -m 'Initial commit'
```

Step 6 Modify your cluster model to install the formula from git repo. Edit `/srv/salt/reclass/classes/cluster/drivetrain-ha/infra/hello.yml`:

```
parameters:
  salt:
    master:
      environment:
        prd:
          formula:
            hello:
              source: git
              name: salt-formula-hello
              address: /root/hello_repo
              branch: master
```

Edit `/srv/salt/reclass/classes/cluster/drivetrain-ha/infra/config/init.yml`:

```
classes:
- cluster.drivetrain-ha.infra.hello
```

Step 7 Sync pillars:

```
root@cfg01:~# salt \* saltutil.sync_all
```

Step 8 Check if salt state installs your hello formula:

```
root@cfg01:~# salt "cfg01*" state.show_sls salt --output=yaml | less
```

Step 9 Install your hello fomula:

```
root@cfg01:~# salt "cfg01*" state.sls salt
```

Step 10 Check if the old ad-hoc formula is replaced with the new one from git repo:

```
root@cfg01:~# ls -l /srv/salt/env/prd/hello

lrwxrwxrwx 1 root root 50 Sep  2 13:52 /srv/salt/env/prd/hello -> /usr/share/salt-formu

root@cfg01:~# cat /srv/salt/env/prd/hello/files/hello.conf

This is hello file version 2
```

Step 11 Run your hello formula:

```
root@cfg01:~# salt "cfg01*" state.sls hello
```

Step 12 Verify the result:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 2
```

5.3.3. Modify the formula, so hello.conf is parametrized with pillar

Step 1 Add hello version parameter (key-value pair). Edit `/srv/salt/reclass/classes/cluster/drivetrain-ha/infra/hello.yml`:

```
parameters:
  hello:
    version: 3
  salt:
    master:
      environment:
        prd:
          formula:
            hello:
              source: git
              name: salt-formula-hello
              address: /root/hello_repo
              branch: master
```

Step 2 Modify the formula:

```
root@cfg01:~# cd /root/hello_repo
```

Edit `hello/init.sls`:

```
/etc/hello.conf:
  file.managed:
    - source: salt://hello/files/hello.conf
    - user: root
    - group: root
    - mode: 644
    - template: jinja
```

Edit `hello/files/hello.conf`

```
This is hello file version {{ salt['pillar.get']('hello:version') }}
```

Step 3 Add and commit to git repo:

```
root@cfg01:~# git add -A

root@cfg01:~# git commit -m 'Hello ver from pillar'
```

Step 4 Sync:

```
root@cfg01:~# salt \* saltutil.sync_all
```

Step 5 Install the new formula:

```
root@cfg01:~# salt "cfg01*" state.sls salt
```

Step 6 Run the formula:

```
root@cfg01:~# salt "cfg01*" state.sls hello
```

Step 7 Verify the result:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 3
```

5.3.4. Store cluster model in a git repository and source it from there

Step 1 Create git repository:

```
root@cfg01:~# mkdir /root/model_repo
root@cfg01:~# cd /root/model_repo
root@cfg01:~# git init
root@cfg01:~# git config receive.denyCurrentBranch ignore
```

Step 2 Modify the model to use your git repo:

```
root@cfg01:~# cd /srv/salt/reclass
```

Edit `/srv/salt/reclass/classes/cluster/drivetrain-ha/infra/config/nodes.yml`:

```
parameters:
  reclass:
    storage:
      data_source:
        # don't use Git storage for model, instead use whatever is provided
        # with the lab
        # engine: local
        engine: git
        address: /root/model_repo
        branch: master
        force_reset: true
```

Step 3 Add and commit to the git repo in /srv/salt/reclass:

```
root@cfg01:~# git add -A

root@cfg01:~# git commit -m 'Switch storage to Git'
```

Reclass model note:

/srv/salt/reclass repo is created during bootstraping MCP

Step 5 Push the model to your new model repo:

```
root@cfg01:~# git remote set-url origin /root/model_repo/

root@cfg01:~# git push -u origin master
```

Step 6 Regenerate the model in /srv/salt/reclass using your new model repo:

```
root@cfg01:~# salt-call state.sls reclass
```

Step 7 Modify the metadata used by your hello formula:

```
root@cfg01:~# cd /root/model_repo

root@cfg01:~# git checkout
```

Edit `classes/cluster/drivetrain-ha/infra/hello.yml`:

```
parameters:
  hello:
    version: 4
  salt:
    master:
      environment:
        prd:
          formula:
            hello:
              source: git
              name: salt-formula-hello
              address: /root/hello_repo
              branch: master
```

Step 7 Add and commit changes:

```
root@cfg01:~# git config --global user.email "student@mirantis.com"

root@cfg01:~# git add -A

root@cfg01:~# git commit -m 'Hello ver 4'
```

Step 8 Run your hello formula:

```
root@cfg01:~# salt "cfg01*" state.sls hello
```

Step 9 Verify the result:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 3
```

Step 10 It should be still older version, we need to regenerate the model in /srv/salt/reclass:

```
root@cfg01:~# salt-call state.sls reclass
```

Step 11 Verify if the model is regenerated:

```
root@cfg01:~# cat /srv/salt/reclass/classes/cluster/drivetrain-ha/infra/hello.yml

parameters:
   hello:
     version: 4
```

Step 12 Run your hello formula:

```
root@cfg01:~# salt "cfg01*" state.sls hello
```

Step 13 Verify the result:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 4
```

5.3.5. Use Jenkins GUI to run your hello formula

Step 1 Modify the formula metadata in the cluster model:

```
root@cfg01:~# cd /root/model_repo
```

Edit `/root/model_repo/classes/cluster/drivetrain-ha/infra/hello.yml`:

```
parameters:
  hello:
    version: 5
```

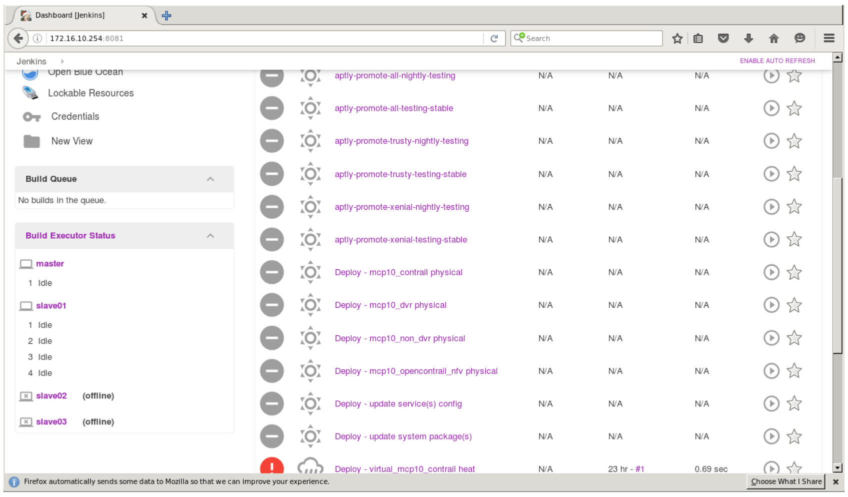
Step 2 Add and commit:

