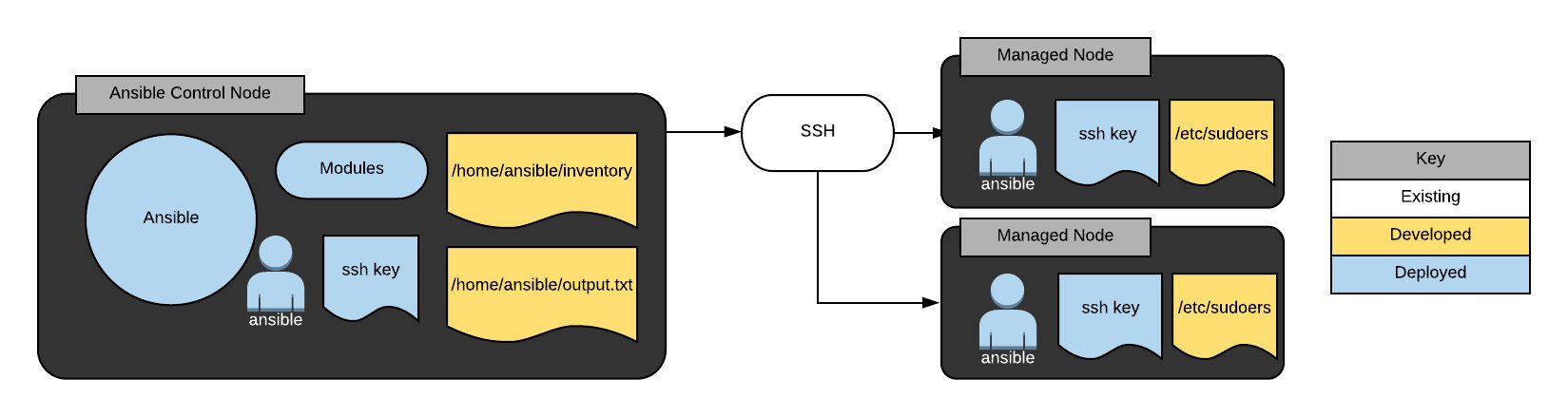
# Ansible Practioner 1-6

1 Getting Started with Ansible



# Getting Started with Ansible

# In this hands-on lab, we'll install Ansible on a control node and configure two managed servers for use with Ansible. We will also create a simple inventory and run an Ansible command to verify our configuration is correct.

*This course is not approved or sponsored by Red Hat.*

## Install Ansible on the Control Node

## Log in to the control node using ssh, cloud\_user, and the provided public IP address and password:

ssh cloud\_user@<PUBLIC IP>

To install Ansible on the control node:

sudo yum install ansible

## Configure the ansible User on the Control Node

## Next, we'll configure the ansible user on the control node for ssh shared key access to managed nodes.

**Note:** Do not use a passphrase for the key pair.

Create a key pair for the ansible user on the control host, accepting the defaults when prompted:

su - ansible  
ssh-keygen

Copy the public key to both node1 and node2:

ssh-copy-id node1  
ssh-copy-id node2

## Create a Simple Ansible Inventory

## Next, we'll create a simple Ansible inventory on the control node in /home/ansible/inventory containing node1 and node2.

On the control host:

su - ansible  
touch /home/ansible/inventory  
echo "node1" >> /home/ansible/inventory  
echo "node2" >> /home/ansible/inventory

## Configure sudo Access for Ansible

## Now, we'll configure sudo access for Ansible on node1 and node2 such that Ansible may use sudo for any command with no password prompt.

Log in to node1 as cloud\_user and edit the sudoers file to contain appropriate access for the ansible user:

ssh cloud\_user@node1  
sudo visudo

Add the following line to the file and save:

ansible    ALL=(ALL)       NOPASSWD: ALL

Enter:

logout

Repeat these steps for node2, and then back out to the control node.

## Verify Each Managed Node Is Accessible

## Finally, we'll verify each managed node is able to be accessed by Ansible from the control node using the ping module.

Redirect the output of a successful command to /home/ansible/output.

To verify each node, run the following as the ansible user from the control host:

ansible -i /home/ansible/inventory node1 -m ping  
ansible -i /home/ansible/inventory node2 -m ping

To redirect output of a successful command to /home/ansible/output:

ansible -i /home/ansible/inventory node1 -m ping > /home/ansible/output

## Conclusion

## Congratulations on completing this lab!

## Additional Resources

Your CIO has greenlit a proof of concept for Ansible in your environment. You are to set up an Ansible control node in a test environment and verify basic functionality. You have three demo hosts, one to be the control node (control1), and two to serve as managed nodes (node1 and node2). You must complete the following steps:

1. Install Ansible on the control node.
2. Configure the ansible user on the control node for ssh shared key access to managed nodes.  
   **Note**: do not use a passphrase for the key pair.
3. Create a simple Ansible inventory on the control node in /home/ansible/inventory containing node1 and node2.
4. Configure sudo access for Ansible on node1 and node2 so that Ansible may use sudo for any command with no password prompt.
5. Verify each managed node can be accessed by Ansible from the control node using the ping module. Redirect the output of a successful command to /home/ansible/output.

Important Notes:

* The user ansible is already present on all servers for your convenience.
* The ansible user has the same password as the cloud\_user.
* /etc/hosts entries are present on control1 for the managed nodes.

## Learning Objectives

0 of 5 completed

Install Ansible on the control node.

To install Ansible on the control node, run sudo yum install ansible.

Configure the `ansible` user on the control node for ssh shared key access to managed nodes. Do not use a passphrase for the key pair.

1. To create a keypair for the ansible user on the control host, run the following:

* sudo su - ansible
* ssh-keygen (accept all defaults: press **enter** for each prompt)
* Copy the public key to both node1 and node2.
* As the ansible user on the control host:
  + ssh-copy-id node1 (accept the host key if prompted, authenticate as ansible user)
  + ssh-copy-id node2 (accept the host key if prompted, authenticate as ansible user)

Create a simple Ansible inventory on the control node in `/home/ansible/inventory` containing `node1` and `node2`.

* On the control host:

1. sudo su - ansible (if not already ansible user)

* touch /home/ansible/inventory
* echo "node1" >> /home/ansible/inventory
* echo "node2" >> /home/ansible/inventory

Configure sudo access for Ansible on `node1` and `node2` such that Ansible may use sudo for any command with no password prompt.

* Log in to node1 as cloud\_user and edit the sudoers file to contain appropriate access for the ansible user:
  + ssh cloud\_user@node1
  + sudo visudo
  + Add the following line to the file and save:

ansible    ALL=(ALL)       NOPASSWD: ALL

* Repeate these steps for node2.

Verify each managed node is able to be accessed by Ansible from the control node using the `ping` module. Redirect the output of a successful command to `/home/ansible/output`.

* To verify each node, run the following as the ansible user from the control host:
  + ansible -i /home/ansible/inventory node1 -m ping
  + ansible -i /home/ansible/inventory node2 -m ping
* To redirect output of a successful command to /home/ansible/output:
  + ansible -i /home/ansible/inventory node1 -m ping > /home/ansible/output

.

CONTROL NODE :

[cloud\_user@control1 ~]$ curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo '37n(&jyK' | /bin/passwd cloud\_user —stdin

/usr/bin/hostnamectl set-hostname control1

yum-config-manager —enable rhui-REGION-rhel-server-extras

curl <https://raw.githubusercontent.com/ACloudGuru-Resources/Getting-Started-with-Ansible/master/my-custom.lang.sh> -o/etc/profile.d/my-custom.lang.sh

source /etc/profile.d/my-custom.lang.sh

/bin/echo 10.0.1.231 node1 >> /etc/hosts

/bin/echo 10.0.1.148 node2 >> /etc/hosts

/sbin/useradd ansible

/bin/echo '37n(&jyK' | /bin/passwd ansible —stdin

cd /tmp

yum install <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm> -y

/usr/bin/pip3 install <https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz>

/usr/local/bin/cfn-signal -e 0 —stack cfst-1571-12b50ce924598a9b2eb82d72abb3452d —resource Control1 —region us-east-1

[cloud\_user@control1 ~]$ ansible —version

-bash: ansible: command not found

[cloud\_user@control1 ~]$ sudo cat /etc/os-release

[sudo] password for cloud\_user:

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[cloud\_user@control1 ~]$

NODE 1 :

[cloud\_user@node1 ~]$ curl 169.254.169.254/latest/user-data

#!/bin/bash

/usr/bin/hostnamectl set-hostname node1

/bin/echo '37n(&jyK' | /bin/passwd cloud\_user —stdin

/usr/bin/yum-config-manager —enable rhui-REGION-rhel-server-extras —save

/sbin/useradd ansible

/bin/echo '37n(&jyK' | /bin/passwd ansible —stdin

curl <https://raw.githubusercontent.com/ACloudGuru-Resources/Getting-Started-with-Ansible/master/my-custom.lang.sh> -o/etc/profile.d/my-custom.lang.sh

source /etc/profile.d/my-custom.lang.sh

yum install <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm> -y

/usr/bin/pip3 install <https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz>

