Stateless Image Validation and Deployment at CU

Leveraging Systemd, Ansible, and CI practices to provide policy compliant OS images for computational clusters

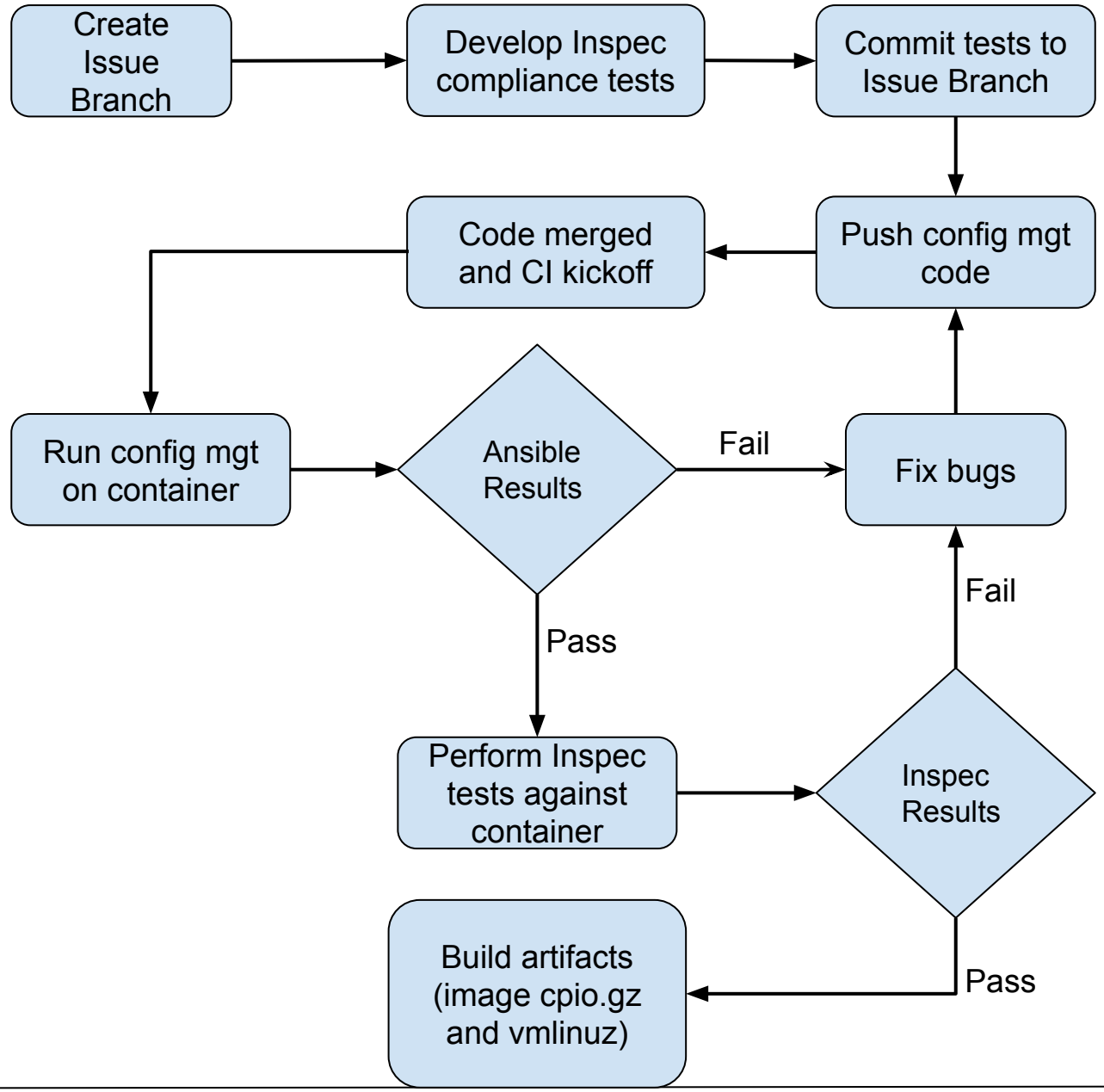
John Blaas
University of Colorado Boulder

Where it started

- OS image/directories as git repos
  - Worked well but can lead to fairly unwieldy git repos
  - Can’t be shared with others
  - Unencrypted secrets in the repo =(
- Used chroot to apply updates and configuration against the image directories
  - For some updates this also required us to bind mount certain filesystems from the host OS as well
- Very little testing after an image had been built - relied on deploying an image to staging hardware first to reveal any issues.

