RED HAT SUMMIT

Automation: Making the Best Choice for Your Organization

Subheading goes here

Steve Clatterbuck
Infrastructure Architect, Crossvale Inc
4/7/2018

Lee Rich
Sr. Specialist Solution Architect, Red Hat
4/7/2018

Steve Clatterbuck

Infrastructure Architect, Crossvale Inc

Bio

- Recovering Pupp-aholic
- Over 12 years in the SysAdmin Space
- Not a Developer :)
- Specialize in “Infrastructure as Code”
- 5 Years experience with Puppet Enterprise

Lee Rich

Sr. Solutions Architect Specialist, Red Hat

Bio

- Certified pre-owned Puppeteer (Former Puppet SA)
- Former Red Hat SA - RHCE
- Former SysAdmin
- Former Cisco Network Engineer
- Over 25 years in the IT space

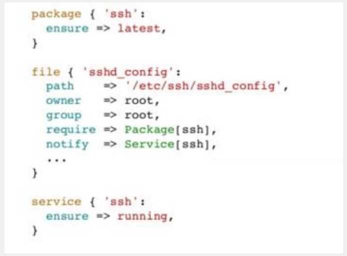
What Automation Tool is Right for your organization?

- Do I want a tool that’s complicated to do simple and complex tasks?
- Do I want a tool that’s simple, to do simple and complex tasks?

Don't Freak out!!!!

What is Puppet?

- Agent based automation tool (Pull)
- Uses Master/Slave configuration model
- Written in a C++ and Clojure (used to be Ruby)
- Uses a Ruby-like DSL language for writing manifests
- Uses facter to gather facts about systems



```
package { 'ssh':
  ensure => latest,
}

file { 'sshd_config':
  path     => '/etc/ssh/sshd_config',
  owner    => root,
  group    => root,
  require  => Package[ssh],
  notify   => Service[ssh],
  ...
}

service { 'ssh':
  ensure => running,
}
```

Likes and Dislikes of Puppet

Likes

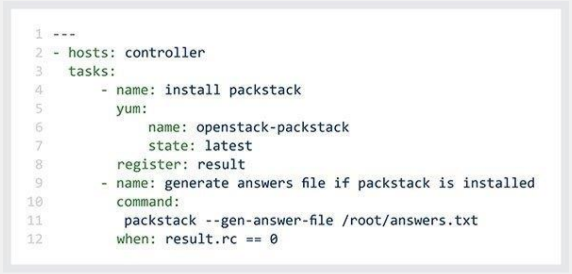
- Letting you define the state of your infrastructure
- Reporting dashboard.
- Although not as powerful, beginners can use the console to group nodes, and apply modules fairly easily.
- If you know what you’re doing, Hiera is very powerful, though, it’s complicated.

Dislikes

- Hiera is complicated
- The DSL is really picky, and hard for some people to pick up, which can scare folks away
- Documentation could use some cleanup
- Maintaining the Masters is a pain.
- The configuration that you really need to run is very complicated (at least for me)
- Order of execution can be a pain to deal with

What is Ansible?

- Push based automation tool
- Can really be run from anywhere
- Written in a Python backend
- Uses YAML for Playbooks, which are very easy to ready
- Built in tools to gather facts, and can utilized other tools, such as Facter to gather facts



```
---
- hosts: controller
  tasks:
      - name: install packstack
        yum:
            name: openstack-packstack
            state: latest
        register: result
      - name: generate answers file if packstack is installed
        command:
         packstack --gen-answer-file /root/answers.txt
        when: result.rc == 0
```

