



redhat®

CI PROVISIONING AND AUTOMATION

USING ANSIBLE

CLINT SAVAGE - CONTINUOUS INFRASTRUCTURE

DAVID ROBLE - CENTRAL CI

PREVIOUSLY

THERE WAS PROVISIONER 1.0

NOT SIMPLE TO USE

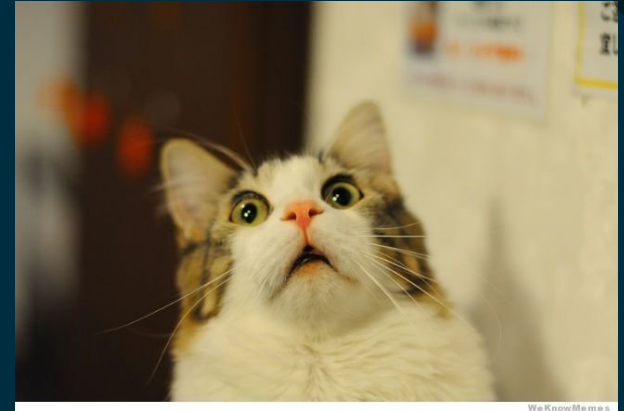
CI-FACTORY, CI-OPS-CENTRAL (AND
OTHER DERIVATIVES)

POWERFUL!
CUMBERSOME!
COMPLEX!



INSTALLATION

docs



```
mkdir -p <source code directory>; cd <source code directory>
git clone https://code.engineering.redhat.com/gerrit/ci-ops-central
cd ci-ops-central
sudo ./install.sh

#!/bin/bash

if grep -q 'Red Hat Enterprise Linux' /etc/redhat-release; then
    # Determine Version of RHEL
    export MAJOR_VER=$(egrep ' 6| 7' /etc/redhat-release | awk '{print $7}' | cut -d= -f1)
    export MINOR_VER=$(egrep ' 6| 7' /etc/redhat-release | awk '{print $7}')

    if [ "$MAJOR_VER" == "6" ]; then
        echo -n "Release and Optional Repos"
        export RHEL_RELEASE=http://download.eng.bos.redhat.com/released/RHEL-$MAJOR_VER
        export RHEL_OPTIONAL=http://download.lab.bos.redhat.com/rel-eng/latest-RHEL-$MAJOR_VER
        export PKG_LIST='git python-unittest2 python-nose python-futures
python-paramiko python-lxml python-six python-configobj python-pip
python-argparse python-glanceclient python-keystoneclient
python-novaclient gcc compat-gcc-34.x86_64 libffi-devel python-devel'
```

CLI

docs

```
ci-ops-central/bootstrap/provision_jslave.sh \  
--site=ci-osp \  
--project_defaults=/path/to/project_defaults \  
--topology=ci-ops-central/project/config/aio_jslave \  
--ssh_keyfile=/path/to/keyfile \  
--jslavename=jslave-projex-slave \  
--jslaveflavor=m1.xlarge \  
--jslaveimage=rhel-7.1-server-x86_64-released \  
--jslave_execs=10 --jslavecreate \  
--resources_file=jslave-projex-slave.json
```

UMM...



NO THANKS!

SIMPLE IS BETTER

- CLEANER INSTALLATION
- SIMPLE TOPOLOGIES
- CLOUD PLUGINS
- SIMPLE COMMAND LINE (VAGRANT-LIKE)
- POWERFUL EXTENSIBILITY
- SIMPLE PROVISION/TEARDOWN
- COMPLETE INVENTORIES
- AND MUCH, MUCH MORE!