New process

- Policy defined for how computes should look and behave is fully documented in the compliance tests
- Use containers to run an entire OS to check the configuration code deployment against, and subsequently compliance tests to verify configuration code deployment and sanity
- Use Ansible to deploy configuration management, secrets are kept in repo but are now encrypted at rest.
- Use previously defined compliance tests that specify our environment policy to transition from one configuration language to another (Puppet -> Ansible)


Create Issue Branch
Develop Inspec compliance tests
Commit tests to Issue Branch
Push config mgt code
Code merged and CI kickoff
Run config mgt on container
Ansible Results
Fail
Fix bugs
Pass
Fail
Perform Inspec tests against container
Inspec Results
Build artifacts (image cpio.gz and vmlinuz)
Pass

What is systemd-nspawn?

- Improved chroot, no need to bind mount the usual suspects
- Tool that spawns unique namespaces
- Designed with building, testing, and debugging in mind
- Incredibly simple to get started with

```
> rpm -i --root=/tmp/centos7 centos7-release.rpm
> yum --installroot=/tmp/centos7 groupinstall Base
> systemd-nspawn -bD /tmp/centos7
```

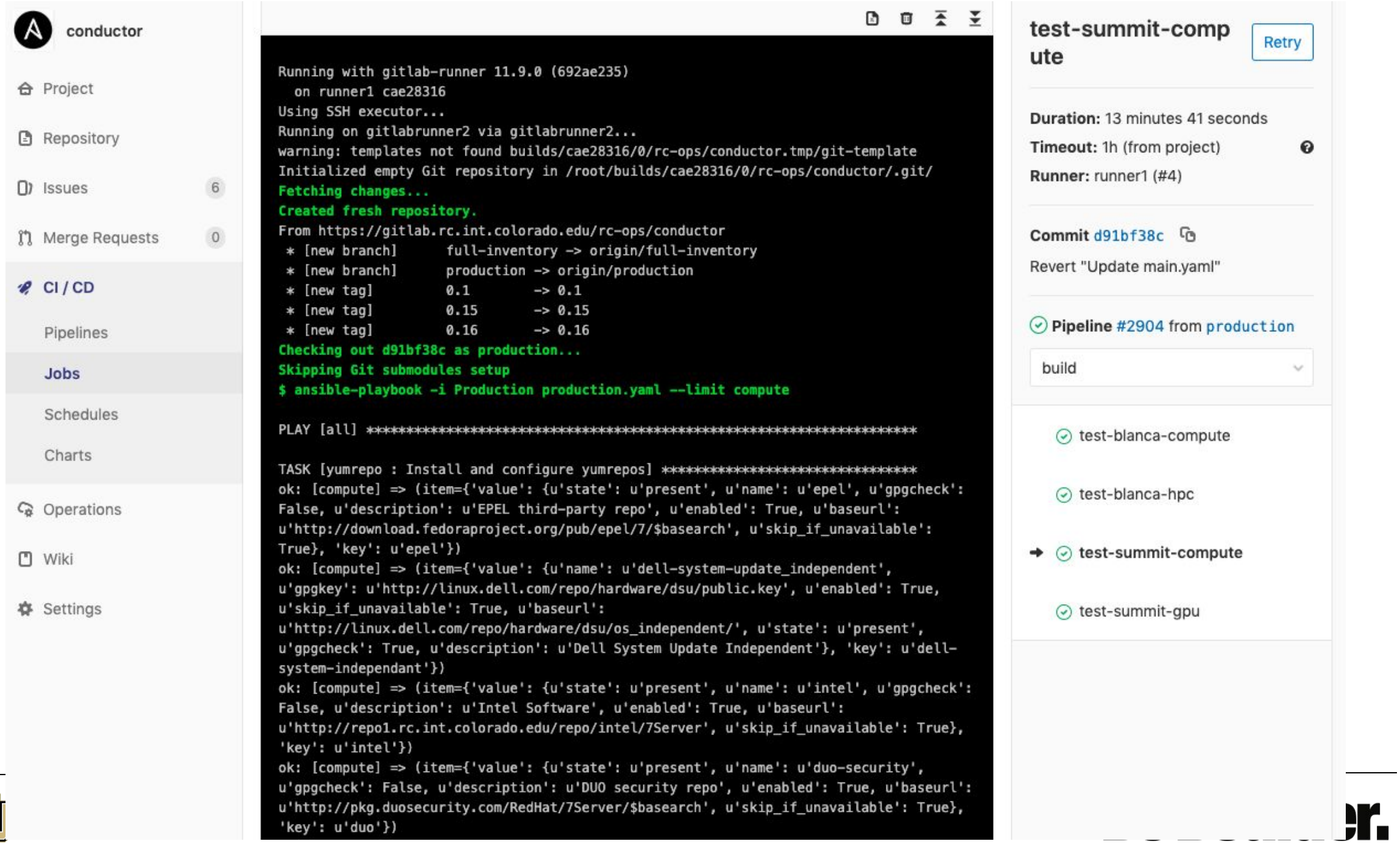
What is Inspec?