Likes and Dislikes of Ansible

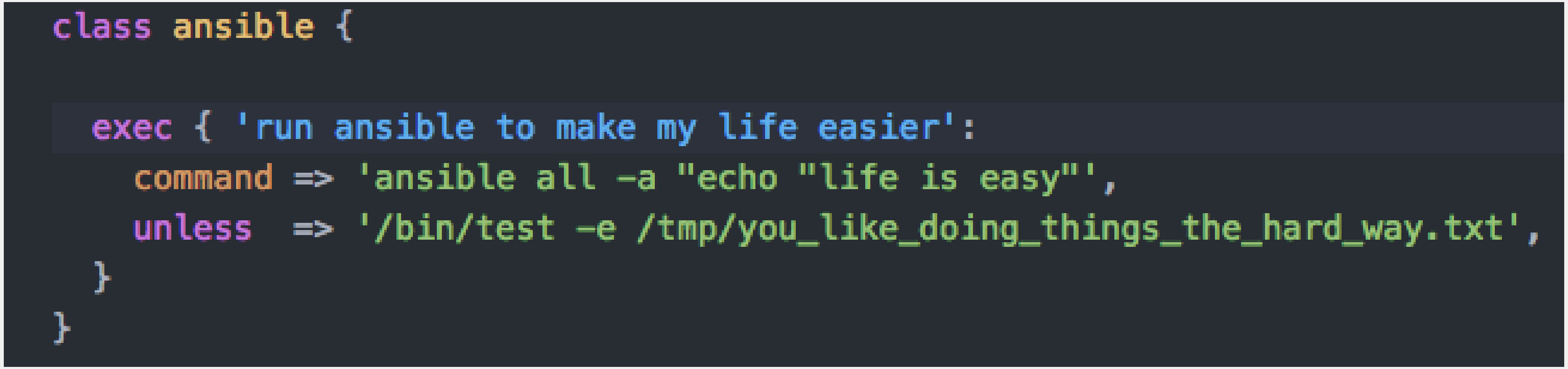
Likes

- Installed via a simple package, which can really be used on any machine.
- Doesn't rely on a master
- Executes tasks in Order
- EASY to get up and running on. It's designed to be easily learned.
- As a sysadmin, you have the control, do the things, when the things need to be done, without relying on check ins

Dislikes

- I personally don't have any

Why did I convert to Ansible?



```
class ansible {

  exec { 'run ansible to make my life easier':
    command => 'ansible all -a "echo "life is easy"',
    unless  => '/bin/test -e /tmp/you_like_doing_things_the_hard_way.txt',
  }
}
```

The Great Debate.. Push vs Pull

Push

- In cases like Ansible, no agents are required
- Typically gives you the ability to do what you want to do, whenever you want to do it, which as a sysadmin, that's exactly what I want
- With Ansible, you can run it from any machine. No need for a master

Pull

- Typically relies on an agent or a script to fetch configurations
- Like any other software, you're responsible for maintaining those agents
- Relies on a master to pull configurations from
- Agent check in is often ten's of minutes apart

The Learning Curve

Ansible

- Know SSH or WinRM
- Ability to read yaml, and create simple plays
- Create an inventory file, ensure connectivity
- Execute Play

Modules

Ansible Core Modules shipped with install

Ansible Modules

- 1500+ modules ship in the module library with Ansible (no need to download externally)
- Provides default Modules to interface with many technologies
- Ansible Galaxy is also accessible to download community modules

