



CentOS

The CentOS CI: A Getting Started Guide

CentOS Dojo - Brussels: 2016-01-29

Dusty Mabe

Software Container Engineer, Red Hat

dusty@dustymabe.com



CentOS

whoami

- Dusty Mabe
 - Software Container Engineer – Red Hat
 - Fedora Cloud WG Member
 - CentOS Atomic SIG Member
 - Previously, Platform Consultant – Red Hat
 - Previously, Software Engineer – Tekelec



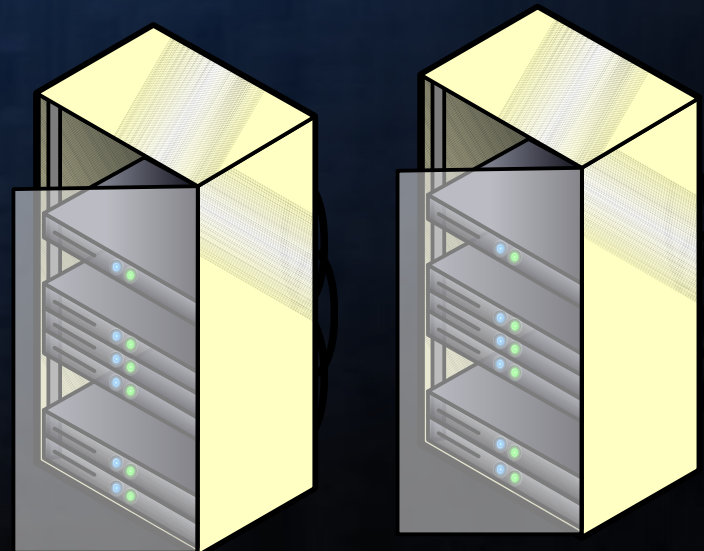
Agenda

- Background of the CentOS CI
- A Recipe For Your First Test: Web UI
- A Recipe For Your Next Test: JJB
- Final Thoughts and Future Plans



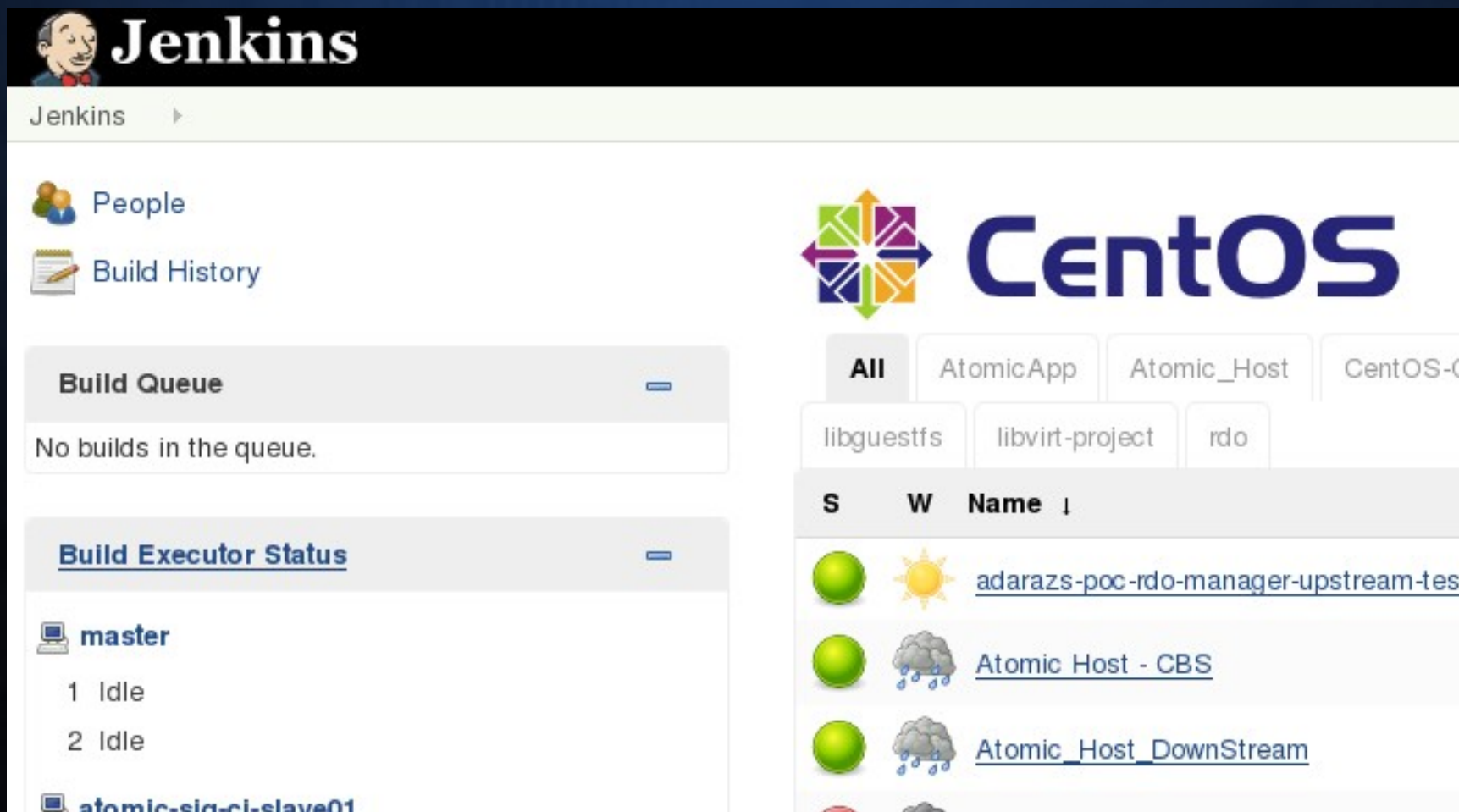
What is the CentOS CI Infrastructure?