- Compliance testing framework based on Serverspec
- Designed to have clear syntax and be platform agnostic
- Describes tests as a collection of controls which can be grouped into compliance profiles
- Allows you to define policy for the state of a node
- Tests can be run locally or with SSH
- Originally developed by the Chef team

```
44
45  control "beegfs-4" do
46    impact 1.0
47    title "Beegfs client config check"
48    desc "Check to ensure that the client configuration is in place"
49    describe file ('/etc/beegfs/beegfs-client.conf') do
50      it { should exist }
51      it { should be_file }
52      its('owner') { should eq 'root' }
53      its('group') { should eq 'root' }
54      its('mode') { should cmp '0644' }
55      its('content') { should include 'sysMgmtdHost                = 10.225.144.131' }
56      its('content') { should include 'connRDMABufSize             = 8192' }
57      its('content') { should include 'connUseRDMA                 = true' }
58    end
59  end
60
61  control "beegfs-5" do
62    impact 1.0
63    title "Beegfs mount configuration check"
64    desc "Checks that beegfs mounts configuration file is in place"
65    describe file ('/etc/beegfs/beegfs-mounts.conf') do
66      it { should exist }
67      it { should be_file }
68      its('owner') { should eq 'root' }
69      its('group') { should eq 'root' }
70      its('mode') { should cmp '0644' }
71      its('content') { should include '/beegfs/pl-active /etc/beegfs/beegfs-client.conf beegfs rw' }
72    end
73  end
74
75  control "beegfs-6" do
76    impact 1.0
77    title "Beegfs helperd service file check"
```