Network Devices

Managing Network Devices with Ansible

Ansible

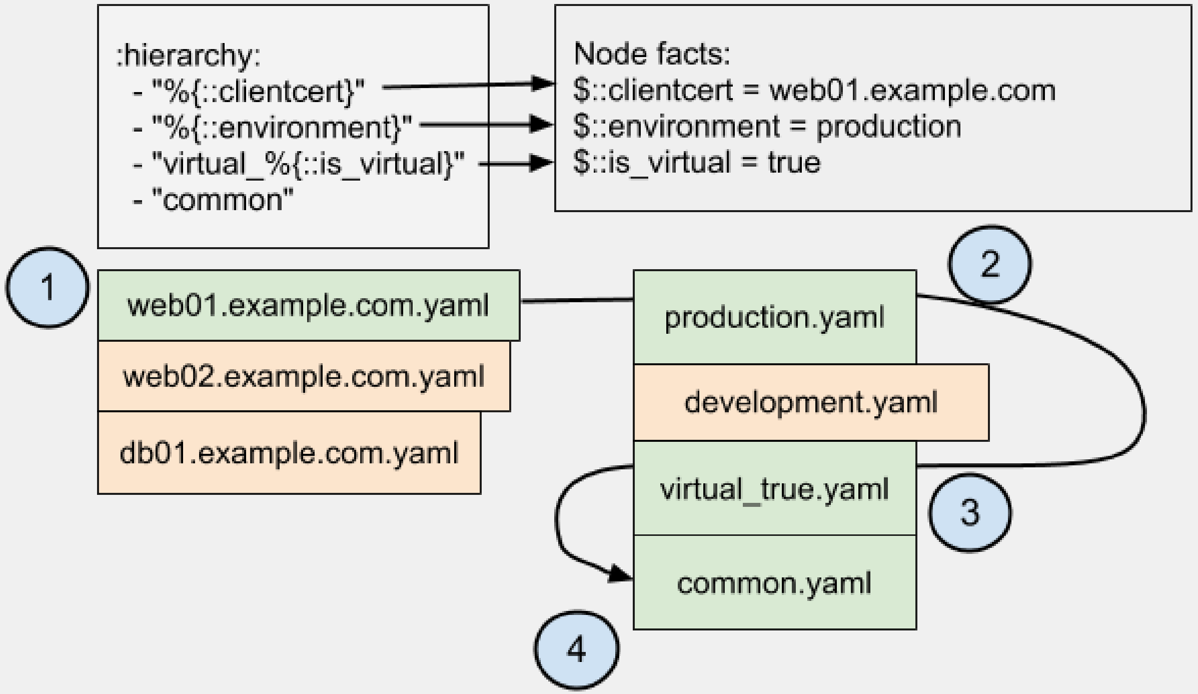
- Agentless, no proxy server required
- 33+ Network Vendors provided out of the box
- Manage Network Devices the same way you manage servers

Scenero: NTP in Puppet vs Ansibe

Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions

1. Puppet Steps
    a. Set up and configure an Open Source Puppet Master, and Puppet DB (easier said than done)
    b. Put in request for firewall team to open the required ports (8140 will give you basic communication)
    c. Enable required agent repos are available on the Puppet Master (RHEL, Ubuntu, SUSE, AIX, Solaris, etc)
    d. Ensure agents are on all nodes in the Raleigh Datacenter
    e. Sign Certificates on Master, or configure autosigning
    f. Only real option for node classification is using Hiera or an External ENC
    g. Come up with Plan to distribute Custom Facts to identify which servers are in Raleigh DC (external facts, pluginsync, etc)
    h. Create a facts distribution module
    i. Download NTP Module from the forge, and make sure it's in your hiera.yaml file

Example Hiera Hierarchy



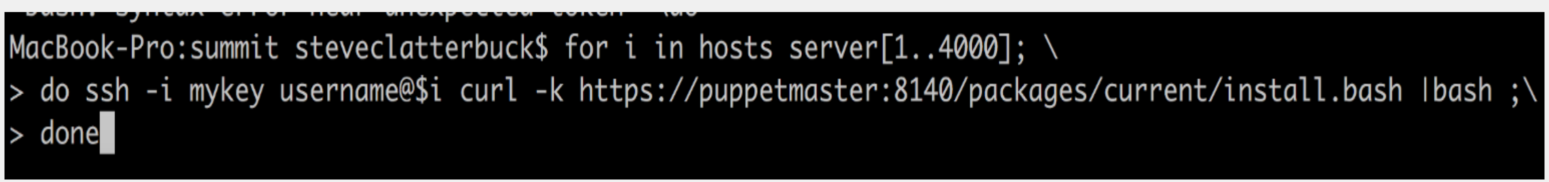
```
:hierarchy:
  - "%{::clientcert}"
  - "%{::environment}"
  - "virtual_%{::is_virtual}"
  - "common"
```

```
Node facts:
$::clientcert = web01.example.com
$::environment = production
$::is_virtual = true
```