- 256 physical machines spread across 4 chassis
 - physical machines == bare metal
 - More information at <https://wiki.centos.org/QaWiki/PubHardware>
- Bare metal allows for testing of unique workloads
 - Allows for testing of Virtualization Technology



What is the CentOS CI Infrastructure?

- Jenkins Frontend for ci.centos.org



The screenshot displays the Jenkins web interface. At the top left is the Jenkins logo and name. Below it, there are navigation links for 'People' and 'Build History'. The main content area is divided into several sections:

- Build Queue:** A section with a minus sign icon, containing the text 'No builds in the queue.'
- Build Executor Status:** A section with a minus sign icon, showing the status of various executors. Under the 'master' node, there are two 'Idle' executors. Under the 'atomic-sig-ci-slave01' node, there are two 'Idle' executors.
- CentOS Logo and Filtered Build List:** On the right side, there is a large CentOS logo. Below it, there are filter tabs for 'All', 'AtomicApp', 'Atomic_Host', and 'CentOS-C'. Underneath these are more specific filters: 'libguestfs', 'libvirt-project', and 'rdo'. Below the filters is a table of build jobs.

S	W	Name ↓
●	☀	adarazs-poc-rdo-manager-upstream-test
●	☁	Atomic Host - CBS
●	☁	Atomic_Host_DownStream



Who is the CentOS CI for?

- Upstream Projects/Communities
- Requirements:
 - Project Must Be Re-Distributable
 - Open Source is a +!
 - Project Must Utilize CentOS as a Platform



Who is currently using the CI?

- RDO Project – Openstack Testing
- Libvirt Project
- Libguestfs Project
- Foreman Project
- Project Atomic (Atomic Host & Atomic App)
- Software Collections
- CentOS