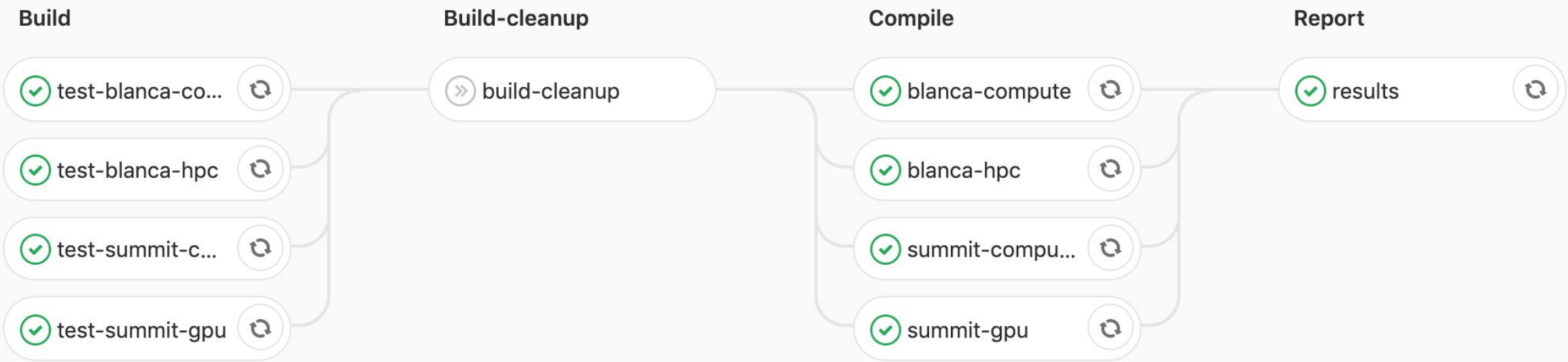
Gitlab CI

- We use Gitlab to manage all of our internal repositories so using Gitlab runner is leveraged to provide a CI environment
- We make heavy use of concurrent pipelines to build all stateless node images at once
- We make use of reporting stages to sync up image build and compile stages
- We do not use any of the continuous delivery components at this time.


conductor
Project
Repository
Issues 6
Merge Requests 0
CI / CD
Pipelines
Jobs
Schedules
Charts
Operations
Wiki
Settings
Running with gitlab-runner 11.9.0 (692ae235)
on runner1 cae28316
Using SSH executor...
Running on gitlabrunner2 via gitlabrunner2...
warning: templates not found builds/cae28316/0/rc-ops/conductor.tmp/git-template
Initialized empty Git repository in /root/builds/cae28316/0/rc-ops/conductor/.git/
Fetching changes...
Created fresh repository.
From https://gitlab.rc.int.colorado.edu/rc-ops/conductor
* [new branch] full-inventory -> origin/full-inventory
* [new branch] production -> origin/production
* [new tag] 0.1 -> 0.1
* [new tag] 0.15 -> 0.15
* [new tag] 0.16 -> 0.16
Checking out d91bf38c as production...
Skipping Git submodules setup
$ ansible-playbook -i Production production.yaml --limit compute
PLAY [all] **********
TASK [yumrepo : Install and configure yumrepos] **********
ok: [compute] => (item={'value': {u'state': u'present', u'name': u'epel', u'gpgcheck': False, u'description': u'EPEL third-party repo', u'enabled': True, u'baseurl': u'http://download.fedoraproject.org/pub/epel/7/$basearch', u'skip_if_unavailable': True}, 'key': u'epel'})
ok: [compute] => (item={'value': {u'name': u'dell-system-update_independent', u'gpgkey': u'http://linux.dell.com/repo/hardware/dsu/public.key', u'enabled': True, u'skip_if_unavailable': True, u'baseurl': u'http://linux.dell.com/repo/hardware/dsu/os_independent/', u'state': u'present', u'gpgcheck': True, u'description': u'Dell System Update Independent'}, 'key': u'dell-system-independant'})
ok: [compute] => (item={'value': {u'state': u'present', u'name': u'intel', u'gpgcheck': False, u'description': u'Intel Software', u'enabled': True, u'baseurl': u'http://repo1.rc.int.colorado.edu/repo/intel/7Server', u'skip_if_unavailable': True}, 'key': u'intel'})
ok: [compute] => (item={'value': {u'state': u'present', u'name': u'duo-security', u'gpgcheck': False, u'description': u'DUO security repo', u'enabled': True, u'baseurl': u'http://pkg.duosecurity.com/RedHat/7Server/$basearch', u'skip_if_unavailable': True}, 'key': u'duo'})
test-summit-compute
Retry
Duration: 13 minutes 41 seconds
Timeout: 1h (from project)
Runner: runner1 (#4)
Commit d91bf38c
Revert "Update main.yaml"
Pipeline #2904 from production
build
test-blanca-compute
test-blanca-hpc
test-summit-compute
test-summit-gpu