/usr/local/bin/cfn-signal -e 0 —stack cfst-1571-12b50ce924598a9b2eb82d72abb3452d —resource Node1 —region us-east-1

[cloud\_user@node1 ~]$ ansible —version

-bash: ansible: command not found

[cloud\_user@node1 ~]$ sudo cat /etc/os-release

[sudo] password for cloud\_user:

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

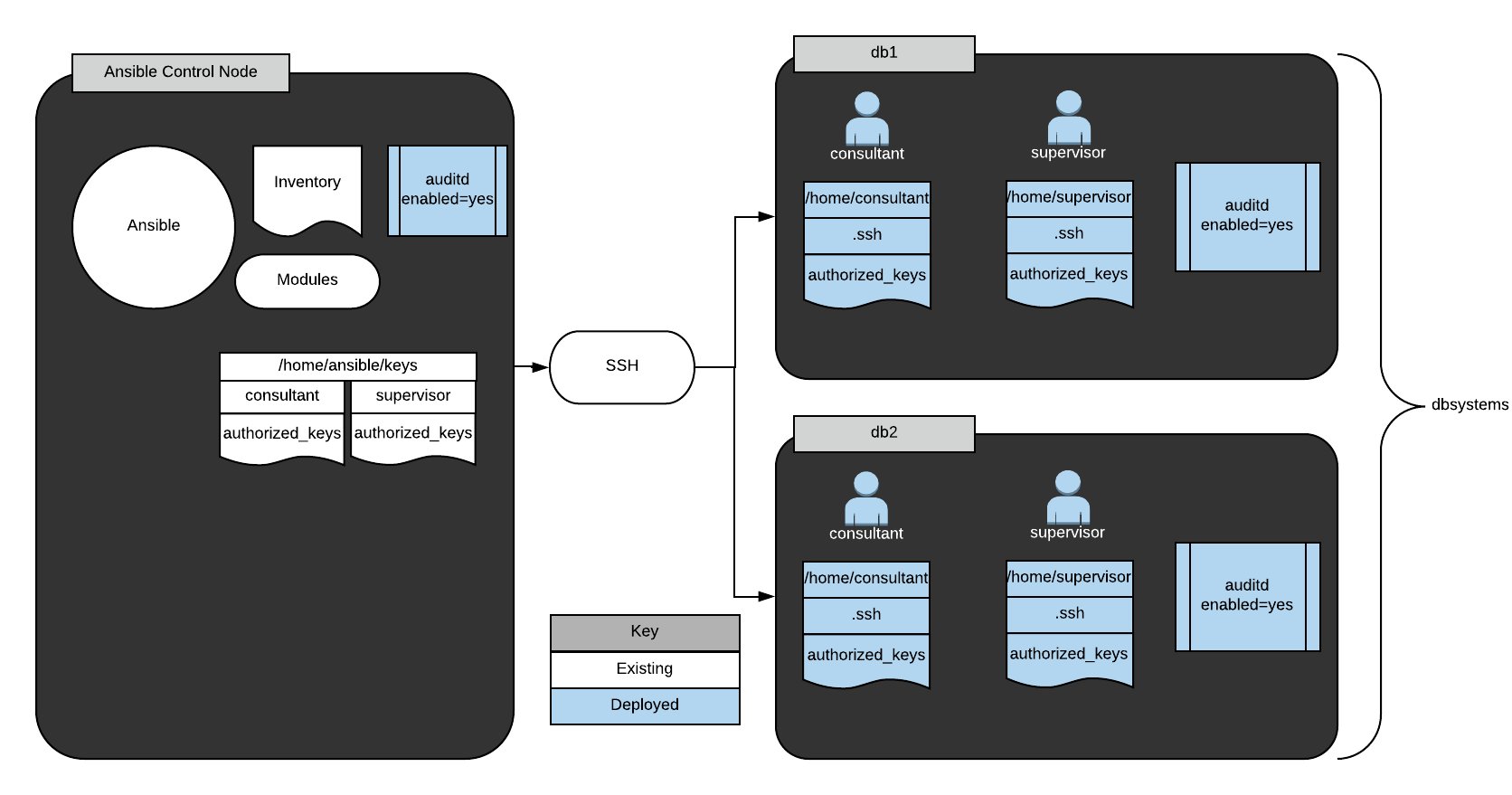
REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[cloud\_user@node1 ~]$

2. :

## Ad-Hoc Ansible Commands



# Ad-Hoc Ansible Commands

# *This course is not approved or sponsored by Red Hat.*

## The Scenario

## Some consultants will be performing audits on a number of systems in our company's environment. We've got to create the user accounts listed in */home/ansible/userlist.txt* and set up the provided public keys for their accounts. The security team has built a jump host for the consultants to access production systems and provided us with the full key-pair so we can set up and test the connection. All hosts in dbsystems will need that public key installed so the consultants may use key-pair authentication to access the systems. We must also ensure the auditd service is enabled and running on all systems.

Important notes:

* Ansible is already installed on the control node. If we connect to the server by clicking on the Public IP address in a web browser, we need to make sure we change to the ansible user, with the sudo su - ansible command.
* The user ansible is present on all servers with appropriate shared keys for access to managed servers from the control node. We need to make sure we use this user to complete the commands.
* The ansible user has the same password as cloud\_user.
* The default Ansible inventory has already been configured with the appropriate hosts and groups.
* /etc/hosts entries are present on the control1 host for the managed servers.

### Get Logged In

### Login credentials are all on the lab overview page. Once we're logged into the control1 server, become the ansible user (su - ansible) and we can get going.

### Create the User Accounts Noted in /home/ansible/userlist.txt

### If we read the userlist.txt file in our home directory, we'll see consultant and supervisor. Those are the two new user accounts we have to create:

[ansible@control1]$ ansible dbsystems -b -m user -a "name=consultant"  
[ansible@control1]$ ansible dbsystems -b -m user -a "name=supervisor"

### Place Key Files in the Correct Location, /home/$USER/.ssh/authorized\_keys, on Hosts in dbsystems

[ansible@control1]$ ansible dbsystems -b -m file -a "path=/home/consultant/.ssh state=directory owner=consultant group=consultant mode=0755"  
[ansible@control1]$ ansible dbsystems -b -m copy -a "src=/home/ansible/keys/consultant/authorized\_keys dest=/home/consultant/.ssh/authorized\_keys mode=0600 owner=consultant group=consultant"  
[ansible@control1]$ ansible dbsystems -b -m file -a "path=/home/supervisor/.ssh state=directory owner=supervisor group=supervisor mode=0755"  
[ansible@control1]$ ansible dbsystems -b -m copy -a "src=/home/ansible/keys/supervisor/authorized\_keys dest=/home/supervisor/.ssh/authorized\_keys mode=0600 owner=supervisor group=supervisor"

### Ensure auditd Is Enabled and Running on All Hosts

[ansible@control1]$ ansible all -b -m service -a "name=auditd state=started enabled=yes"

### Conclusion

### We can see, by watching output from those commands, that they all ran successfully. Congratulations!

## Additional Resources

Some consultants have been employed to perform audits on a number of systems in your company's environment. You must create the user accounts noted in */home/ansible/userlist.txt* and set up the provided public keys for their accounts. The security team has built a jump host for the consultants to access production systems and provided the full key-pair to you so you may set up and test the connection. All hosts in dbsystems will need the provided public key installed so the consultants may use key-pair authentication to access the systems. Also, you must ensure the auditd service is enabled and running on all systems.

To summarize, you must do the following:

1. Create the user accounts noted in /home/ansible/userlist.txt.

* Copy the authorized\_keys file for each user to the correct location so the new accounts can log in with ssh key authentication.
* Ensure auditd is enabled and running on all systems.

Important notes:

* For your convenience, Ansible is already on the control node. If you connect to the server by clicking on the Public IP address in your browser, make sure to change to the ansible user with the sudo su - ansible command.
* The user ansible is present on all servers with appropriate shared keys for access to managed servers from the control node. Make sure to use this user to complete the commands.
* The ansible user has the same password as cloud\_user.
* The default Ansible inventory has been configured for you with the appropriate hosts and groups.
* /etc/hosts entries are present on control1 for the managed servers.