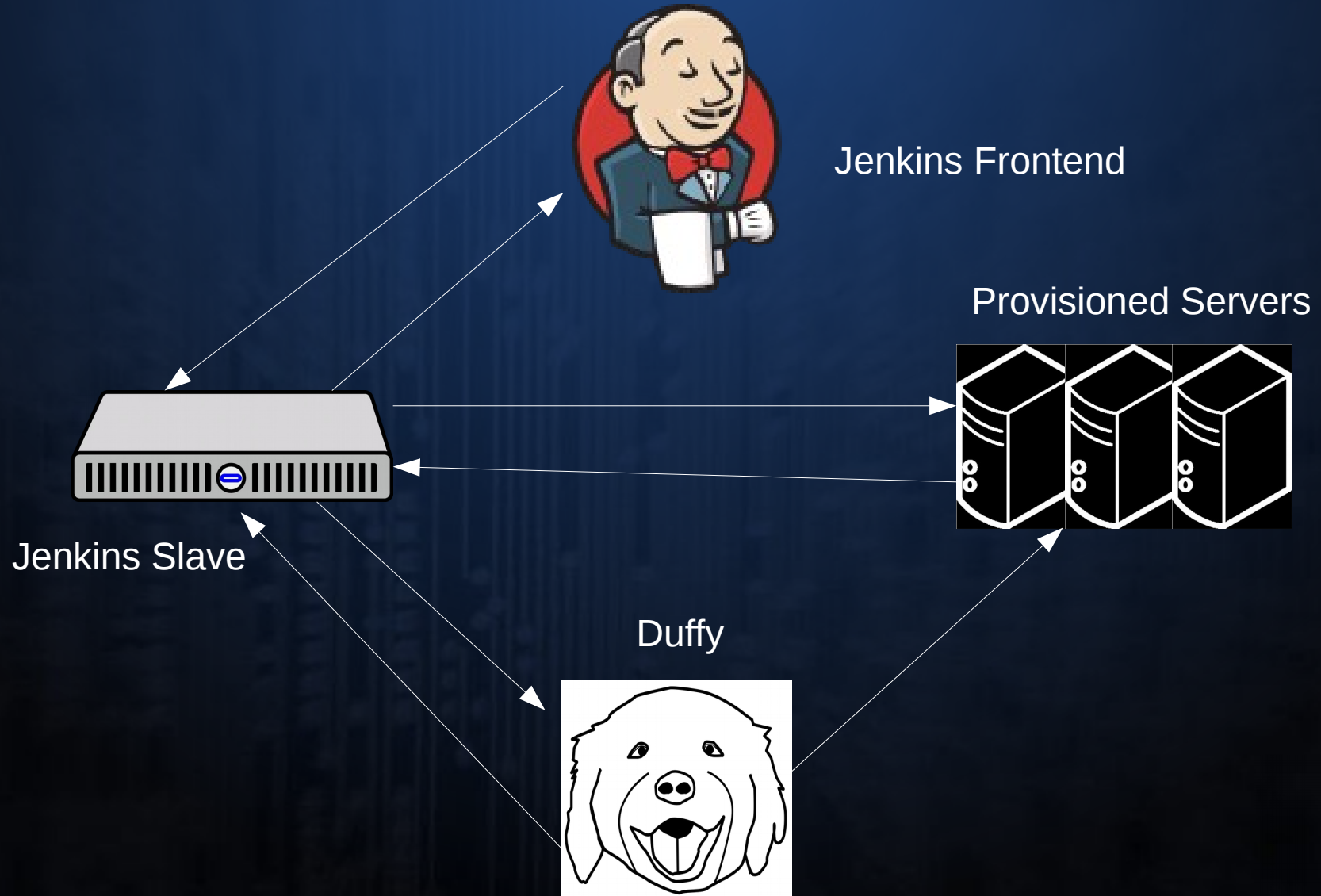


Testing Architecture

- What happens when I run a test?
 - Jenkins contacts a slave and executes commands
 - These commands should talk to Duffy to provision machines.
 - After machine provisioning, the slave can then execute tests on the provisioned machines.