```
root@cfg01:~# git add -A

root@cfg01:~# git commit -m 'Hello ver 5'
```

Step 3 Open browser and navigate to http://172.16.10.104:8081

Use credentials **admin/password** to login and open list of pipelines:



Dashboard [Jenkins]

172.16.10.254:8081

Jenkins › ENABLE AUTO REFRESH

Open Blue Ocean
Lockable Resources
Credentials
New View

Build Queue

No builds in the queue.

Build Executor Status

master
1 Idle

slave01
1 Idle
2 Idle
3 Idle
4 Idle

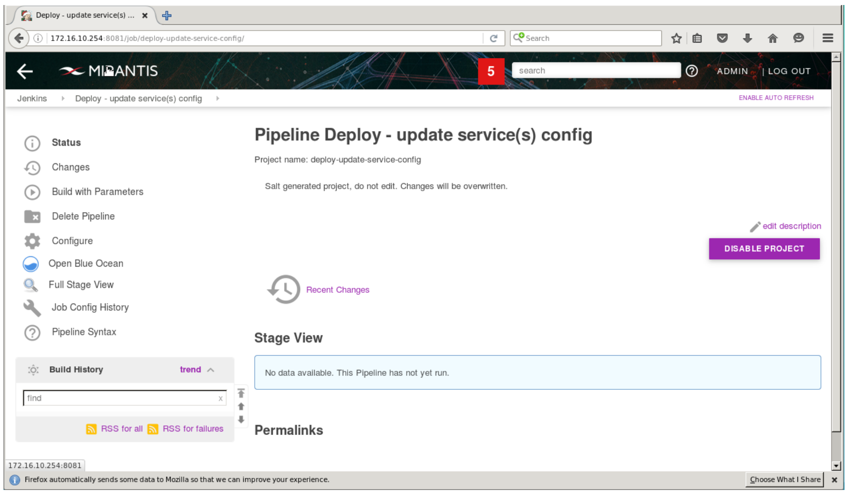
slave02 (offline)

slave03 (offline)

aptly-promote-all-nightly-testing	N/A	N/A	N/A
aptly-promote-all-testing-stable	N/A	N/A	N/A
aptly-promote-trusty-nightly-testing	N/A	N/A	N/A
aptly-promote-trusty-testing-stable	N/A	N/A	N/A
aptly-promote-xenial-nightly-testing	N/A	N/A	N/A
aptly-promote-xenial-testing-stable	N/A	N/A	N/A
Deploy - mcp10_contrail physical	N/A	N/A	N/A
Deploy - mcp10_dvr physical	N/A	N/A	N/A
Deploy - mcp10_non_dvr physical	N/A	N/A	N/A
Deploy - mcp10_opencontrail_nfv physical	N/A	N/A	N/A
Deploy - update service(s) config	N/A	N/A	N/A
Deploy - update system package(s)	N/A	N/A	N/A
Deploy - virtual_mcp10_contrail heat	N/A	23 hr - #1	0.69 sec

Firefox automatically sends some data to Mozilla so that we can improve your experience. Choose What I Share

Find Deploy - Update service(s) config job:



Deploy - update service(s) ...

172.16.10.254:8081/job/deploy-update-service-config/

MIRANTIS 5 search ADMIN | LOG OUT

Jenkins › Deploy - update service(s) config › ENABLE AUTO REFRESH

Status
Changes
Build with Parameters
Delete Pipeline
Configure
Open Blue Ocean
Full Stage View
Job Config History
Pipeline Syntax

Build History trend

find

RSS for all RSS for failures

Pipeline Deploy - update service(s) config

Project name: deploy-update-service-config

Salt generated project, do not edit. Changes will be overwritten.

edit description

DISABLE PROJECT

Recent Changes

Stage View

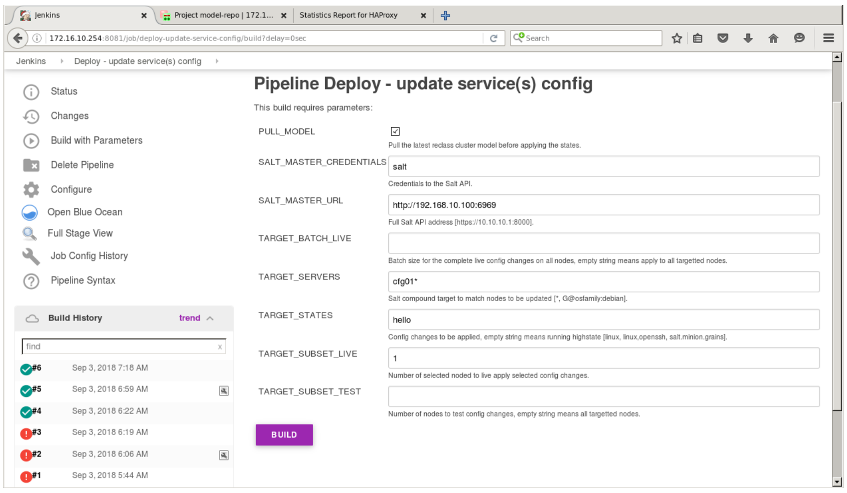
No data available. This Pipeline has not yet run.

Permalinks

172.16.10.254:8081

Firefox automatically sends some data to Mozilla so that we can improve your experience. Choose What I Share

Launch the job with following parameters:



Jenkins | Project model-repo | 172.1... | Statistics Report for HAProxy

172.16.10.254:8081/job/deploy-update-service-config/build?delay=0sec

Jenkins › Deploy - update service(s) config ›

Status
Changes
Build with Parameters
Delete Pipeline
Configure
Open Blue Ocean
Full Stage View
Job Config History
Pipeline Syntax

Build History trend

find

#6	Sep 3, 2018 7:18 AM
#5	Sep 3, 2018 6:59 AM
#4	Sep 3, 2018 6:22 AM
#3	Sep 3, 2018 6:19 AM
#2	Sep 3, 2018 6:06 AM
#1	Sep 3, 2018 5:44 AM

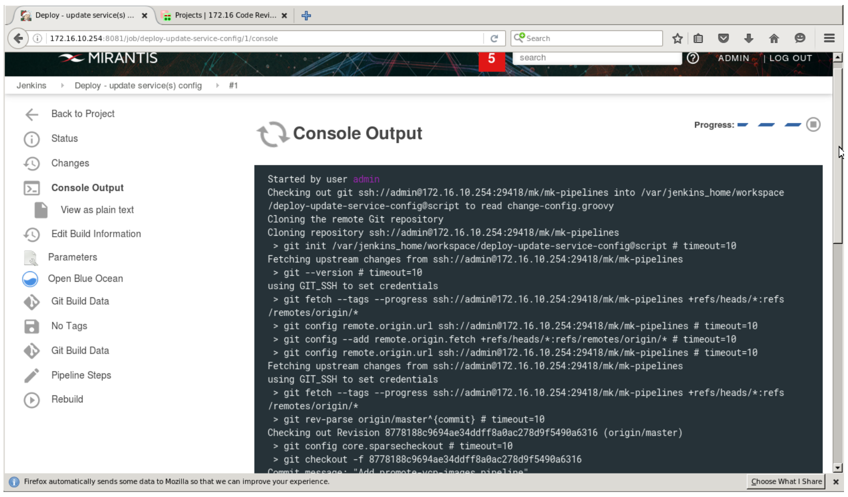
Pipeline Deploy - update service(s) config

This build requires parameters:

PULL_MODEL	☑ Pull the latest reclass cluster model before applying the states.
SALT_MASTER_CREDENTIALS	salt — Credentials to the Salt API.
SALT_MASTER_URL	http://192.168.10.100:6969 — Full Salt API address [https://10.10.10.1:8000].
TARGET_BATCH_LIVE	Batch size for the complete live config changes on all nodes, empty string means apply to all targetted nodes.
TARGET_SERVERS	cfg01* — Salt compound target to match nodes to be updated [*, G@osfamily:debian].
TARGET_STATES	hello — Config changes to be applied, empty string means running highstate [linux, linux,openssh, salt.minion.grains].
TARGET_SUBSET_LIVE	1 — Number of selected noded to live apply selected config changes.
TARGET_SUBSET_TEST	Number of nodes to test config changes, empty string means all targetted nodes.

BUILD

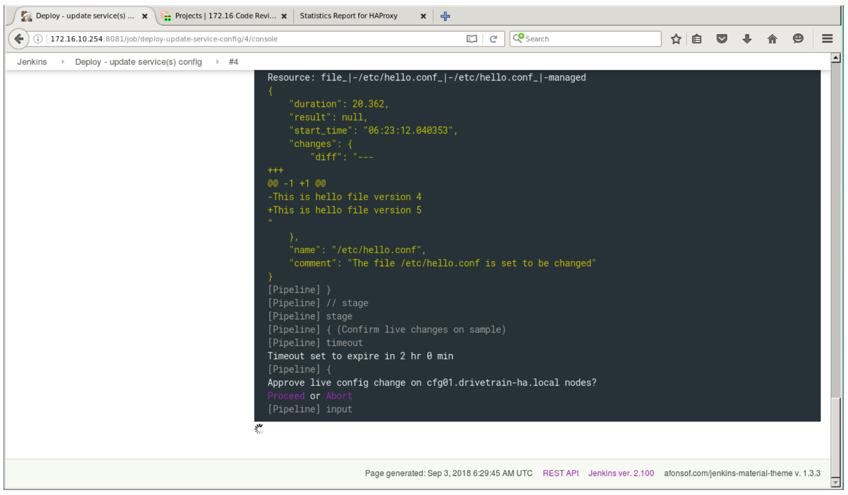
Open a console output to monitor pipeline execution:



```
Started by user admin
Checking out git ssh://admin@172.16.10.254:29418/mk/mk-pipelines into /var/jenkins_home/workspace
/deploy-update-service-config@script to read change-config.groovy
Cloning the remote Git repository
Cloning repository ssh://admin@172.16.10.254:29418/mk/mk-pipelines
 > git init /var/jenkins_home/workspace/deploy-update-service-config@script # timeout=10
Fetching upstream changes from ssh://admin@172.16.10.254:29418/mk/mk-pipelines
 > git --version # timeout=10
using GIT_SSH to set credentials
 > git fetch --tags --progress ssh://admin@172.16.10.254:29418/mk/mk-pipelines +refs/heads/*:refs
/remotes/origin/*
 > git config remote.origin.url ssh://admin@172.16.10.254:29418/mk/mk-pipelines # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url ssh://admin@172.16.10.254:29418/mk/mk-pipelines # timeout=10
Fetching upstream changes from ssh://admin@172.16.10.254:29418/mk/mk-pipelines
using GIT_SSH to set credentials
 > git fetch --tags --progress ssh://admin@172.16.10.254:29418/mk/mk-pipelines +refs/heads/*:refs
/remotes/origin/*
 > git rev-parse origin/master^{commit} # timeout=10
Checking out Revision 8778188c9694ae34ddff8a0ac278d9f5490a6316 (origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 8778188c9694ae34ddff8a0ac278d9f5490a6316
Commit message: "Add promote-vcp-images pipeline"
```

Firefox automatically sends some data to Mozilla so that we can improve your experience. Choose What I Share

When Pipeline asks to Approve config change click on **Approve**:


Deploy - update service(s) ...
Projects | 172.16 Code Revi...
Statistics Report for HAProxy
172.16.10.254:8081/job/deploy-update-service-config/4/console
Jenkins › Deploy - update service(s) config › #4

```
Resource: file_|-/etc/hello.conf_|-/etc/hello.conf_|-managed
{
    "duration": 20.362,
    "result": null,
    "start_time": "06:23:12.040353",
    "changes": {
        "diff": "---
+++
@@ -1 +1 @@
-This is hello file version 4
+This is hello file version 5
"
    },
    "name": "/etc/hello.conf",
    "comment": "The file /etc/hello.conf is set to be changed"
}
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Confirm live changes on sample)
[Pipeline] timeout
Timeout set to expire in 2 hr 0 min
[Pipeline] {
Approve live config change on cfg01.drivetrain-ha.local nodes?
Proceed or Abort
[Pipeline] input
```

Page generated: Sep 3, 2018 6:29:45 AM UTC REST API Jenkins ver. 2.100 afonsof.com/jenkins-material-theme v. 1.3.3

Step 4 Verify the result:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 5
```

5.3.6. Use Jenkins API to run your hello formula

Step 1 Modify the metadata used by your hello formula:

```
root@cfg01:~# cd /root/model_repo
```