## Learning Objectives

0 of 3 completed

Create the User Accounts Noted in `/home/ansible/userlist.txt`

* ansible dbsystems -b -m user -a "name=consultant"
* ansible dbsystems -b -m user -a "name=supervisor"

Place Key Files in the Correct Location, `/home/$USER/.ssh/authorized\_keys`, on Hosts in `dbsystems`

* ansible dbsystems -b -m file -a "path=/home/consultant/.ssh state=directory owner=consultant group=consultant mode=0755"
* ansible dbsystems -b -m copy -a "src=/home/ansible/keys/consultant/authorized\_keys dest=/home/consultant/.ssh/authorized\_keys mode=0600 owner=consultant group=consultant"
* ansible dbsystems -b -m file -a "path=/home/supervisor/.ssh state=directory owner=supervisor group=supervisor mode=0755"
* ansible dbsystems -b -m copy -a "src=/home/ansible/keys/supervisor/authorized\_keys dest=/home/supervisor/.ssh/authorized\_keys mode=0600 owner=supervisor group=supervisor"

Ensure `auditd` Is Enabled and Running on All Hosts

ansible all -b -m service -a "name=auditd state=started enabled=yes"

.

CONTROL NODE :  
  
  
[cloud\_user@control1 ~]$ sudo su

[sudo] password for cloud\_user:

[root@control1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo '7D+4f|QR' | /bin/passwd root —stdin

/bin/echo '7D+4f|QR' | /bin/passwd cloud\_user —stdin

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

sed -i s/requiretty/\!requiretty/ /etc/sudoers

hostnamectl set-hostname control1

/bin/echo 10.0.1.168 db1 node1>> /etc/hosts

/bin/echo 10.0.1.121 db2 node2>> /etc/hosts

yum-config-manager —enable rhui-REGION-rhel-server-extras

cd /tmp

wget <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

yum -y —enablerepo=epel install openssl-devel python-devel

yum -y install <http://mirror.centos.org/centos/7/extras/x86_64/Packages/sshpass-1.06-2.el7.x86_64.rpm>

yum -y install <https://releases.ansible.com/ansible/rpm/release/epel-7-x86_64/ansible-2.6.2-1.el7.ans.noarch.rpm>

/usr/bin/pip3 install <https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz>

/sbin/useradd ansible

/bin/echo '7D+4f|QR' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

sudo -u ansible /bin/mkdir -p /home/ansible/.ssh

sudo -u ansible /bin/ssh-keygen -q -N "" -f /home/ansible/.ssh/id\_rsa

sudo -u ansible sshpass -p '7D+4f|QR' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@localhost

sudo -u ansible sshpass -p '7D+4f|QR' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@db1

sudo -u ansible sshpass -p '7D+4f|QR' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@db2

mkdir -p /home/ansible/keys/{consultant,supervisor}/.ssh

/bin/ssh-keygen -q -N "" -f /home/ansible/keys/consultant/.ssh/id\_rsa

/bin/ssh-keygen -q -N "" -f /home/ansible/keys/supervisor/.ssh/id\_rsa

cat /home/ansible/keys/consultant/.ssh/id\_rsa.pub > /home/ansible/keys/consultant/authorized\_keys

cat /home/ansible/keys/supervisor/.ssh/id\_rsa.pub > /home/ansible/keys/supervisor/authorized\_keys

echo "localhost" >> /etc/ansible/hosts

echo "[dbsystems]" >> /etc/ansible/hosts

echo "db1" >> /etc/ansible/hosts

echo "db2" >> /etc/ansible/hosts

/bin/yum install -y git

git clone <https://github.com/linuxacademy/content-rh-ansible.git> /root/content-rh-ansible

/bin/mv /root/content-rh-ansible/lab2/control/userlist.txt /home/ansible/userlist.txt

/bin/chmod -R ansible:ansible /home/ansible

/bin/rm -rf /root/content-rh-ansible

/usr/local/bin/cfn-signal -e 0 —stack cfst-612-757da3d65cac2ac0f18d807cb6092bee —resource Control1 —region us-east-1

[root@control1 cloud\_user]# ansible —version

ansible 2.6.2

config file = /etc/ansible/ansible.cfg

configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /bin/ansible

python version = 2.7.5 (default, May 27 2022, 11:27:32) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]

[root@control1 cloud\_user]# sudo cat /etc/os-release

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[root@control1 cloud\_user]#

NODE ! AND NODE 2 :

[cloud\_user@db1 ~]$ sudo su

[sudo] password for cloud\_user:

[root@db1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo '7D+4f|QR' | /bin/passwd cloud\_user —stdin

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

/usr/bin/hostnamectl set-hostname db1

/usr/bin/yum-config-manager —enable rhui-REGION-rhel-server-extras/sbin/useradd ansible

/bin/echo '7D+4f|QR' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

yum install -y mariadb-server

systemctl start mariadbsystemctl enable mariadb

systemctl disable auditd

[root@db1 cloud\_user]# ansible —version

bash: ansible: command not found

[root@db1 cloud\_user]# sudo cat /etc/os-release

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

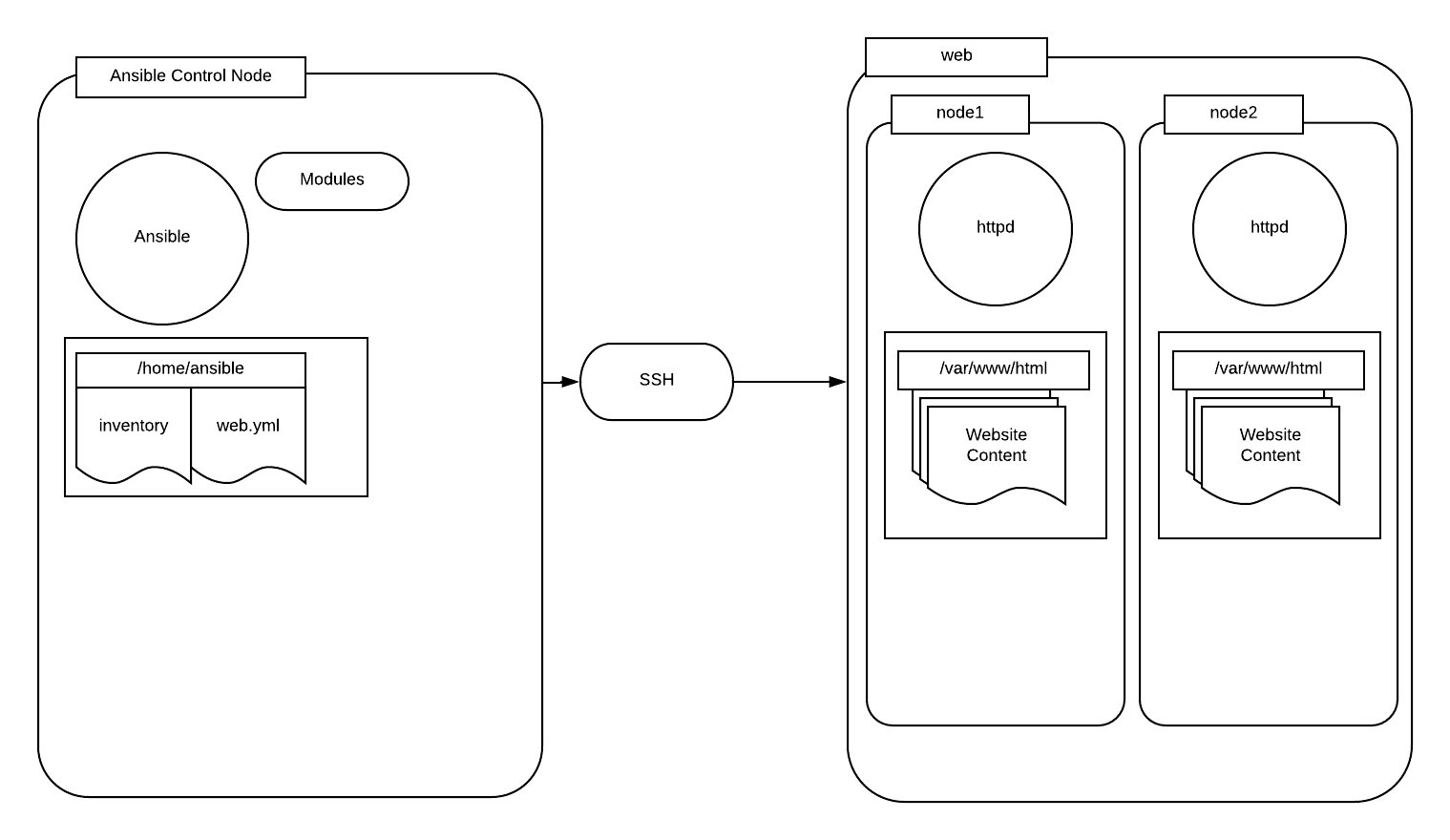
REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[root@db1 cloud\_user]#

3:

## Ansible Playbooks: The Basics.



. Ansible Playbooks - the Basics

# *This course is not approved or sponsored by Red Hat.*

## The Scenario

## Our company has been increasing the deployment of small brochure-style websites for clients. The head of IT has decided that each client should have their own web server, for better client isolation, and has tasked us with creating concept automation to quickly deploy web-nodes with simple static website content.