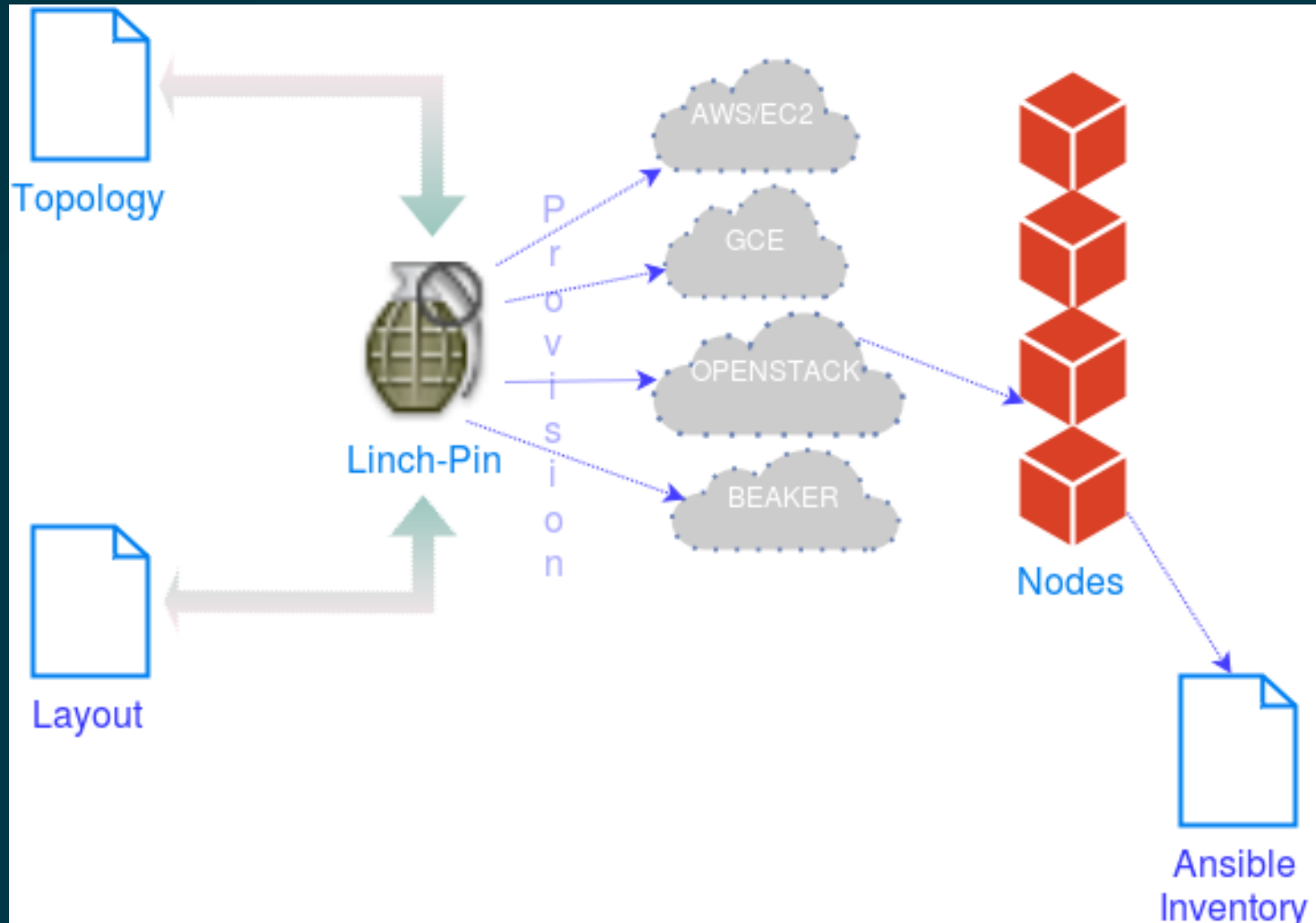
EXTENSIBLE,
MULTI-CLOUD,
HYBRID PROVISIONER

WRITTEN IN ANSIBLE

ENTER LINCH-
PIN



LINCH-PIN FLOW



WHY ANSIBLE?

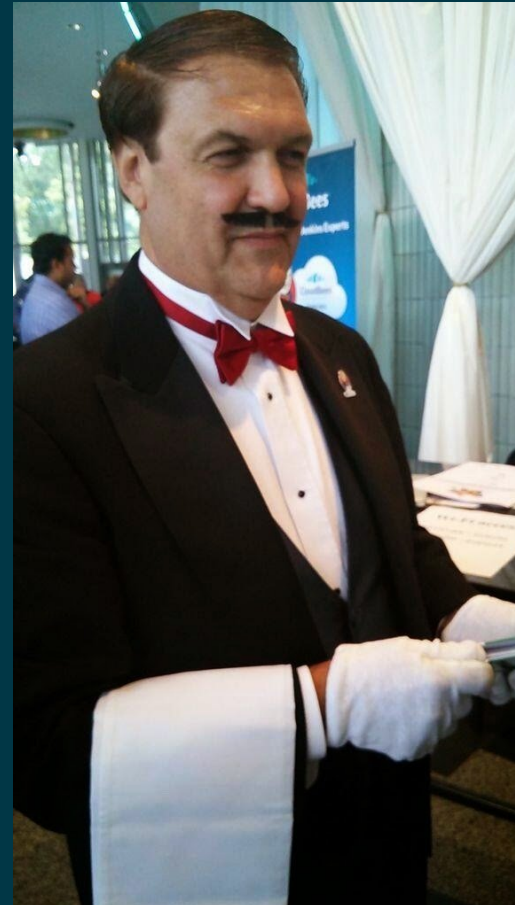


- Better logging and exception handling
- Cloud Modules
- Ansible is a Red Hat product (dogfooding)
- Asynchronous
- Good docs



JENKINS SLAVE
CONFIGURATION FOR
LINCH-PIN

LEVERAGING
LINCH-PIN IS
A **C**INCH



WHAT IS CINCH?