Edit `/root/model_repo/classes/cluster/drivetrain-ha/infra/hello.yml`

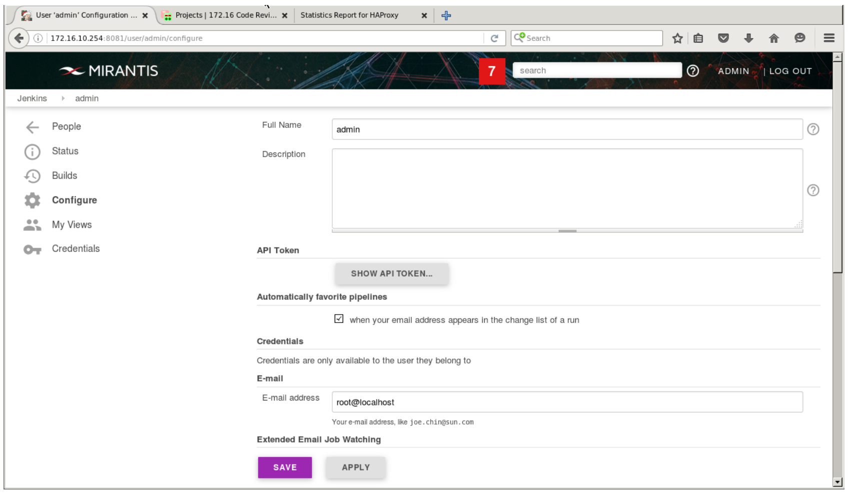
```
parameters:
  hello:
    version: 6
```

Step 2 Add and commit

```
root@cfg01:~# git add -A

root@cfg01:~# git commit -m 'Hello ver 6'
```

Step 3 Get the user token. Click on Admin button at the right-top. :



172.16.10.254:8081/user/admin/configure

Full Name: admin

Description:

API Token

Automatically favorite pipelines

☑ when your email address appears in the change list of a run

Credentials

Credentials are only available to the user they belong to

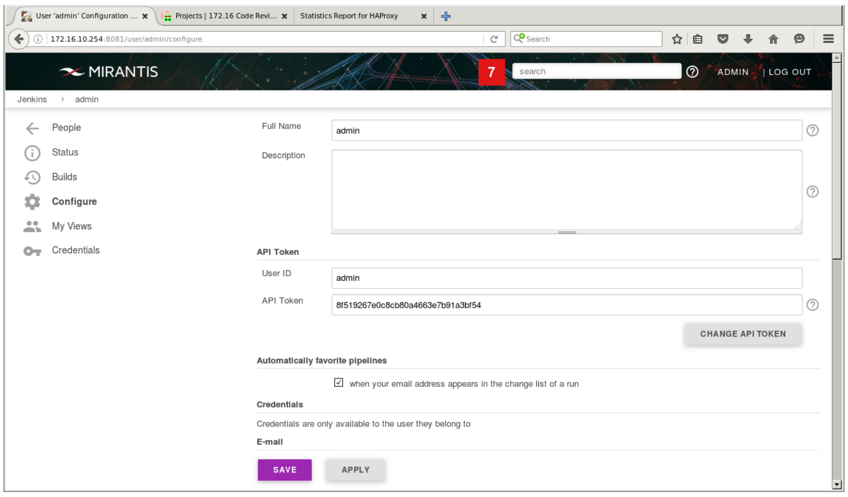
E-mail

E-mail address: root@localhost

Your e-mail address, like joe.chin@sun.com

Extended Email Job Watching

Open *Configure* tab and look for API token:



172.16.10.254:8081/user/admin/configure

Full Name: admin

Description:

API Token

User ID: admin

API Token: 8f519267e0c8cb80a4663e7b91a3bf54

Automatically favorite pipelines

☑ when your email address appears in the change list of a run

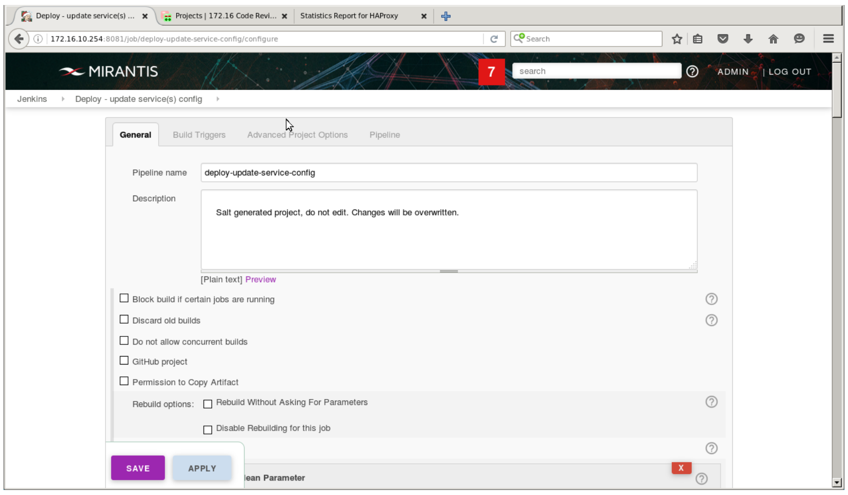
Credentials

Credentials are only available to the user they belong to

E-mail

In our case it’s admin/8f519267e0c8cb80a4663e7b91a3bf54

Step 4 Get back to the list of pipelines and open Configuration tab of pipeline “Deploy - Update service(s) config” and get the project/pipeline name:



172.16.10.254:8081/job/deploy-update-service-config/configure

Pipeline name: deploy-update-service-config

Description: Salt generated project, do not edit. Changes will be overwritten.

[Plain text] Preview

☐ Block build if certain jobs are running

☐ Discard old builds

☐ Do not allow concurrent builds

☐ GitHub project

☐ Permission to Copy Artifact

Rebuild options: ☐ Rebuild Without Asking For Parameters

☐ Disable Rebuilding for this job

ean Parameter

It should be *deploy-update-service-config*

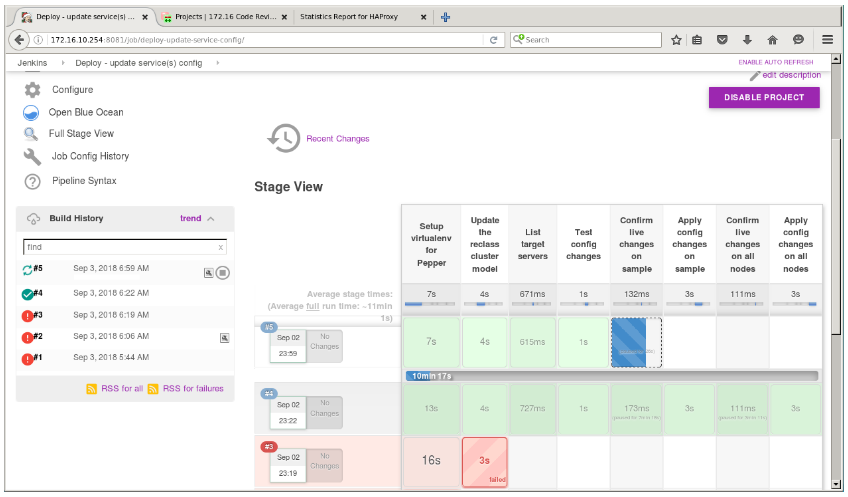
Step 5 Call the Jenkins API using userID/token/pipeline values:

```
root@cfg01:~# curl -d "TARGET_SERVERS=cfg01.drivetrain-ha.local" -d "TARGET_STATES=hell
-i -X POST http://admin:8f519267e0c8cb80a4663e7b91a3bf54@172.16.10.104:8081/job/deploy-
```

Remember this command - we will use it in next steps.