We have been provided an Ansible control node (control1) and 2 test lab servers (node1 and node2) that have been preconfigured with the ansible user and key.

We must create an Ansible inventory in */home/ansible/inventory* containing a host group named *web*. The web group should contain node1 and node2.

Then we've got to design an Ansible playbook that will execute the following tasks on your configured inventory:

* Install *httpd*
* Start and enable the *httpd* service
* Install a simple website provided on a repository server.

The playbook will be in */home/ansible/web.yml*. The simple website may be accessed from [*http://repo.example.com/website.tgz*](http://repo.example.com/website.tgz).

## Get Logged In

## Login credentials are all on the lab overview page. Once we're logged into the control1 server, become the ansible user (su - ansible) and we can get going.

## Create an inventory in /home/ansible/inventory That Contains a Host Group Named web. The web Group Should Contain node1 and node2

## Use Vim to create the inventory file:

[ansible@control1]$ echo "[web]" >> /home/ansible/inventory  
[ansible@control1]$ echo "node1" >> /home/ansible/inventory  
[ansible@control1]$ echo "node2" >> /home/ansible/inventory

It should contain this when we're done:

[web]  
node1  
node2

## Create a Playbook in /home/ansible/web.yml

## Using Vim, we'll create our web.yml file with these contents:

---  
- hosts: web  
  become: yes  
  tasks:  
    - name: install httpd  
      yum: name=httpd state=latest  
    - name: start and enable httpd  
      service: name=httpd state=started enabled=yes  
    - name: retrieve website from repo  
      get\_url: url=http://repo.example.com/website.tgz dest=/tmp/website.tgz  
    - name: install website  
      unarchive: remote\_src=yes src=/tmp/website.tgz dest=/var/www/html/

## Verify the Work by Executing the Playbook Using the Inventory

[ansible@control1]$ ansible-playbook -i /home/ansible/inventory /home/ansible/web.yml

## Conclusion

## This new setup is everything we needed. The Ansible playbook installs httpd, starts and enables it, and creates a simple website, all on it's own web node. That's what we needed. Congratulations!

## Additional Resources

Your company has been increasing the deployment of small broacher-style websites for clients. The head of IT has decided that each client should have their own web servers for better client isolation and has tasked you with creating concept automation to quickly deploy web-nodes with simple static website content.

You have been provided an ansible control node and 2 test lab servers (node1 and node2) that have been preconfigured with the ansible user and key.

You must create an ansible inventory in */home/ansible/inventory* containing a host group named *web*. The web group should contain node1 and node2.

Furthermore, you must design an Ansible playbook that will execute the following tasks on your configured inventory: install *httpd*, start and enable the *httpd* service, and install a simple website provided on a repo server. Create the playbook in */home/ansible/web.yml*. The simple website may be accessed from [*http://repo.example.com/website.tgz*](http://repo.example.com/website.tgz).

Note: Please wait an extra minute before starting the lab to make sure it is fully provisioned.

Summary tasks list:

1. Create an inventory in */home/ansible/inventory* containing a host group named *web*. The web group should contain node1 and node2.

* Create a playbook in */home/ansible/web.yml*.
* Configure the playbook to install *httpd* on the *web* group.
* Configure the playbook to start and enable the *httpd* service on the *web* group.
* Configure the playbook to retrieve the website from [*http://repo.example.com/website.tgz*](http://repo.example.com/website.tgz) on each server in the web group.
* Configure the playbook to unarchive the website into */var/www/html* on all servers in the *web* group.
* Execute the playbook you created using the inventory you created to verify your work.

Important notes:

* For your convenience, Ansible has been installed on the control node.
* The user *ansible* is present on all servers with appropriate shared keys for access to managed servers from the control node.
* The *ansible* user has the same password as *cloud\_user*.
* */etc/hosts* entries have been made on control1 for the managed servers.

## Learning Objectives

0 of 7 completed

Create an Inventory in `/home/ansible/inventory `That Contains a Host Group Named `web`. The `web` Group Should Contain `node1` and `node2`

* echo "[web]" >> /home/ansible/inventory
* echo "node1" >> /home/ansible/inventory
* echo "node2" >> /home/ansible/inventory

Create a Playbook in `/home/ansible/web.yml`

echo "---" >> /home/ansible/web.yml

Configure the Playbook to Install `httpd` on the `web` Group

Using a text editor, such as vim, edit */home/ansible/web.yml* to contain the following text block below the line containing "---": - hosts: web become: yes tasks: - name: install httpd yum: name=httpd state=latest

Configure the Playbook to Start and Enable the `httpd` Service on the `web` Group

Using a text editor such as vim, edit */home/ansible/web.yml* to contain the following task block after the "install httpd task": - name: start and enable httpd service: name=httpd state=started enabled=yes

Configure the Playbook to Retrieve the Website from \*http://repo.example.com/website.tgz\* on Each Server in the `web` Group

Using a text editor such as vim, edit */home/ansible/web.yml* to contain the following task block after the "start and enable httpd" task: - name: retrieve website from repo get\_url: url=<http://repo.example.com/website.tgz> dest=/tmp/website.tgz

Configure the Playbook to Unarchive the Website into `/var/www/html` on All Servers in the `web` Group

Using a text editor such as vim, edit */home/ansible/web.yml* to contain the following task block after the "retrieve website from repo" task: - name: install website unarchive: remote\_src=yes src=/tmp/website.tgz dest=/var/www/html/

Verify the Work by Executing the Playbook Using the Inventory

ansible-playbook -i /home/ansible/inventory /home/ansible/web.yml

.

CONTROL NODE :

[root@control1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo '8-7nC[ec' | /bin/passwd cloud\_user —stdin

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

sed -i s/requiretty/\!requiretty/ /etc/sudoers

hostnamectl set-hostname control1

/bin/echo 10.0.1.126 node1>> /etc/hosts

/bin/echo 10.0.1.42 node2>> /etc/hosts

/bin/echo "127.0.0.1 repo.example.com" >> /etc/hosts

yum-config-manager —enable rhui-REGION-rhel-server-extras

cd /tmp

wget <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpmyum> install -y epel-release-latest-7.noarch.rpm

yum install -y ansible

yum install -y sshpass

/bin/echo '8-7nC[ec' | /bin/passwd root —stdin

/sbin/useradd ansible

/bin/echo '8-7nC[ec' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

sudo -u ansible /bin/mkdir -p /home/ansible/.ssh

sudo -u ansible /bin/ssh-keygen -q -N "" -f /home/ansible/.ssh/id\_rsa

sudo -u ansible sshpass -p '8-7nC[ec' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@localhost

sudo -u ansible sshpass -p '8-7nC[ec' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@node1

sudo -u ansible sshpass -p '8-7nC[ec' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@node2

echo "[nodes]" >> /etc/ansible/hosts

echo "node1" >> /etc/ansible/hosts

echo "node2" >> /etc/ansible/hosts

head -3 /etc/hosts > /tmp/hosts

echo "$(hostname -i | awk {'print $2'}) repo.example.com" >> /tmp/hostssudo -u ansible ansible nodes -b -m copy -a "src=/tmp/hosts dest=/etc/hosts"

/bin/rm -rf /tmp/hosts

yum install -y httpd

systemctl start httpd

systemctl enable httpd

/bin/yum install -y gitgit clone <https://github.com/linuxacademy/content-rh-ansible.git> /root/content-rh-ansible

/bin/mv /root/content-rh-ansible/lab4/control/website.tgz /var/www/html/website.tgz

restorecon /var/www/html/website.tgz

/bin/chmod -R ansible:ansible /home/ansible

/bin/rm -rf /root/content-rh-ansible

/usr/bin/pip3 install <https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz>

/usr/local/bin/cfn-signal -e 0 —stack cfst-611-5e5c5228a0971957296e331d84a7781c —resource Control1 —region us-east-1

[root@control1 cloud\_user]# ansible —version

ansible 2.9.27

config file = /etc/ansible/ansible.cfg

configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /bin/ansible

python version = 2.7.5 (default, May 27 2022, 11:27:32) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]

[root@control1 cloud\_user]# sudo cat /etc/os-release

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[root@control1 cloud\_user]#

NODE1 :

[root@node1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo '8-7nC[ec' | /bin/passwd cloud\_user —stdin

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

/usr/bin/hostnamectl set-hostname node1

/usr/bin/yum-config-manager —enable rhui-REGION-rhel-server-extras

cd /tmp

wget <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

yum install -y epel-release-latest-7.noarch.rpm

/sbin/useradd ansible

/bin/echo '8-7nC[ec' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers/usr/bin/pip3 install <https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz>