- linch-pin provisioning + automated jenkins slave configuration
- open source, hosted on GitHub for easier use for upstream
- optimized for the Central CI use case
- all using Ansible!

WHY TWO PROJECTS?

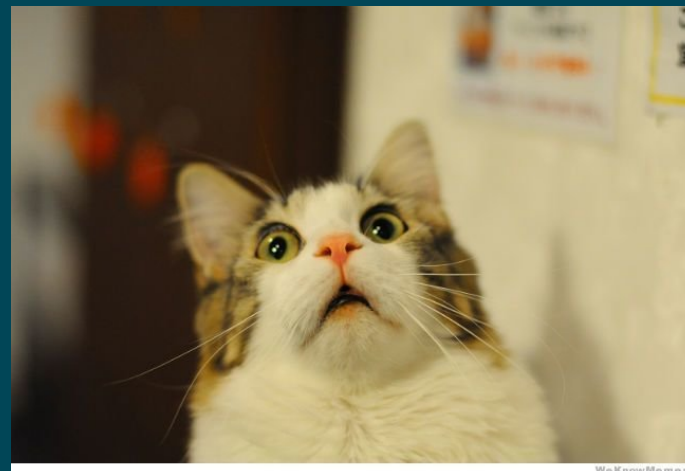
- linch-pin handles provisioning
- cinch optionally handles Jenkins slave configuration after provisioning
- not everyone needs the Jenkins slave configuration feature
- separation of concerns

SIMPLE INSTALLATION*

```
$ sudo dnf install -y libvirt-devel python-virtualenv \
    libyaml-devel openssl-devel libffi-devel git gcc \
    redhat-rpm-config
$ virtualenv cinchpin && source cinchpin/bin/activate
$ pip install git+https://github.com/CentOS-PaaS-SIG/linch-pin
$ pip install git+https://github.com/RedHatQE/cinch
```

* Pip & RPM packages are on the roadmap, installation will be simpler in the future

NOT SIMPLE INSTALLATION



```
mkdir -p <source code directory>; cd <source code directory>
git clone https://code.engineering.redhat.com/gerrit/ci-ops-central
cd ci-ops-central
sudo ./install.sh

#!/bin/bash

if grep -q 'Red Hat Enterprise Linux' /etc/redhat-release; then
    # Determine Version of RHEL
    export MAJOR_VER=$(egrep ' 6| 7' /etc/redhat-release | awk '{print $7}' | cut -d' ' -f1)
    export MINOR_VER=$(egrep ' 6| 7' /etc/redhat-release | awk '{print $7}' | cut -d' ' -f2)

    if [ "$MAJOR_VER" == "6" ]; then
        echo -n "Release and Optional Repos"
        export RHEL_RELEASE=http://download.eng.bos.redhat.com/released/RHEL-$MAJOR_VER-$MINOR_VER
        export RHEL_OPTIONAL=http://download.lab.bos.redhat.com/rel-eng/latest-RHEL-$MAJOR_VER-$MINOR_VER
        export PKG_LIST='git python-unittest2 python-nose python-futures
python-paramiko python-lxml python-six python-configobj python-pip
python-argparse python-glanceclient python-keystoneclient
python-novaclient gcc compat-gcc-34.x86_64 libffi-devel python-devel'
```


THIS IS

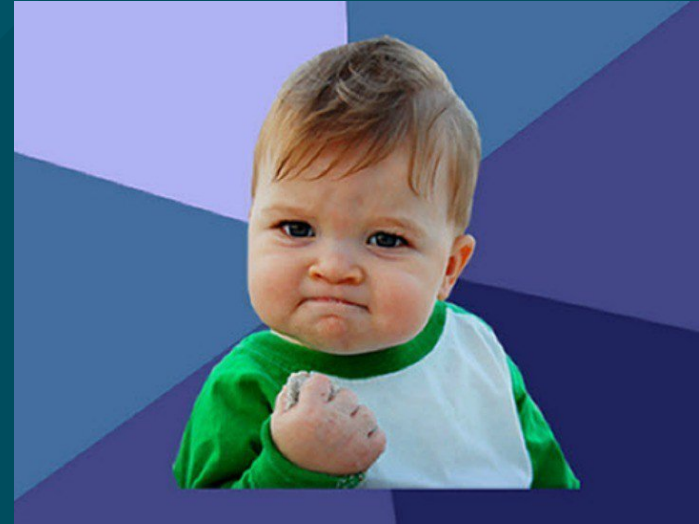
MASSIVE

```
ci-ops-central/bootstrap/provision_jslave.sh \  
--site=ci-osp \  
--project_defaults=/path/to/project_defaults \  
--topology=ci-ops-central/project/config/aio_jslave \  
--ssh_keyfile=/path/to/keyfile \  
--jslavename=jslave-projex-slave \  
--jslaveflavor=m1.xlarge \  
--jslaveimage=rhel-7.1-server-x86_64-released \  
--jslave_execs=10 --jslavecreate \  
--resources_file=jslave-projex-slave.json
```