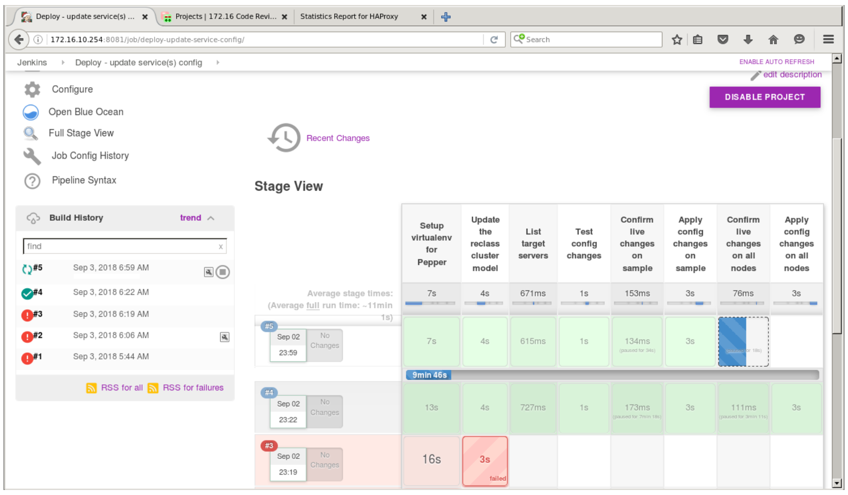
Step 6 Accept steps in the pipeline. This time you can do it on Stage View by clicking on Confirm changes stage:

Confirm live changes on sample:



Deploy - update service(s) ...
Projects | 172.16 Code Revi...
Statistics Report for HAProxy
172.16.10.254:8081/job/deploy-update-service-config/
Jenkins
Deploy - update service(s) config
ENABLE AUTO REFRESH
edit description
DISABLE PROJECT
Configure
Open Blue Ocean
Full Stage View
Job Config History
Pipeline Syntax
Recent Changes
Stage View
Build History
trend
RSS for all
RSS for failures
Setup virtualenv for Pepper
Update the reclass cluster model
List target servers
Test config changes
Confirm live changes on sample
Apply config changes on sample
Confirm live changes on all nodes
Apply config changes on all nodes
Average stage times:
(Average full run time: ~11min 1s)

Confirm live changes on all nodes:



Deploy - update service(s) ...
Projects | 172.16 Code Revi...
Statistics Report for HAProxy
172.16.10.254:8081/job/deploy-update-service-config/
Jenkins
Deploy - update service(s) config
ENABLE AUTO REFRESH
edit description
DISABLE PROJECT
Configure
Open Blue Ocean
Full Stage View
Job Config History
Pipeline Syntax
Recent Changes
Stage View
Build History
trend
RSS for all
RSS for failures
Setup virtualenv for Pepper
Update the reclass cluster model
List target servers
Test config changes
Confirm live changes on sample
Apply config changes on sample
Confirm live changes on all nodes
Apply config changes on all nodes
Average stage times:
(Average full run time: ~11min 1s)

Step 7 Verify the result:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 6
```

5.3.7. Modify your git repo to automatically call the Jenkins pipeline

Step 1 Add a post commit hook:

```
root@cfg01:~# cd /root/model_repo
```

Edit file `/root/model_repo/.git/hooks/post-commit`

```
#!/bin/bash
curl -d "TARGET_SERVERS=cfg01.drivetrain-ha.local" -d "TARGET_STATES=hello" -i -X POST
```

```
root@cfg01:~# chmod 755 .git/hooks/post-commit
```

Step 2 Modify the metadata used by your hello formula. Edit file `/root/model_repo/classes/cluster/drivetrain-ha/infra/hello.yml`

```
parameters:
  hello:
    version: 7
```

Step 3 Add and commit:

```
root@cfg01:~# git add -A

root@cfg01:~# git commit -m 'Hello ver 7'
```

Step 4 Accept steps in the pipeline

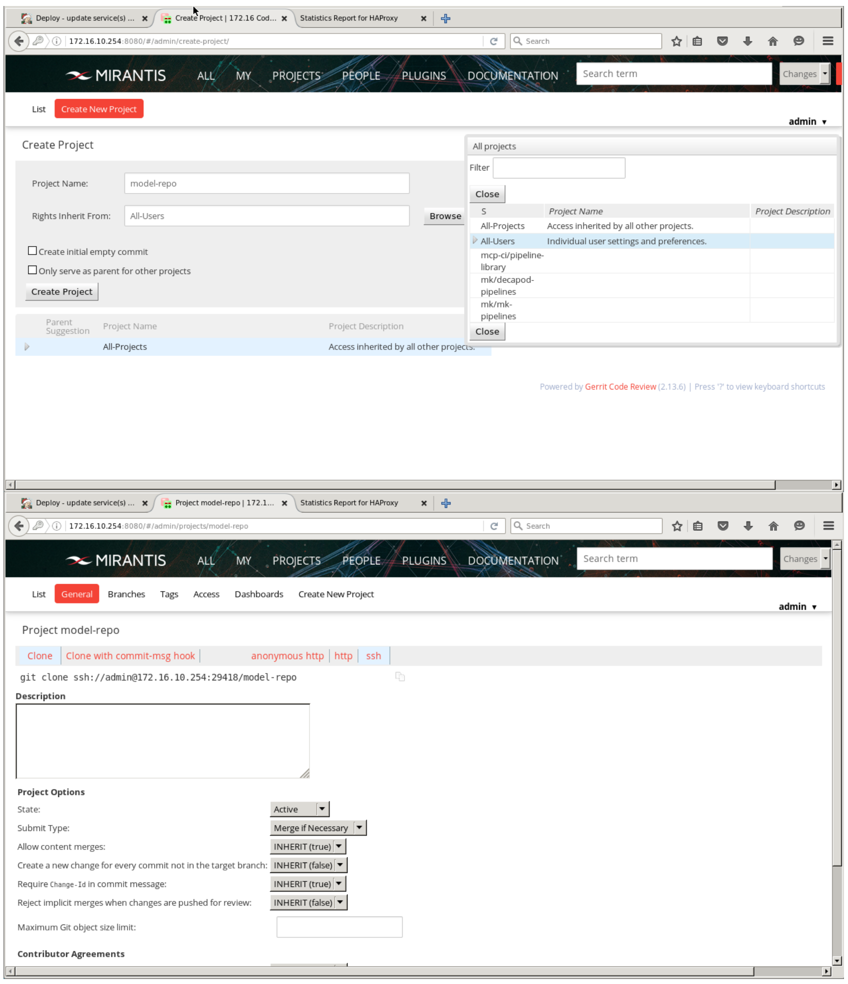
Step 5 Verify the results:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 7
```