Scenero: NTP in Puppet vs Ansibe

Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions

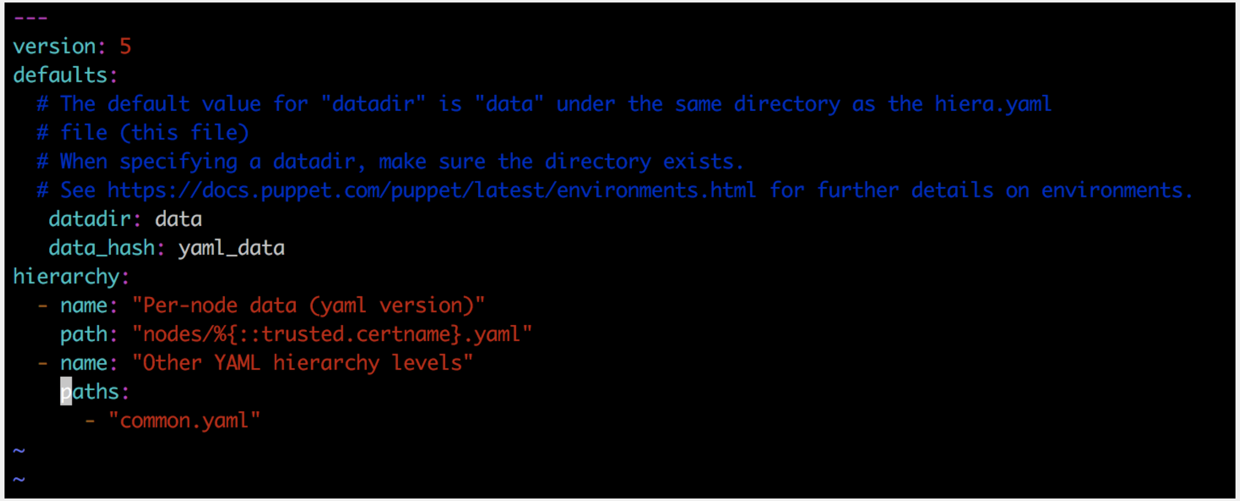
I really hope this works



```
MacBook-Pro:summit steveclatterbuck$ for i in hosts server[1..4000]; \
> do ssh -i mykey username@$i curl -k https://puppetmaster:8140/packages/current/install.bash |bash ;\
> done
```