/usr/local/bin/cfn-signal -e 0 —stack cfst-611-5e5c5228a0971957296e331d84a7781c —resource Node1 —region us-east-1

[root@node1 cloud\_user]# ansible —version

bash: ansible: command not found

[root@node1 cloud\_user]# sudo cat /etc/os-release

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

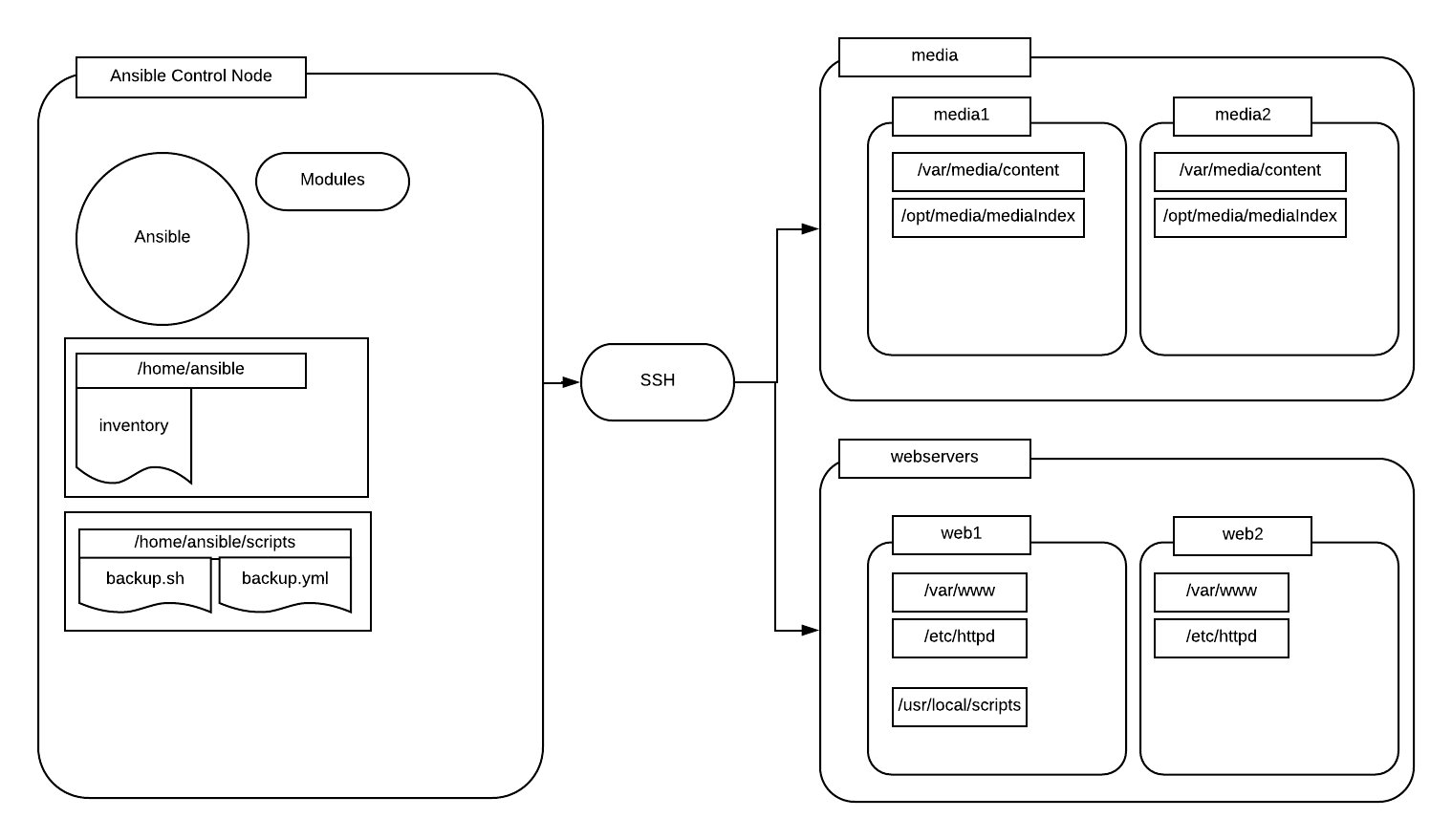
REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[root@node1 cloud\_user]#

4 :

## Working with Ansible Inventories. .



# Working with Ansible Inventories

# *This course is not approved or sponsored by Red Hat.*

## Introduction

## Your company decided that their backup software license was frivolous and unnecessary. Because of this, the license was not renewed. As a stopgap measure, your supervisor has created a simple script and an Ansible playbook to create an archive of select files, depending on pre-defined Ansible host groups. You will create the inventory file to complete the backup strategy.

## Solution

Log in to the server using the provided lab credentials:

ssh ansible@<PUBLIC\_IP\_ADDRESS>

If you are logged in as cloud\_user, switch to ansible using the same password provided for cloud\_user:

su - ansible

### Configure the media Host Group to Contain media1 and media2

Create the inventory file:

touch /home/ansible/inventory

Edit the inventory file:

vim /home/ansible/inventory

Paste in the following:

[media]  
media1  
media2

To save and exit the file, press **ESC**, type :wq, and press **Enter**.

### Define Variables for media with Their Accompanying Values

Create a group\_vars directory:

mkdir group\_vars

Move into the group\_vars directory:

cd group\_vars/

Create a media file:

touch media

Edit the media file:

vim media

Paste in the following:

media\_content: /tmp/var/media/content/  
media\_index: /tmp/opt/media/mediaIndex

To save and exit the file, press **ESC**, type :wq, and press **Enter**.

### Configure the webservers Host Group to Contain the Hosts web1 and web2

Move into the home directory:

cd ~

Edit the inventory file:

vim inventory

Beneath media2, paste in the following:

[webservers]  
web1  
web2

To save and exit the file, press **ESC**, type :wq, and press **Enter**.

### Define Variables for webservers with Their Accompanying Values

Move into the group\_vars directory:

cd group\_vars/

Create a webservers file:

touch webservers

Edit the file:

vim webservers

Paste in the following:

httpd\_webroot: /var/www/  
httpd\_config: /etc/httpd/

To save and exit the file, press **ESC**, type :wq, and press **Enter**.

### Define the script\_files Variable for web1 and Set Its Value to /usr/local/scripts

Move into the home directory:

cd ~

Create a host\_vars directory:

mkdir host\_vars

Move into the host\_vars directory:

cd host\_vars/

Create a web1 file:

touch web1

Edit the file:

vim web1

Paste in the following:

script\_files: /tmp/usr/local/scripts

To save and exit the file, press **ESC**, type :wq, and press **Enter**.

## Testing

Return to the home directory:

cd ~

Run the backup script:

/home/ansible/scripts/backup.sh

If you have correctly configured the inventory, it should not error.

## Conclusion

## Congratulations — You've completed this hands-on lab!

## Additional Resources

Your company decided that their backup software license was frivolous and unnecessary. Because of this, the license was not renewed. As a stopgap measure, your supervisor has created a simple script and an Ansible playbook to create an archive of select files, depending on pre-defined Ansible host groups. You will create the inventory file to complete the backup strategy.

#### **Important notes:**

* For your convenience, Ansible has been installed on the control node.
* The user ansible has already been created on all servers with appropriate shared keys for access to managed servers from the control node.
* The ansible user has the same password as cloud\_user.
* /etc/hosts entries have been made on control1 for the managed servers.

## Learning Objectives

0 of 6 completed

Create the inventory File in /home/ansible/

Create the inventory file in /home/ansible/.

Configure the media Host Group to Contain media1 and media2

Use an editor, such as Vim, to configure the media host group to contain media1 and media2.

Define Variables for media with Their Accompanying Values

Define the following variables for media with their accompanying values:

* media\_content should be set to /tmp/var/media/content/.
* media\_index should be set to /tmp//opt/media/mediaIndex.

Configure the webservers Host Group to Contain the Hosts web1 and web2

Configure the webservers host group to contain the hosts web1 and web2.

Define Variables for webservers with Their Accompanying Values

Define the following variables for webservers with their accompanying values:

* httpd\_webroot should be set to /var/www/.
* httpd\_config should be set to /etc/httpd/.

Define the script\_files Variable for web1 and Set Its Value to /usr/local/scripts

Define the variable script\_files specifically for web1. The value of script\_files should be set to /tmp/usr/local/scripts.

To test your inventory, run /home/ansible/scripts/backup.sh.

If you have correctly configured the inventory, it should not error.