Testing Architecture



Agenda

- Background of the CentOS CI
- A Recipe For Your First Test: Web UI
- A Recipe For Your Next Test: JJB
- Final Thoughts and Future Plans



Step 1: Get Credentials

- When you gain access to the CI you will get:
 - A username/password for the Jenkins frontend
 - An API key to use with Duffy
 - A target slave type to be used for your testing



Step 2: Git Repo With Test

- Create a test script to run
 - Install Software
 - Set up Machine for Test
 - Execute Test
- Place it in a git repo that can be cloned
 - This will be the source of the tests that get run



Step 3: Create a new Job

- “New Item” → “Freestyle Project”



The screenshot shows the Jenkins web interface. At the top left is the Jenkins logo (a cartoon man) and the word "Jenkins". To the right is a search bar with a magnifying glass icon and the word "search". Below the header is a breadcrumb trail: "Jenkins" > "All" > ".". The main content area is divided into two columns. The left column contains three menu items: "New Item" with a yellow folder icon, "People" with a blue and green people icon, and "Build History" with a white notepad icon. The right column shows the "New Item" form. It has a label "Item name" and a text input field containing "dusty-job". Below the input field is a radio button icon and the text "Freestyle project". To the right of this is a description: "This is the central point for managing all the jobs in the system with any build system".



Step 4: Configure Job

- Check “Restrict where this project can be run”
 - Enter label for your slave type

Restrict where this project can be run

Label Expression

Slaves in [label](#): 1

Step 4: Configure Job

- Check “Inject environment variables to the build process” under “Build Environment”
 - Populate these environment variables
 - GIT_REPO_URL – The git repo where your test lives
 - TEST_CMD – The command to execute from within the git repo
 - API_KEY – The Duffy API Key that was provided to you



Step 4: Configure Job

Build Environment

- Delete workspace before build starts
- Abort the build if it's stuck
- Add timestamps to the Console Output
- Color ANSI Console Output
- Inject environment variables to the build process

Properties File Path

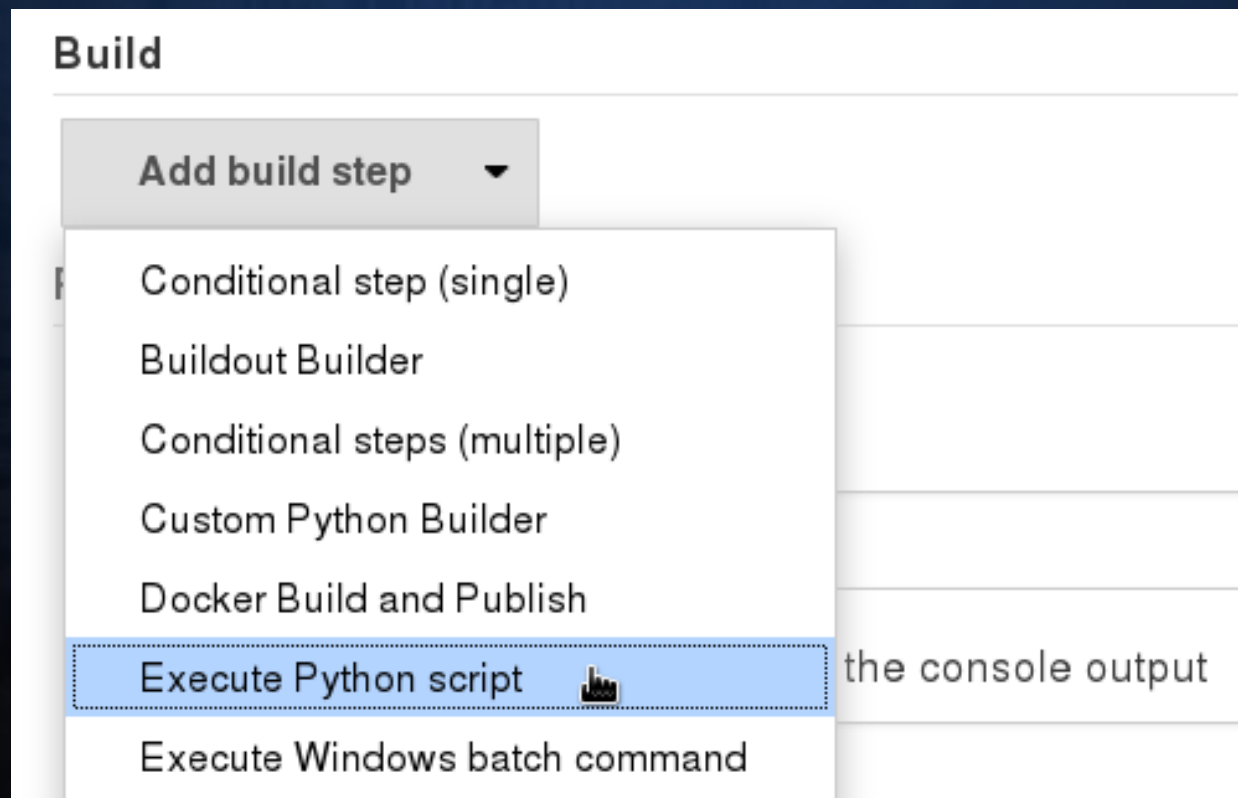
Properties Content

```
API_KEY=aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeeeee|  
MACHINE_COUNT=1  
TEST_CMD='./run_tests.sh'  
GIT_REPO_URL='https://github.com/dustymabe/centos-ci-example.git'
```