Scenero: NTP in Puppet vs Ansibe

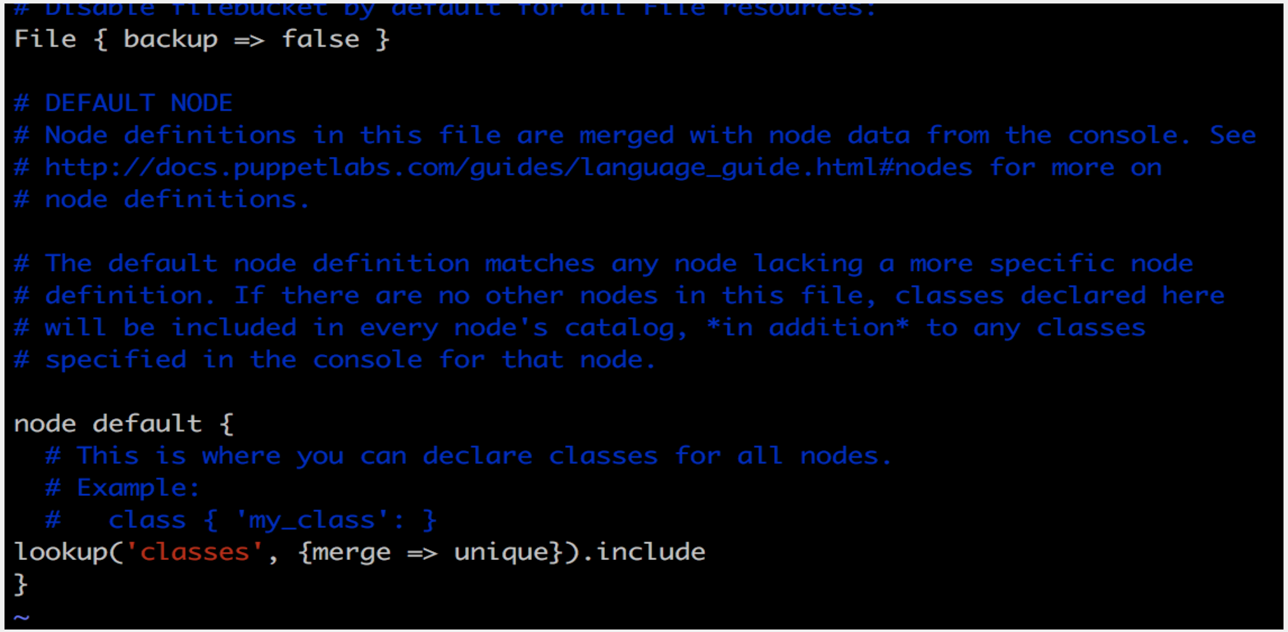
Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions



```
---
version: 5
defaults:
  # The default value for "datadir" is "data" under the same directory as the hiera.yaml
  # file (this file)
  # When specifying a datadir, make sure the directory exists.
  # See https://docs.puppet.com/puppet/latest/environments.html for further details on environments.
   datadir: data
   data_hash: yaml_data
hierarchy:
  - name: "Per-node data (yaml version)"
    path: "nodes/%{::trusted.certname}.yaml"
  - name: "Other YAML hierarchy levels"
    paths:
      - "common.yaml"
~
~
```

Scenero: NTP in Puppet vs Ansibe

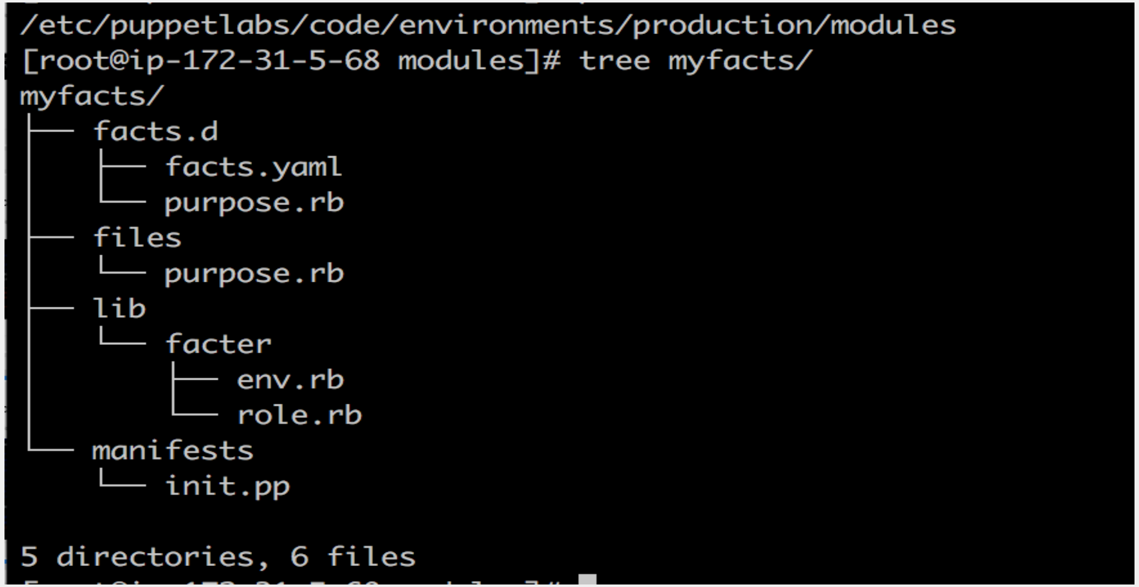
Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions



```
# Disable filebucket by default for all File resources:
File { backup => false }

# DEFAULT NODE
# Node definitions in this file are merged with node data from the console. See
# http://docs.puppetlabs.com/guides/language_guide.html#nodes for more on
# node definitions.

# The default node definition matches any node lacking a more specific node
# definition. If there are no other nodes in this file, classes declared here
# will be included in every node's catalog, *in addition* to any classes
# specified in the console for that node.

node default {
  # This is where you can declare classes for all nodes.
  # Example:
  #   class { 'my_class': }
lookup('classes', {merge => unique}).include
}
~
```