**Note:** Do not edit anything in /home/ansible/scripts/.

[root@control1 cloud\_user]# curl 169.254.169.254/latest/user-data#!/bin/bash/bin/echo 'dHp6uK]F' | /bin/passwd cloud\_user —stdin

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

sed -i s/requiretty/\!requiretty/ /etc/sudoers

hostnamectl set-hostname control1

/bin/echo 10.0.1.64 node1 web1 media1>> /etc/hosts

/bin/echo 10.0.1.200 node2 web2 media2>> /etc/hostsyum-config-manager —enable rhui-REGION-rhel-server-extras

cd /tmp

wget <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpmyum> install -y epel-release-latest-7.noarch.rpmyum install -y ansibleyum install -y sshpass

/sbin/useradd ansible

/bin/echo 'dHp6uK]F' | /bin/passwd ansible —stdin/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

sudo -u ansible /bin/mkdir -p /home/ansible/.ssh

sudo -u ansible /bin/ssh-keygen -q -N "" -f /home/ansible/.ssh/id\_rsasudo -u ansible sshpass -p 'dHp6uK]F' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@localhost

sudo -u ansible sshpass -p 'dHp6uK]F' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@node1

sudo -u ansible sshpass -p 'dHp6uK]F' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@node2

/bin/yum install -y git

git clone <https://github.com/linuxacademy/content-rh-ansible.git> /root/content-rh-ansible

/bin/mv /root/content-rh-ansible/lab3/control/scripts /home/ansible/scripts

/bin/chmod +x /home/ansible/scripts/backup.sh

/bin/chown -R ansible:ansible /home/ansible

/bin/rm -rf /root/content-rh-ansible

/usr/bin/pip3 install <https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz>

/usr/local/bin/cfn-signal -e 0 —stack cfst-602-0b0e9ef05251d7d1b16fd4b5742ac8d0 —resource Control1 —region us-east-1

[root@control1 cloud\_user]# ansible —version

ansible 2.9.27

config file = /etc/ansible/ansible.cfg

configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules'] ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /bin/ansible

python version = 2.7.5 (default, May 27 2022, 11:27:32) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]

[root@control1 cloud\_user]# sudo cat /etc/os-release

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

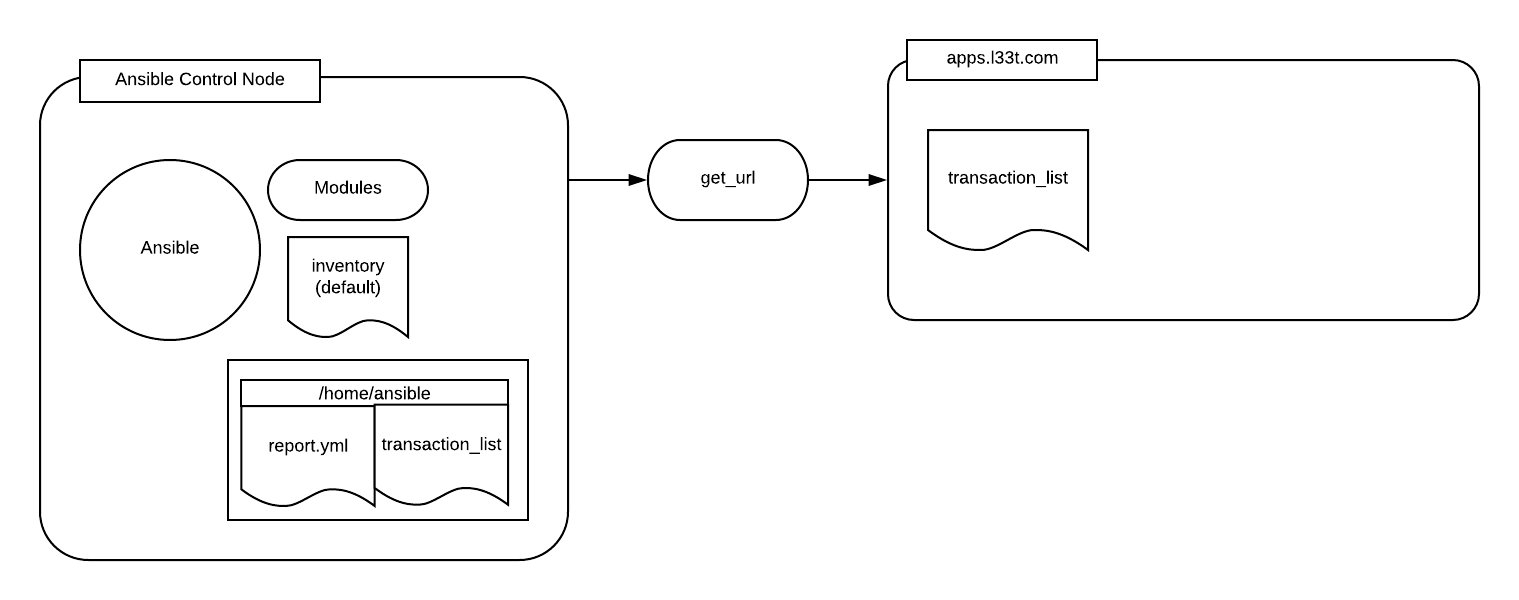
REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[root@control1 cloud\_user]#

5

# Ansible Playbooks - Error Handling

# 



# Ansible Playbooks - Error Handling

# *This course is not approved or sponsored by Red Hat.*

## The Scenario

## We have to set up automation to pull down a data file, from a notoriously unreliable third-party system, for integration purposes. Create a playbook that attempts to pull down [*http://apps.l33t.com/transaction\_list*](http://apps.l33t.com/transaction_list) to localhost. The playbook should gracefully handle the site being down by outputting the message "[l33t.com](http://l33t.com) appears to be down. Try again later." to *stdout*. If the task succeeds, the playbook should write "File downloaded." to *stdout*. No matter if the playbook errors or not, it should always output "Attempt completed." to *stdout*.

If the report is collected, the playbook should write and edit the file to replace all occurrences of #BLANKLINE with a line break \n.

Tasks list summary:

* Create a playbook, /home/ansible/report.yml.
* Configure the playbook to download [*http://apps.l33t.com/transaction\_list*](http://apps.l33t.com/transaction_list) to /home/ansible/transaction\_list on localhost and output "File downloaded." to stdout.
* Configure the playbook to handle connection failure by outputting "[l33t.com](http://l33t.com) appears to be down. Try again later." to stdout.
* Configure the playbook to output "Attempt Completed" to stdout, whether it was successful or not.
* Configure the playbook to replace all instances of #BLANKLINE with the line break character \n.
* Run the playbook using the default inventory to verify whether things work or not.

*Important notes:*

* For convenience, Ansible has been installed on the control node.
* The user ansible already exists on all servers, with appropriate shared keys for access to the necessary servers from the control node.
* The ansible user has the same password as cloud\_user.
* All necessary Ansible inventories have already been created.
* [*apps.l337.com*](http://apps.l337.com) is unavailable by default.
* We may force a state change by running /home/ansible/scripts/change\_l33t.sh.

## Logging In

## Log into the control node (control1) as the ansible user, using login credentials on the hands-on lab overview page.

## Create a playbook: /home/ansible/report.yml

## Create the file with an echo command:

[ansible@control1]$ echo "---" >> /home/ansible/report.yml

Using a text editor, such as vim, edit /home/ansible/report.yml

[ansible@control1]$ vim /home/ansible/report.yml

## Configure the Playbook to Download a File and Output a Message

## First, we'll specify our **host** and **tasks** (**name**, and **debug** message):

---  
- hosts: localhost  
  tasks:  
    - name: download transaction\_list  
      get\_url:  
        url: http://apps.l33t.com/transaction\_list  
        dest: /home/ansible/transaction\_list  
    - debug: msg="File downloaded"

## Reconfigure the Playbook to Handle Connection Failure by Outputting a Message

## We need to reconfigure a bit here, adding a **block** keyword and a **rescue**, in case the URL we're reaching out to is down:

---  
- hosts: localhost  
  tasks:  
    - name: download transaction\_list  
      block:  
        - get\_url:  
            url: http://apps.l33t.com/transaction\_list  
            dest: /home/ansible/transaction\_list  
        - debug: msg="File downloaded"  
      rescue:  
        - debug: msg="l33t.com appears to be down.  Try again later."

## Configure the Playbook to Output a Message Whether It Was Successful or Not

## An **always** block here will let us know that the playbook at least made an attempt to download the file:

---  
- hosts: localhost  
  tasks:  
    - name: download transaction\_list  
      block:  
        - get\_url:  
            url: http://apps.l33t.com/transaction\_list  
            dest: /home/ansible/transaction\_list  
        - debug: msg="File downloaded"  
      rescue:  
        - debug: msg="l33t.com appears to be down.  Try again later."  
      always:  
        - debug: msg="Attempt completed."