5.3.8. Move to a shared repo in Gerrit

Step 1 Open browser and navigate to http://172.16.10.104:8080. Enter **admin/password** as credentials. Create project model-repo in Gerrit


Deploy - update service(s) ...
Create Project | 172.16 Cod...
Statistics Report for HAProxy
172.16.10.254:8080/#/admin/create-project/
Search
MIRANTIS
ALL
MY
PROJECTS
PEOPLE
PLUGINS
DOCUMENTATION
Search term
Changes
List
Create New Project
admin

Create Project

Project Name: model-repo

Rights Inherit From: All-Users Browse

☐ Create initial empty commit

☐ Only serve as parent for other projects

Create Project

Parent Suggestion	Project Name	Project Description
	All-Projects	Access inherited by all other projects.

All projects

Filter

Close

S	Project Name	Project Description
All-Projects	Access inherited by all other projects.	
All-Users	Individual user settings and preferences.	
mcp-ci/pipeline-library		
mk/decapod-pipelines		
mk/mk-pipelines		

Close

Powered by Gerrit Code Review (2.13.6) | Press '?' to view keyboard shortcuts

Deploy - update service(s) ...
Project model-repo | 172.1...
Statistics Report for HAProxy
172.16.10.254:8080/#/admin/projects/model-repo
Search
MIRANTIS
ALL
MY
PROJECTS
PEOPLE
PLUGINS
DOCUMENTATION
Search term
Changes
List
General
Branches
Tags
Access
Dashboards
Create New Project
admin

Project model-repo

Clone | Clone with commit-msg hook | anonymous http | http | ssh

```
git clone ssh://admin@172.16.10.254:29418/model-repo
```

Description

Project Options

State:	Active
Submit Type:	Merge if Necessary
Allow content merges:	INHERIT (true)
Create a new change for every commit not in the target branch:	INHERIT (false)
Require Change-Id in commit message:	INHERIT (true)
Reject implicit merges when changes are pushed for review:	INHERIT (false)
Maximum Git object size limit:	

Contributor Agreements

Step 2 Push to Gerrit repo:

```
root@cfg01:~# cd /root/model_repo/

root@cfg01:~# git remote add origin ssh://admin@172.16.10.104:29418/model-repo

root@cfg01:~# git push origin master

root@cfg01:~# rm .git/hooks/post-commit
```

Step 3 Modify your model to use Gerrit:

```
root@cfg01:~# cd /root/model_repo/
```

Edit `/srv/salt/reclass/classes/cluster/drivetrain-ha/infra/config/nodes.yml`:

```
parameters:
  reclass:
    storage:
      data_source:
        # don't use Git storage for model, instead use whatever is provided
        # with the lab
        # engine: local
        engine: git
        address: ssh://admin@172.16.10.104:29418/model-repo
        branch: master
        force_reset: true
```

Step 4 Add, commit and push to Gerrit project (master branch):

```
root@cfg01:~# git add -A

root@cfg01:~# git commit -m 'Switch storage to Gerrit'

root@cfg01:~# git push origin master
```

Step 5 Regenerate the model:

```
root@cfg01:~# salt-call state.sls reclass
```

Step 6 Check the result:

```
root@cfg01:~# cat /srv/salt/reclass/classes/cluster/drivetrain-ha/infra/config/nodes.ym
```

Step 7 Modify the metadata used by your hello formula:

```
root@cfg01:~# cd /root/model_repo/
```

Edit `/root/model_repo/classes/cluster/drivetrain-ha/infra/hello.yml`:

```
parameters:
  hello:
    version: 8
```

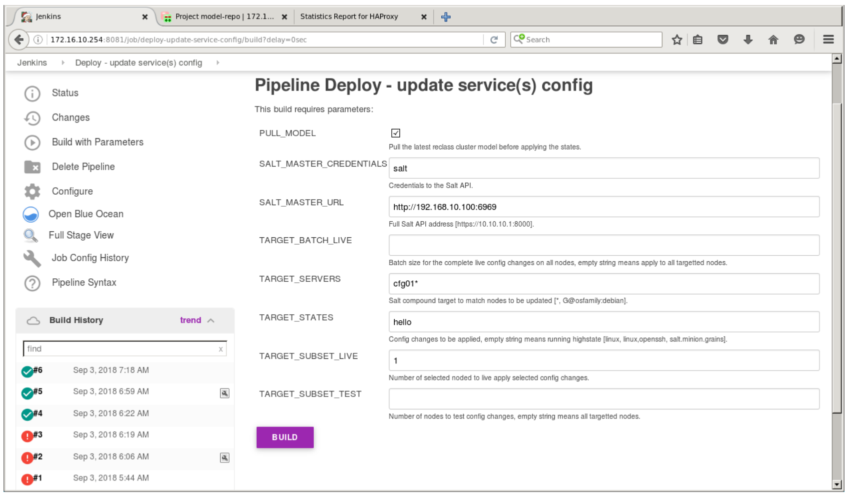
Step 8 Add, commit and push to Gerrit project (master branch):

```
root@cfg01:~# git add -A

root@cfg01:~# git commit -m 'Hello ver 8'

root@cfg01:~# git push origin master
```

Step 9 Jenkins GUI - Build with Parameters:



172.16.10.254:8081/job/deploy-update-service-config/build?delay=0sec

Jenkins › Deploy - update service(s) config ›

Pipeline Deploy - update service(s) config

This build requires parameters:

PULL_MODEL	☑ Pull the latest reclass cluster model before applying the states.
SALT_MASTER_CREDENTIALS	salt — Credentials to the Salt API.
SALT_MASTER_URL	http://192.168.10.100:6969 — Full Salt API address [https://10.10.10.1:8000].
TARGET_BATCH_LIVE	Batch size for the complete live config changes on all nodes, empty string means apply to all targetted nodes.
TARGET_SERVERS	cfg01* — Salt compound target to match nodes to be updated [*, G@osfamily:debian].
TARGET_STATES	hello — Config changes to be applied, empty string means running highstate [linux, linux,openssh, salt.minion.grains].
TARGET_SUBSET_LIVE	1 — Number of selected noded to live apply selected config changes.
TARGET_SUBSET_TEST	Number of nodes to test config changes, empty string means all targetted nodes.