d91bf38c

Pipeline | Jobs 12



passed **Job #5768** triggered 5 days ago by John Blaas

```
Running with gitlab-runner 11.9.0 (692ae235)
  on runner1 cae28316
Using SSH executor...
Running on gitlabrunner2 via gitlabrunner2...
warning: templates not found builds/cae28316/3/rc-ops/conductor.tmp/git-template
Initialized empty Git repository in /root/builds/cae28316/3/rc-ops/conductor/.git/
Fetching changes...
Created fresh repository.
From https://gitlab.rc.int.colorado.edu/rc-ops/conductor
 * [new branch]      full-inventory -> origin/full-inventory
 * [new branch]      production -> origin/production
 * [new tag]         0.1        -> 0.1
 * [new tag]         0.15       -> 0.15
 * [new tag]         0.16       -> 0.16
Checking out 479e3a25 as production...
Skipping Git submodules setup
$ cd /containers/bhpc
$ echo $CI_COMMIT_SHA
479e3a252ab6bfcae437f296e59bb2f3508118a2
$ echo $CI_COMMIT_REF_NAME $CI_COMMIT_SHORT_SHA > /containers/bhpc/etc/image-release
$ cp /containers/compute/boot/vmlinuz-* /images/vmlinuz-blanca-bhpc-$CI_COMMIT_SHORT_SHA
$ find | cpio -oc | gzip > /images/blanca-bhpc-$CI_COMMIT_SHORT_SHA.cpio.gz
4742993 blocks
Skipping cache archiving due to empty cache key
Job succeeded
```

blanca-hpc

Retry

Duration: 6 minutes 24 seconds

Timeout: 1h (from project)

Runner: runner1 (#4)

Commit 479e3a25

Update fstab.conf.j2

Pipeline #2894 from production

compile

- blanca-compute
- **blanca-hpc**
- summit-compute
- summit-gpu

Deploying the image

- We have Foreman provision the node as if it were a stateful node to fall back on, but nodes intending to be run in a stateless manner get a few extra parameters assigned
- These parameters tell Foreman if the node is stateless or not, what cluster it belongs to, and node type or group(ie. GPU)
- Our default PXELinux global profile assigns the right PXElinux.cfg file based on the parameters in Foreman allowing for booting up stateful or stateless nodes.

Foreman parameters we use

stateless (True, False/Not specified)

cluster (Summit, Blanca)

type (compute, gpu, bhpc)

release (Short SHA commit value, latest)

Release can be set to the short shas to test an image via host parameter on a subset of nodes first, after testing the new image can be symlinked to latest

Global Parameters

Name	Value	Actions
activation_key	research-computing-c31c1bc9-6449-4c20-bd70-621da87fd6a8	Override
cluster	blanca	Override
datacenter	hpcf	Override
kickstart-packages	-abrt*	Override
puppetmaster	foreman.rc.int.colorado.edu	Override
~~release~~	latest	
ssh_authorized_keys	ssh-rsa	Override
stateless	true	Override
subscription_manager	true	Override
subscription_manager_org	3585003	Override
subscription_manager_repos	rhel-7-server-optional-rpms,rhel-7-server-extras-rpms,rhel-7-server-supplementary-rpms	Override
time-zone	America/Denver	Override
type	bhpc	Override