## Configure the Playbook to Replace All Instances of #BLANKLINE with the Line Break Character \n

## We can use the **replace** module for this task, and we'll sneak it in between the **get\_url** and first **debug** tasks.

---  
- hosts: localhost  
  tasks:  
    - name: download transaction\_list  
      block:  
        - get\_url:  
            url: http://apps.l33t.com/transaction\_list  
            dest: /home/ansible/transaction\_list  
        - replace:  
            path: /home/ansible/transaction\_list  
            regexp: "#BLANKLINE"  
            replace: '\n'  
        - debug: msg="File downloaded"  
      rescue:  
        - debug: msg="l33t.com appears to be down.  Try again later."  
      always:  
        - debug: msg="Attempt completed."

## Verify Configuration by Running the Playbook

## We can run the playbook with this:

[ansible@control1]$ ansible-playbook /home/ansible/report.yml

If all went well, we can read the downloaded text file:

[ansible@control1]$ cat /home/ansible/transaction\_list

The file looks ok. Let's read the original, up where it sits on l33t.com:

[ansible@control1]$ curl apps.l33t.com/transaction\_list

We'll see instances of #BLANKLINE there that our playbook actually turned into new lines.

Now we'll test to see how gracefully we deal with errors. We'll shut l33t.com down:

[ansible@control1]$ ./scripts/change\_l33t.sh

Then we can run our playbook again:

[ansible@control1]$ ansible-playbook /home/ansible/report.yml

In the output, we see that the **get\_url** task failed, but that the playbook did not stop executing. It outputted all of the appropriate messages.

## Conclusion

## We successfully created a playbook that downloads a file from a URL, and errors out gracefully when something (like the URL is down) goes awry. Congratulations!

## Additional Resources

We have to set up automation to pull down a data file, from a notoriously unreliable third-party system, for integration purposes. Create a playbook that attempts to pull down [*http://apps.l33t.com/transaction\_list*](http://apps.l33t.com/transaction_list) to localhost. The playbook should gracefully handle the site being down by outputting the message "[l33t.com](http://l33t.com) appears to be down. Try again later." to *stdout*. If the task succeeds, the playbook should write "File downloaded." to *stdout*. No matter if the playbook errors or not, it should always output "Attempt completed." to *stdout*.

If the report is collected, the playbook should write and edit the file to replace all occurrences of #BLANKLINE with a line break \n.

Tasks list summary:

* Create a playbook, /home/ansible/report.yml.
* Configure the playbook to download [*http://apps.l33t.com/transaction\_list*](http://apps.l33t.com/transaction_list) to /home/ansible/transaction\_list on localhost and output "File downloaded." to stdout.
* Configure the playbook to handle connection failure by outputting "[l33t.com](http://l33t.com) appears to be down. Try again later." to stdout.
* Configure the playbook to output "Attempt Completed" to stdout, whether it was successful or not.
* Configure the playbook to replace all instances of #BLANKLINE with the line break character \n.
* Run the playbook using the default inventory to verify whether things work or not.

Important notes:

* For convenience, Ansible has been installed on the control node.
* The user ansible already exists on all servers, with appropriate shared keys for access to the necessary servers from the control node.
* The ansible user has the same password as cloud\_user.
* All necessary Ansible inventories have already been created.
* [*apps.l337.com*](http://apps.l337.com) is unavailable by default.
* We may force a state change by running /home/ansible/scripts/change\_l33t.sh.

## Learning Objectives

0 of 6 completed

Create a playbook: /home/ansible/report.yml

echo "---" >> /home/ansible/report.yml

Configure the Playbook to Download \*http://apps.l33t.com/transaction\_list\* to `/home/ansible/transaction\_list` on `localhost` and Outputs the Message "File downloaded." to stdout

Using a text editor, such as vim, edit /home/ansible/report.yml to contain the following text block below the line containing "---":

- hosts: localhost  
  tasks:  
    - name: download transaction\_list  
      get\_url:  
        url: http://apps.l33t.com/transaction\_list   
        dest: /home/ansible/transaction\_list  
    - debug: msg="File downloaded"

Configure the Playbook to Handle Connection Failure by Outputting "l33t.com appears to be down. Try again later." to stdout

Using a text editor, such as vim, edit the tasks section in /home/ansible/report.yml to contain the new lines as shown below. Note that the get\_url line was changed to include a leading hyphen:

---  
- hosts: localhost  
  tasks:  
    - name: download transaction\_list  
      block:  
        - get\_url:  
            url: http://apps.l33t.com/transaction\_list  
            dest: /home/ansible/transaction\_list  
        - debug: msg="File downloaded"  
      rescue:  
        - debug: msg="l33t.com appears to be down.  Try again later."

Configure the Playbook to Output "Attempt Completed" to `stdout`, Whether It Was Successful or Not

Using a text editor, such as vim, edit the tasks section in /home/ansible/report.yml to contain the new lines as shown below.

---  
- hosts: localhost  
  tasks:  
    - name: download transaction\_list  
      block:  
        - get\_url:  
            url: http://apps.l33t.com/transaction\_list  
            dest: /home/ansible/transaction\_list  
        - debug: msg="File downloaded"  
      rescue:  
        - debug: msg="l33t.com appears to be down.  Try again later."  
      always:  
        - debug: msg="Attempt completed."

Configure the Playbook to Replace All Instances of `#BLANKLINE` with the Line Break Character \n

Using a text editor, such as vim, edit the block section in /home/ansible/report.yml to contain the new lines as shown below:

---  
- hosts: localhost  
  tasks:  
    - name: download transaction\_list  
      block:  
        - get\_url:  
            url: http://apps.l33t.com/transaction\_list  
            dest: /home/ansible/transaction\_list  
        - replace:   
            path: /home/ansible/transaction\_list   
            regexp: "#BLANKLINE"  
            replace: '\n'  
        - debug: msg="File downloaded"  
      rescue:  
        - debug: msg="l33t.com appears to be down.  Try again later."  
      always:  
        - debug: msg="Attempt completed."

Verify Configuration by Running the Playbook

ansible-playbook /home/ansible/report.yml

CONTROL NODE :

[root@control1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo '0WwS=R[\*' | /bin/passwd cloud\_user —stdin

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

sed -i s/requiretty/\!requiretty/ /etc/sudoershostnamectl set-hostname control1

/bin/echo 10.0.1.157 apps.l33t.com node1>> /etc/hosts

rpm -Uvh <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

yum-config-manager —enable rhui-REGION-rhel-server-extrasyum install -y ansibleyum install -y sshpass

/sbin/useradd ansible/bin/echo '0WwS=R[\*' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

sudo -u ansible /bin/mkdir -p /home/ansible/.ssh

sudo -u ansible /bin/ssh-keygen -q -N "" -f /home/ansible/.ssh/id\_rsa

sudo -u ansible sshpass -p '0WwS=R[\*' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@localhost

sudo -u ansible sshpass -p '0WwS=R[\*' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@node1

echo "localhost" >> /etc/ansible/hostsecho "node1" >> /etc/ansible/hosts

yum install -y nmap-ncat

/bin/yum install -y git

git clone <https://github.com/linuxacademy/content-rh-ansible.git> /root/content-rh-ansible

/bin/mv /root/content-rh-ansible/lab5/control/scripts /home/ansible/scripts

/bin/rm -rf /root/content-rh-ansiblechmod +x /home/ansible/scripts/change\_l33t.sh

chown -R ansible:ansible /home/ansible

[root@control1 cloud\_user]# ansible —version

ansible 2.9.27

config file = /etc/ansible/ansible.cfg configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /bin/ansible python version = 2.7.5 (default, May 27 2022, 11:27:32) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]

[root@control1 cloud\_user]# sudo cat /etc/os-release

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[root@control1 cloud\_user]#

NODE 1 :

[root@node1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo '0WwS=R[\*' | /bin/passwd cloud\_user —stdin

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

/usr/bin/hostnamectl set-hostname node1

/sbin/useradd ansible

/bin/echo '0WwS=R[\*' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

rpm -Uvh <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

yum install -y httpd

systemctl start httpd

systemctl enable httpd/bin/yum install -y git

git clone <https://github.com/linuxacademy/content-rh-ansible.git> /root/content-rh-ansible

/bin/mv /root/content-rh-ansible/lab5/managed/transaction\_list /var/www/html/transaction\_list

restorecon /var/www/html/transaction\_list

/bin/rm -rf /root/content-rh-ansible

[root@node1 cloud\_user]# ansible —version

bash: ansible: command not found

[root@node1 cloud\_user]# sudo cat /etc/os-release

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"HOME\_URL="https://www.redhat.com/"