Build History

#6	Sep 3, 2018 7:18 AM
#5	Sep 3, 2018 6:59 AM
#4	Sep 3, 2018 6:22 AM
#3	Sep 3, 2018 6:19 AM
#2	Sep 3, 2018 6:06 AM
#1	Sep 3, 2018 5:44 AM

Step 10 Verify the result:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 8
```

5.3.9. Use gerrit branching and review GUI

Step 1 Install git hook from Gerrit:

```
root@cfg01:~# cd /root/model_repo/

root@cfg01:~# scp -p -P 29418 admin@172.16.10.104:hooks/commit-msg .git/hooks/
```

Step 2 Modify the metadata used by your hello formula. Edit `/root/model_repo/classes/cluster/drivetrain-ha/infra/hello.yml`:

```
parameters:
    hello:
        version: 9
```

Step 3 Add, commit and push to Gerrit project, but this time review branch:

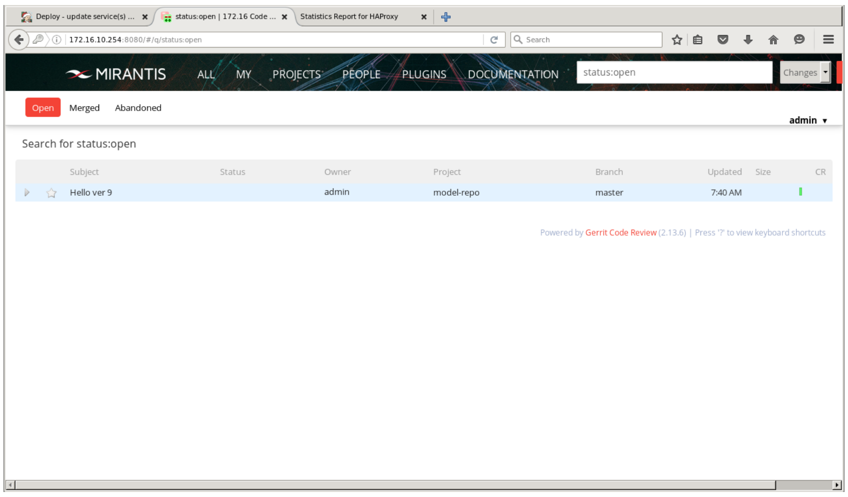
```
root@cfg01:~# git add -A

root@cfg01:~# git commit -m 'Hello ver 9'

root@cfg01:~# git push origin HEAD:refs/for/master
```

Step 4 Go to Gerrit - Review and Submit:

Find your changes in Gerrit:

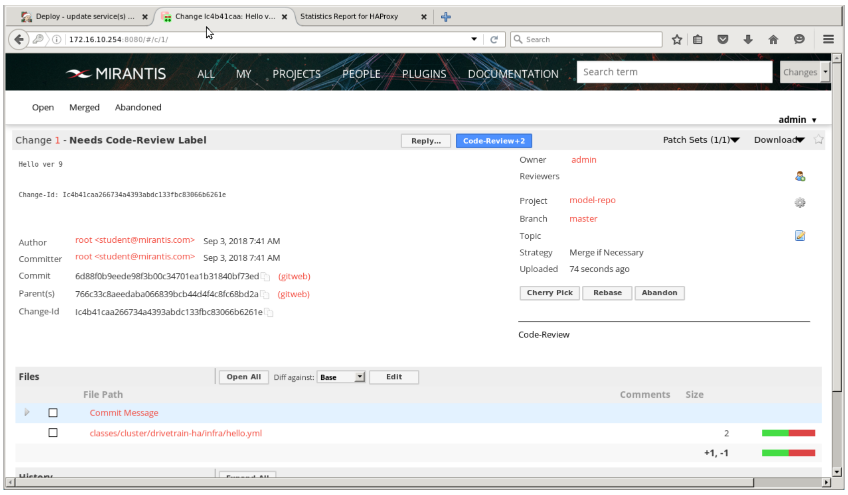

Deploy - update service(s) ...
status:open | 172.16 Code ...
Statistics Report for HAProxy
172.16.10.254:8080/#/q/status:open
MIRANTIS
ALL
MY
PROJECTS
PEOPLE
PLUGINS
DOCUMENTATION
status:open
Changes
Open
Merged
Abandoned
admin

Search for status:open

Subject	Status	Owner	Project	Branch	Updated	Size	CR
Hello ver 9		admin	model-repo	master	7:40 AM		

Powered by Gerrit Code Review (2.13.6) | Press '?' to view keyboard shortcuts

To approve the changes you need to press a “Code-Review +2” button:



Change 1 - Needs Code-Review Label

```
Hello ver 9

Change-Id: Ic4b41caa266734a4393abdc133fbc83066b6261e
```

Author	root <student@mirantis.com>	Sep 3, 2018 7:41 AM
Committer	root <student@mirantis.com>	Sep 3, 2018 7:41 AM
Commit	6d88f0b9eede98f3b00c34701ea1b31840bf73ed (gitweb)	
Parent(s)	766c33c8aeedaba066839bcb44d4f4c8fc68bd2a (gitweb)	
Change-Id	Ic4b41caa266734a4393abdc133fbc83066b6261e	

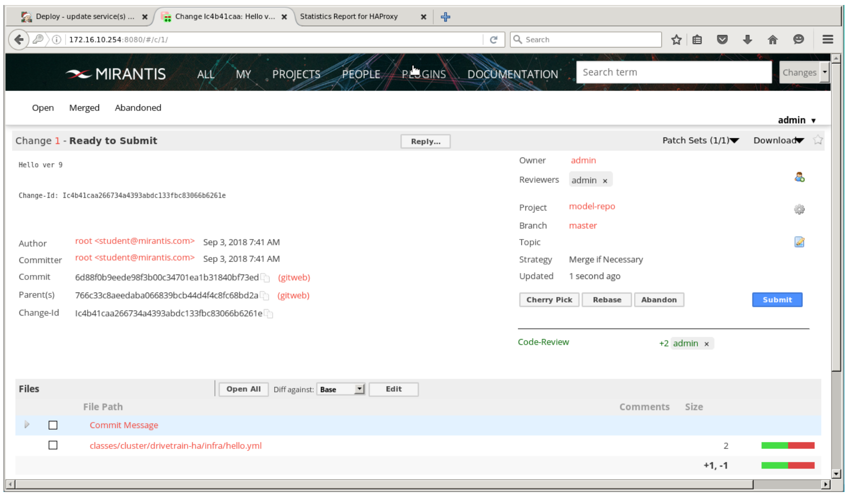
Owner	admin
Reviewers	
Project	model-repo
Branch	master
Topic	
Strategy	Merge if Necessary
Uploaded	74 seconds ago

Code-Review

Files

File Path	Comments	Size
Commit Message		
classes/cluster/drivetrain-ha/infra/hello.yml		2
		+1, -1

After changes were improved press “Submit” button:



Change 1 - Ready to Submit