Scenero: NTP in Puppet vs Ansibe

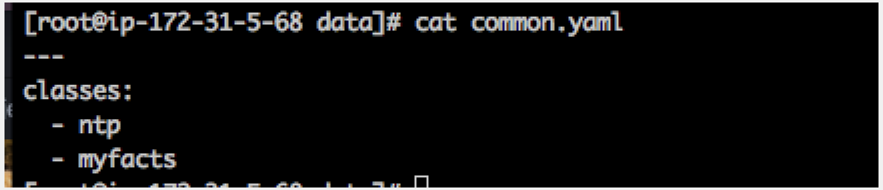
Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions



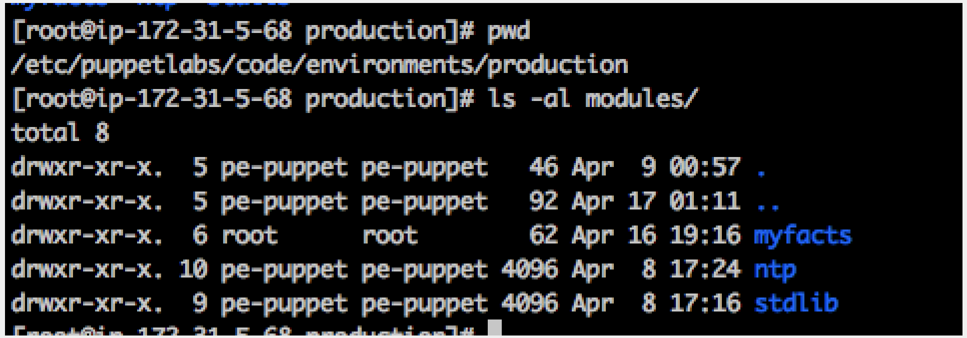
```
/etc/puppetlabs/code/environments/production/modules
[root@ip-172-31-5-68 modules]# tree myfacts/
myfacts/
├── facts.d
│   ├── facts.yaml
│   └── purpose.rb
├── files
│   └── purpose.rb
├── lib
│   └── facter
│       ├── env.rb
│       └── role.rb
└── manifests
    └── init.pp

5 directories, 6 files
```

Scenero: NTP in Puppet vs Ansibe

Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions



```
[root@ip-172-31-5-68 data]# cat common.yaml
---
classes:
  - ntp
  - myfacts
```



```
[root@ip-172-31-5-68 production]# pwd
/etc/puppetlabs/code/environments/production
[root@ip-172-31-5-68 production]# ls -al modules/
total 8
drwxr-xr-x.  5 pe-puppet pe-puppet   46 Apr  9 00:57 .
drwxr-xr-x.  5 pe-puppet pe-puppet   92 Apr 17 01:11 ..
drwxr-xr-x.  6 root      root        62 Apr 16 19:16 myfacts
drwxr-xr-x. 10 pe-puppet pe-puppet 4096 Apr  8 17:24 ntp
drwxr-xr-x.  9 pe-puppet pe-puppet 4096 Apr  8 17:16 stdlib
```