Host Parameters

Name	Value	Actions
release	479e3a25	Remove

+ Add Parameter

```
[root@foreman curc]# ls -al
total 2898300
drwxr-xr-x 2 root root       4096 May 16 11:39 .
drwxr-xr-x 8 root root       4096 May  1 07:37 ..
-rw-r--r-- 1 root root  873473277 May 16 10:53 blanca-bhpc-479e3a25.cpio.gz
lrwxrwxrwx 1 root root         28 May 16 11:39 blanca-bhpc-latest.cpio.gz -> blanca-bhpc-479e3a25.cpio.gz
-rw-r--r-- 1 root root  931441720 May 14 17:43 summit-compute-f2666987.cpio.gz
lrwxrwxrwx 1 root root         31 May 14 17:44 summit-compute-latest.cpio.gz -> summit-compute-f2666987.cpio.gz
-rw-r--r-- 1 root root 1109662455 May 14 17:43 summit-gpu-f2666987.cpio.gz
lrwxrwxrwx 1 root root         27 May 15 12:45 summit-gpu-latest.cpio.gz -> summit-gpu-f2666987.cpio.gz
-rwxr-xr-x 1 root root    5917504 May 16 10:54 vmlinuz-blanca-bhpc-479e3a25
lrwxrwxrwx 1 root root         28 May 16 11:39 vmlinuz-blanca-bhpc-latest -> vmlinuz-blanca-bhpc-479e3a25
-rwxr-xr-x 1 root root    5917504 May 16 10:54 vmlinuz-blanca-compute-479e3a25
-rwxr-xr-x 1 root root    5917504 May 14 17:43 vmlinuz-blanca-compute-f2666987
-rwxr-xr-x 1 root root    5917504 May 16 10:54 vmlinuz-summit-compute-479e3a25
-rwxr-xr-x 1 root root    5917504 May 16 09:34 vmlinuz-summit-compute-71a2db4d
-rwxr-xr-x 1 root root    5917504 May 14 17:43 vmlinuz-summit-compute-f2666987
lrwxrwxrwx 1 root root         31 May 14 17:45 vmlinuz-summit-compute-latest -> vmlinuz-summit-compute-f2666987
-rwxr-xr-x 1 root root    5917504 May 16 10:54 vmlinuz-summit-gpu-479e3a25
-rwxr-xr-x 1 root root    5917504 May 16 09:34 vmlinuz-summit-gpu-71a2db4d
-rwxr-xr-x 1 root root    5917504 May 14 17:43 vmlinuz-summit-gpu-f2666987
lrwxrwxrwx 1 root root         27 May 15 12:44 vmlinuz-summit-gpu-latest -> vmlinuz-summit-gpu-f2666987
```

Rolling update NHC check

- We deploy new images in a rolling update fashion for components of the stack that don't require synchronization
- We use a NHC check that consults a node's State and Reason to determine if it is safe to reboot the node into a new image

```
scontrol update NodeName=shas01[01-60] State=DRAIN Reason=”update”
```

- When the node goes into a state of DRAIN+IDLE the check executes and reboots the node into the new image
- When the node comes back up again it has to pass NHC before it can be marked online again

2:35 PM **Adam Selene** APP

Rebooting node shas0120.rc.int.colorado.edu
I am starting the reboot process on this node since its Reason was marked as update

Rebooting node shas0114.rc.int.colorado.edu
I am starting the reboot process on this node since its Reason was marked as update

Rebooting node shas0121.rc.int.colorado.edu
I am starting the reboot process on this node since its Reason was marked as update

2:41 PM **Adam Selene** APP

Releasing node shas0114.rc.int.colorado.edu back to production
I am returning this node back to production following a successful reboot

Releasing node shas0121.rc.int.colorado.edu back to production
I am returning this node back to production following a successful reboot

Future Work

- Better support for building all images from a clean container/ OS directory (can simply copy a clean base directory tree, leverage overlay FS, --volatile, still investigating)
- Developing training repo that can be used to train new system administrators in configuration management
- Publish Inspec compliance profiles that help define our computing environment, standard set of attributes

Thank you

References

[SC18 SIGHPC-Syspros - Stateless Provisioning: Modern Practice in HPC]

[NFSroot from the CHAOS project (LLNL)]

[Socket Activated containers in Systemd]

[Foremen Template examples from OSC]

[Inspec]