Step 4: Configure Job

- Click on the “Add Build Step” dropdown and select “Execute Python Script”



Step 4: Configure Job

- Populate A Python Script in the Text Box
 - Retrieve python script from following URL:
 - <https://github.com/dustymabe/centos-ci-example/blob/master/jjb/run.py>

```
Build
```

```
Execute Python script
```

Script

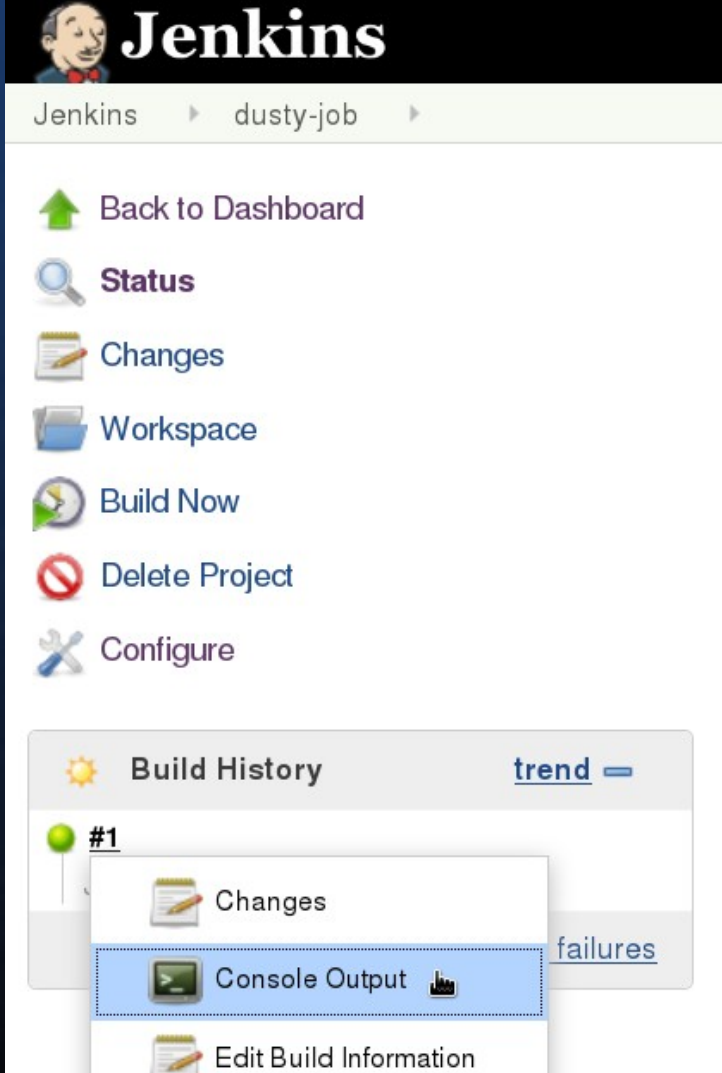
```
#!/usr/bin/python
import json, urllib, subprocess, sys, os

# We will interface with Duffy to request machines:
# Duffy documentation at https://wiki.centos.org/QaWiki/CI/Duffy

# Formulate the url to request nodes
url base = "http://admin.ci.centos.org:8080"
```