Scenero: NTP in Puppet vs Ansibe

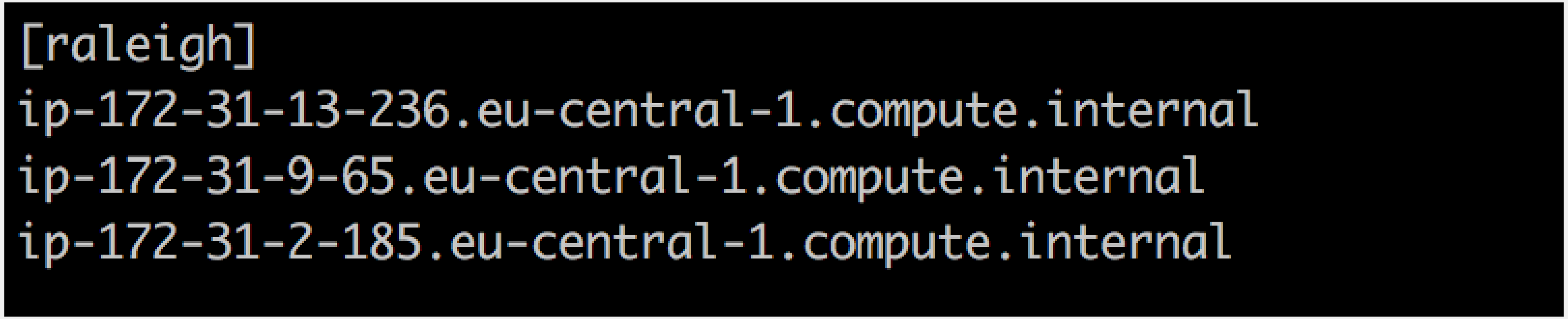
Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions

1. Ansible Steps
    a. Install ansible package via yum
    b. Ensure SSH is open to Raleigh DC (Most environments have this in place)
    c. Create a host file, add Raleigh servers
    d. Download NTP Role from Galaxy
    e. Write 3 lines of yaml
    f. Execute Play

Scenero: NTP in Puppet vs Ansibe

Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions

Ansible

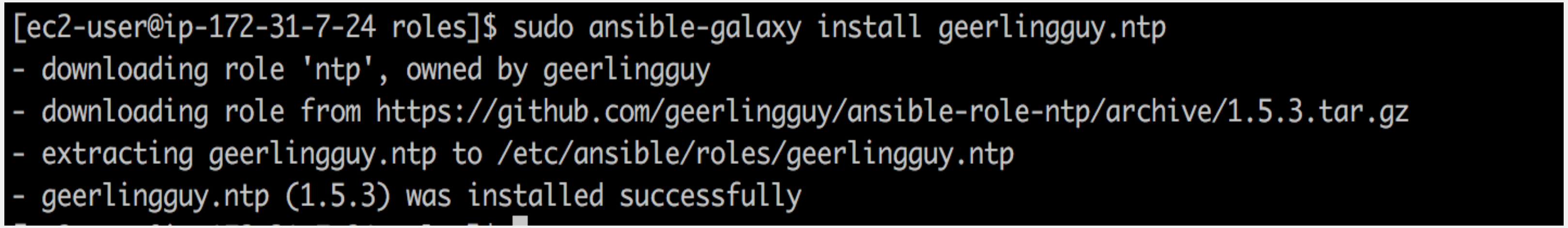


```
[raleigh]
ip-172-31-13-236.eu-central-1.compute.internal
ip-172-31-9-65.eu-central-1.compute.internal
ip-172-31-2-185.eu-central-1.compute.internal
```