THIS IS

SIMPLE

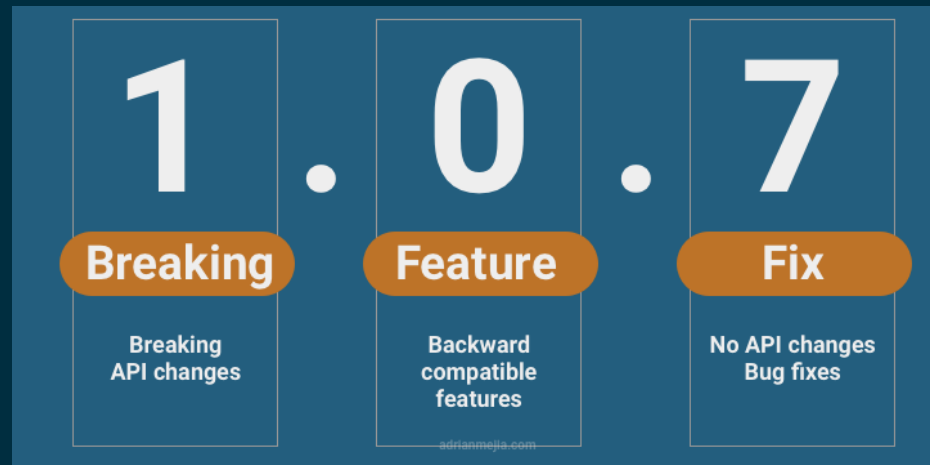
```
cinchpin rise -w jslave
```



- Jenkins Job Builder templates get much simpler!
- CLI can be run from any directory!
- Configuration data is stored in YAML topology files instead of JSON
- CLI is simpler because all topology data is now stored in a topology file

WE HAVE SUPPORTED VERSIONS!

- linch-pin is currently at version
- cinch is currently at version 0.3.0



semver.org

CUSTOMIZATION

A linch-pin topology file makes customization simple

```
---
topology_name: "simple_ae2e_test"
site: "ci-osp"
resource_groups:
- resource_group_name: "ae2e"
  res_group_type: "openstack"
  res_defs:
    - res_name: "test"
      res_type: "os_server"
      flavor: "m1.small"
      image: "rhel-6.5_jeos"
      count: 3
      keypair: "ci-factory"
      networks:
        - "atomic-e2e-jenkins-test"
      fip_pool: "10.8.172.0/22"
  assoc_creds: "ae2e-test_creds"
```

CUSTOMIZATION (2)

Adding an inventory layout file compliments the topology by generating an ansible inventory with desired values

```
---
inventory_layout:
  vars:
    openshift_hostname: __IP__
    openshift_public_hostname: __IP__
  hosts:
    openshift-master:
      host_groups:
        - masters
        - nodes
        - OSEv3
    openshift-node:
      count: 1
      host_groups:
        - nodes
        - OSEv3
    openshift-repo-host:
      host_groups:
        - nodes
        - OSEv3
        - repo_host
  host_groups:
    OSEv3:
      vars:
        openshift_docker_additional_registries: |
```

CUSTOMIZATION (3)

Cinch extends this concept

```
---  
.. layout on previous slide ..  
  
certificate_authority:  
  vars:  
    certificate_authority_urls:  
      - "https://password.corp.redhat.com/legacy.crt"  
      - "https://password.corp.redhat.com/RH-IT-Root-CA.crt"  
      - "https://engineering.redhat.com/Eng-CA.crt"  
repositories:  
  vars:  
    rhel_base: "http://pulp.dist.prod.ext.phx2.redhat.com/content/dist/rhel/server/7/7Server"
```


CURRENT & FUTURE

CURRENT

- PIP & RPM Packages available (RPM before Feb 2017)
- Jenkins in Beaker & Openstack
- Asynchronous Provisioning
- Simple CLI
- Linch-Pin Python API



FUTURE

- Foreman??
- Satellite
- AWS Security Groups
- OpenShift Provisioning
- Vagrant Plugin
- cloud-init
- Provisioning / Teardown Hooks
- Node scaling
- Jenkins Job Builder / Jenkins Pipeline Workflow
- Cloud Bursting

QUESTIONS? COMMENTS?



LINCH-PIN



github.com/CentOS-PaaS-SIG/linch-pin.git



linch-pin.rtf.d.io



contra



continuous-infra@redhat.com

CINCH



github.com/RedHatQE/cinch.git



redhatqe-cinch.rtf.d.io



ci-ops-central



[#ci-ops-central](https://ci-ops-central.slack.com)

Slides available at redhat.slides.com/csavage/ci-provisioning-automation-ansible