Step 5: Run Job

- Now You Can Run Your Job!
 - Click “Build Now” and then view “Console Output”



The screenshot shows the Jenkins web interface for a job named "dusty-job". The page has a dark blue header with the Jenkins logo and the job name. Below the header, there is a navigation menu with the following items:

- Back to Dashboard (green arrow icon)
- Status (magnifying glass icon)
- Changes (notepad icon)
- Workspace (folder icon)
- Build Now (play button icon)
- Delete Project (red prohibition sign icon)
- Configure (wrench icon)

Below the navigation menu, there is a "Build History" section with a "trend" dropdown menu. The build history shows a single build, #1, which is currently in a "success" state (green circle). A context menu is open over the build #1, showing the following options:

- Changes (notepad icon)
- Console Output (terminal icon) - This option is highlighted with a blue border.
- Edit Build Information (notepad icon)

There is also a "failures" link visible on the right side of the build history section.

Agenda

- Background of the CentOS CI
- A Recipe For Your First Test: Web UI
- A Recipe For Your Next Test: JJB
- Final Thoughts and Future Plans



Using Jenkins Job Builder

- Automate creating/updating jobs
 - <http://docs.openstack.org/infra/jenkins-job-builder/>



Using Jenkins Job Builder

- Installing

- # yum install -y /usr/bin/jenkins-jobs

- Making a config file

- # cat <<EOF > jenkins_jobs.ini

- [jenkins]

- user=username

- password=password

- url=https://ci.centos.org

- EOF



Using Jenkins Job Builder

- Create Job description

```
- # curl http://dustymabe.com/content/2016-01-23/run.py > run.py
- # cat <<EOF >job.yaml
  - job:
    name: dusty-ci-example
    node: atomicapp-shared
    builders:
      - inject:
        properties-content: |
          API_KEY=aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee
          MACHINE_COUNT=1
          TEST_CMD='./run_tests.sh'
          GIT_REPO_URL='https://github.com/dustymabe/centos-ci-example.git'
      - centos-ci-bootstrap
  - builder:
    name: centos-ci-bootstrap
    builders:
      - python:
        !include-raw: './run.py'
```

EOF



CentOS

Using Jenkins Job Builder

- Update/Create Jobs in Jenkins

- ```
jenkins-jobs --conf jenkins_jobs.ini update job.yaml
INFO:root:Updating jobs in ['job.yaml'] ([])
INFO:jenkins_jobs.local_yaml:Including file './run.py' from path '.'
INFO:jenkins_jobs.builder:Number of jobs generated: 1
INFO:jenkins_jobs.builder:Reconfiguring jenkins job dusty-ci-example
INFO:root:Number of jobs updated: 1
INFO:jenkins_jobs.builder:Cache saved
```





# Agenda

- Background of the CentOS CI
- A Recipe For Your First Test: Web UI
- A Recipe For Your Next Test: JJB
- Final Thoughts and Future Plans



# Where do I sign up?

- Please see our Getting Started wiki page
  - <https://wiki.centos.org/QaWiki/CI/GettingStarted>
  - Look at the “Asking for your project to be added” Section
- Every project has different needs
  - Start a conversation with us to see if the CI is right for your project
  - [ci-users@centos.org](mailto:ci-users@centos.org) or #centos-devel on Freenode



# Future Plans

- Clustering of instances
  - Allows for setting up private networks
- Openstack instances
  - Allows for testing “cloud” workloads
- Alternate Architectures
  - ARM
  - and beyond!



# Links and Pointers

- CentOS Project:
  - [centos.org](http://centos.org)
- CI Getting Started Wiki:
  - <https://wiki.centos.org/QaWiki/CI/GettingStarted>
- This presentation:
  - As a blog post:
    - <http://dustymabe.com/2016/01/23/the-centos-ci-infrastructure-a-getting-started-guide/>



# Demo (Time Permitting)



Thank you :-)

Any questions?



CentOS