Scenero: NTP in Puppet vs Ansibe

Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions

Ansible

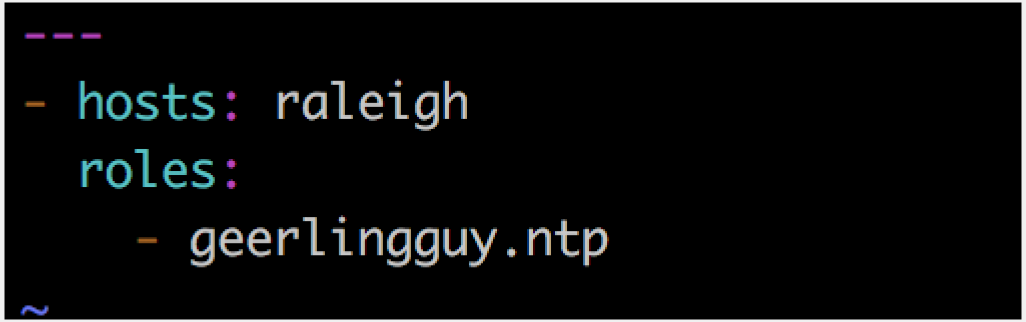


```
[ec2-user@ip-172-31-7-24 roles]$ sudo ansible-galaxy install geerlingguy.ntp
- downloading role 'ntp', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-ntp/archive/1.5.3.tar.gz
- extracting geerlingguy.ntp to /etc/ansible/roles/geerlingguy.ntp
- geerlingguy.ntp (1.5.3) was installed successfully
```

Scenero: NTP in Puppet vs Ansibe

Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions

Ansible



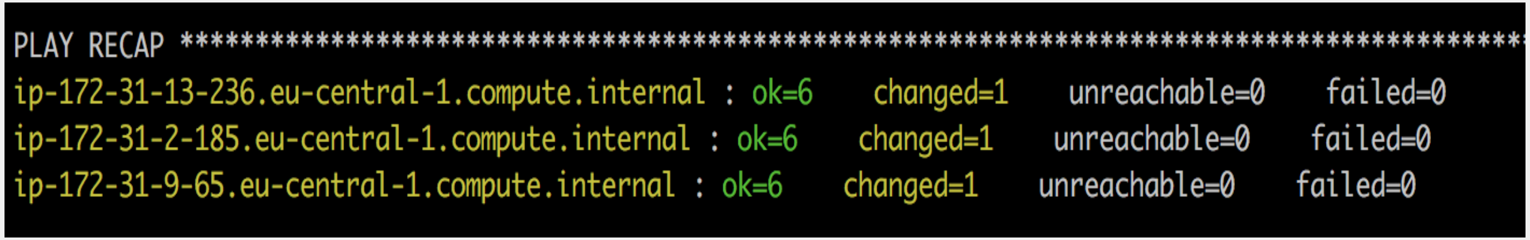
```
---
- hosts: raleigh
  roles:
    - geerlingguy.ntp
~
```

Scenero: NTP in Puppet vs Ansibe

Task: Deploy NTP configs to Raleigh Datacenter in Production Environment using Open Source Solutions

Ansible

```
$ ansible-playbook ntp.yaml
```



```
PLAY RECAP ************************************************************************
ip-172-31-13-236.eu-central-1.compute.internal : ok=6    changed=1    unreachable=0    failed=0
ip-172-31-2-185.eu-central-1.compute.internal : ok=6    changed=1    unreachable=0    failed=0
ip-172-31-9-65.eu-central-1.compute.internal : ok=6    changed=1    unreachable=0    failed=0
```

We’re done!!!!!!!

Demo Time!

Goals

1. Infrastructure Provisioning in AWS using Ansible
2. Deploy a Webserver, a Database server based on AWS tags used in the task above
3. Apply users to new servers using Ansible Tower with Ansible Dynamic inventory scripts.

RED HAT
SUMMIT

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHat

youtube.com/user/RedHatVideos