```
Hello ver 9

Change-Id: Ic4b41caa266734a4393abdc133fbc83066b6261e
```

Author	root <student@mirantis.com>	Sep 3, 2018 7:41 AM
Committer	root <student@mirantis.com>	Sep 3, 2018 7:41 AM
Commit	6d88f0b9eede98f3b00c34701ea1b31840bf73ed (gitweb)	
Parent(s)	766c33c8aeedaba066839bcb44d4f4c8fc68bd2a (gitweb)	
Change-Id	Ic4b41caa266734a4393abdc133fbc83066b6261e	

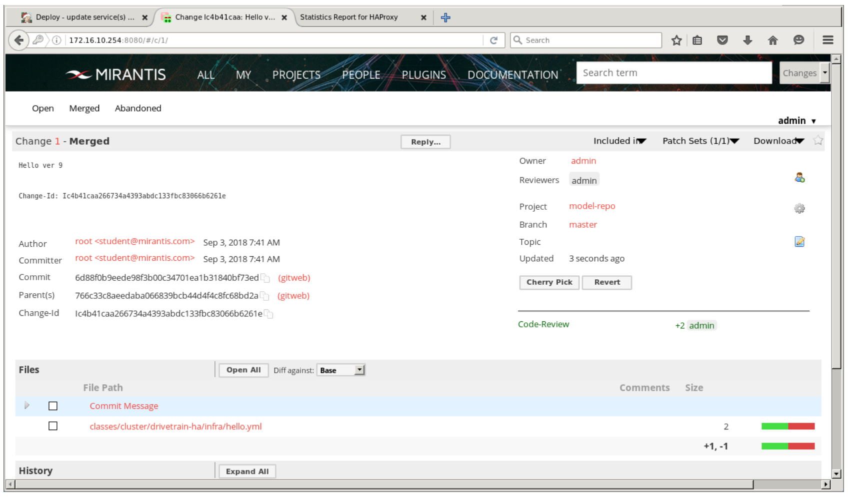
Owner	admin
Reviewers	admin
Project	model-repo
Branch	master
Topic	
Strategy	Merge if Necessary
Updated	1 second ago

Code-Review +2 admin

Files

File Path	Comments	Size
Commit Message		
classes/cluster/drivetrain-ha/infra/hello.yml		2
		+1, -1

Confirm that changes were merged:



Deploy - update service(s) ... | Change Ic4b41caa: Hello v... | Statistics Report for HAProxy

172.16.10.254:8080/#/c/1/

Open Merged Abandoned

admin

Change 1 - Merged

```
Hello ver 9

Change-Id: Ic4b41caa266734a4393abdc133fbc83066b6261e
```

Author	root <student@mirantis.com>	Sep 3, 2018 7:41 AM
Committer	root <student@mirantis.com>	Sep 3, 2018 7:41 AM
Commit	6d88f0b9eede98f3b00c34701ea1b31840bf73ed	(gitweb)
Parent(s)	766c33c8aeedaba066839bcb44d4f4c8fc68bd2a	(gitweb)
Change-Id	Ic4b41caa266734a4393abdc133fbc83066b6261e	

Owner	admin
Reviewers	admin
Project	model-repo
Branch	master
Topic	
Updated	3 seconds ago

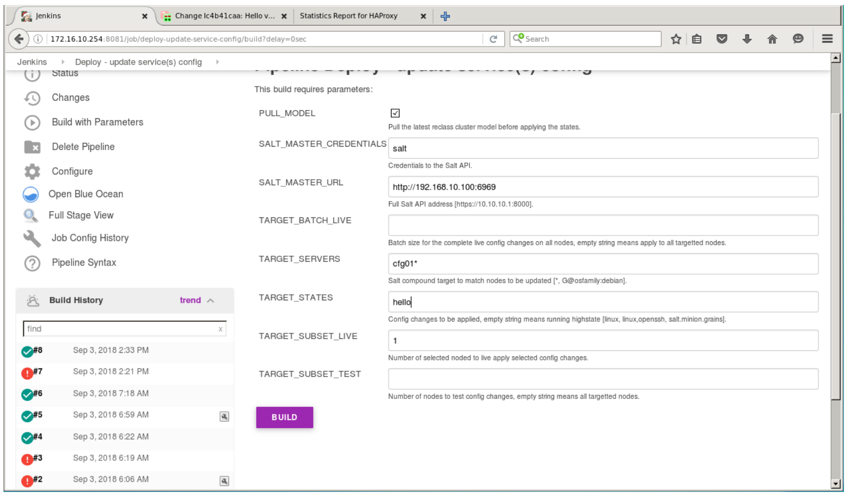
Code-Review +2 admin

Files

File Path	Comments	Size
Commit Message		
classes/cluster/drivetrain-ha/infra/hello.yml		2
		+1, -1

History

Step 5 Jenkins GUI - Build with Parameters:



Jenkins | Change Ic4b41caa: Hello v... | Statistics Report for HAProxy

172.16.10.254:8081/job/deploy-update-service-config/build?delay=0sec

Jenkins › Deploy - update service(s) config ›

This build requires parameters:

PULL_MODEL	✓ Pull the latest reclass cluster model before applying the states.
SALT_MASTER_CREDENTIALS	salt Credentials to the Salt API.
SALT_MASTER_URL	http://192.168.10.100:6969 Full Salt API address [https://10.10.10.1:8000].
TARGET_BATCH_LIVE	 Batch size for the complete live config changes on all nodes, empty string means apply to all targetted nodes.
TARGET_SERVERS	cfg01* Salt compound target to match nodes to be updated [*, G@osfamily:debian].
TARGET_STATES	hello Config changes to be applied, empty string means running highstate [linux, linux,openssh, salt.minion.grains].
TARGET_SUBSET_LIVE	1 Number of selected noded to live apply selected config changes.
TARGET_SUBSET_TEST	 Number of nodes to test config changes, empty string means all targetted nodes.

Build History

#8	Sep 3, 2018 2:33 PM
#7	Sep 3, 2018 2:21 PM
#6	Sep 3, 2018 7:18 AM
#5	Sep 3, 2018 6:59 AM
#4	Sep 3, 2018 6:22 AM
#3	Sep 3, 2018 6:19 AM
#2	Sep 3, 2018 6:06 AM

Step 6 Verify the result:

```
root@cfg01:~# cat /etc/hello.conf

This is hello file version 9
```