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

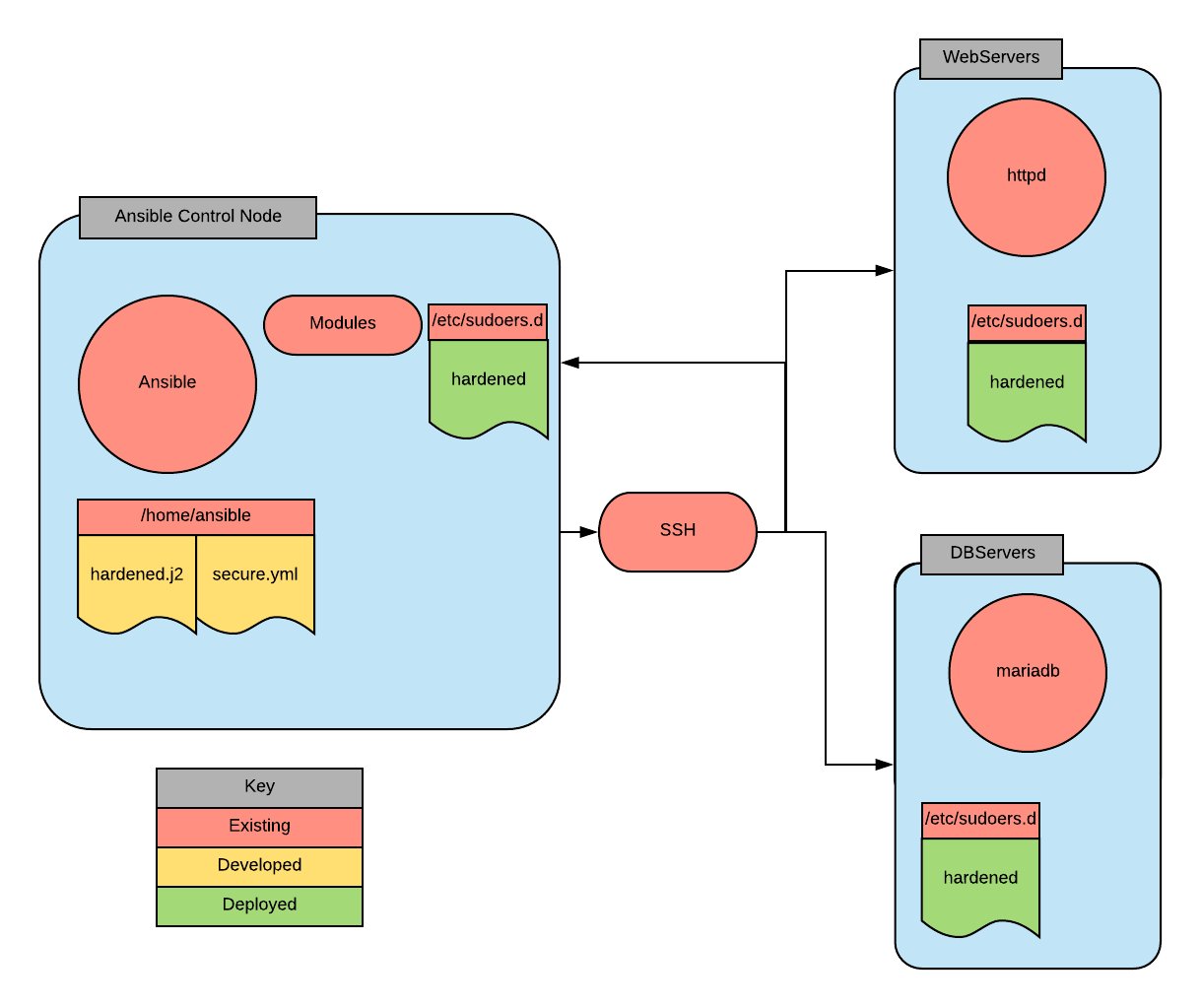
REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"[root@node1 cloud\_user]#

6:

## Working with Ansible Templates, Variables, and Facts



# Working with Ansible Templates, Variables, and Facts

# *This course is not approved or sponsored by Red Hat.*

## The Scenario

## A colleague was the unfortunate victim of a scam email, and their network account was compromised. Shortly after we finished helping them pack up their desk, our boss gave us the assignment to promote system security by deploying a hardened sudoers file. We need to create an Ansible template of the sudoers file.

We also need to create an accompanying playbook in /home/ansible/security.yml that will deploy this template to all servers in the default inventory.

### Important notes:

* Ansible has been installed on the control node.
* The user ansible has been already created on all servers with the appropriate shared keys for access to the necessary servers from the control node. It has the same password as cloud\_user.
* All necessary Ansible inventories have already been created.

## Logging In

## Log into the control node (control1) as the ansible user, using login credentials on the hands-on lab overview page.

## Create a Template *sudoers* File

[ansible@control1]$ vim /home/ansible/hardened.j2

Now that we're in Vim, we'll put these contents in the file:

%sysops {{ ansible\_default\_ipv4.address }} = (ALL) ALL  
Host\_Alias WEBSERVERS = {{ groups['web']|join(' ') }}  
Host\_Alias DBSERVERS = {{ groups['database']|join(' ') }}  
%httpd WEBSERVERS = /bin/su - webuser  
%dba DBSERVERS = /bin/su - dbuser

## Create a Playbook

[ansible@control1]$ vim /home/ansible/security.yml

The security.yml file should look like this:

---  
- hosts: all  
  become: yes  
  tasks:  
  - name: deploy sudo template  
    template:  
      src: /home/ansible/hardened.j2  
      dest: /etc/sudoers.d/hardened  
      validate: /sbin/visudo -cf %s

## Run the Playbook

[ansible@control1]$ ansible-playbook /home/ansible/security.yml

The output will show that everything deployed fine, but we can check locally to make sure. Let's become root (with sudo su -) and then read our file:

[ansible@control1]$ sudo cat /etc/sudoers.d/hardened

The custom IP and host aliases are in there.

## Conclusion

## Congratulations on completing the lab!

## Additional Resources

A colleague of yours was the unfortunate victim of a scam email, and their network account was compromised. Shortly after you finished helping them pack up their desk, your boss gave you the assignment to promote system security through deploying a hardened *sudoers* file. You will need to create an Ansible template of the *sudoers* file that meets the following criteria:

* A file named */etc/sudoers.d/hardened* to deploy on all ansible inventory servers. WARNING: Do NOT edit the default *sudoers* file, doing so may break your exercise environment. Additionally, always validate any file placed in */etc/sudoers.d* with /sbin/visudo -cf <filename> prior to deployment!!
* Grant users in the *sysops* group the ability to run all commands as *root* for each local system by IP address. This would be what the entry in your result file except with the target system's IP: %sysops 34.124.22.55 = (ALL) ALL.
* Define the *host\_alias* group WEBSERVERS to contain all servers in the *ansible web inventory* group: Host\_Alias WEBSERVERS = <host name>
* Define the *host\_alias* group *DBSERVERS* to contain all servers in the *ansible database inventory* group: Host\_Alias DBSERVERS = <host name>
* Grant users in the *httpd* group the ability to sudo su - webuser on the *WEBSERVERS* hosts: %httpd WEBSERVERS = /bin/su - webuser
* Grant users in the dba group sudo su - dbuser on the DBSERVERS hosts: %dba DBSERVERS = /bin/su - dbuser
* The file must be validated using /sbin/visudo -cf before deployment.

You will need to create an accompanying playbook in */home/ansible/security.yml* that will deploy this template to all servers in the default inventory.

Summary tasks list:

* Create a template *sudoers* file in \*/home/ansible/hardened.j2 \*that produces a file with appropriate output for each host.
* The deployed file should resemble the following, except with the *IP* and *hostnames* customized appropriately:

%sysops 34.124.22.55 = (ALL) ALL  
Host\_Alias WEBSERVERS = server1, server2  
Host\_Alias DBSERVERS = serverA, serverB  
%httpd WEBSERVERS = /bin/su - webuser  
%dba DBSERVERS = /bin/su - dbuser

* Create a playbook in */home/ansible/security.yml* that uses the template module to deploy the template on all servers in the default ansible inventory after validating the syntax of the generated file.
  + Note: You may find it easier to have the play output to */home/ansible/test* and validate manually using /sbin/visudo -cf <filename> before using the template module's validate.
  + IMPORTANT: Do not deploy any file to */etc/sudoers.d/* without first validating with visudo! A syntax error in a *sudoers* file will break sudo on the system and require starting the exercise over again!
  + Note: The video shows the use of join(' ') which is a typo. To support multiple hosts in the sudoers file it should instead be join(', ')
* Run the playbook and ensure the files deployed correctly.

Important notes:

* For your convenience, Ansible has been installed on the control node.
* The user *ansible* has been already created on all servers with the appropriate shared keys for access to the necessary servers from the control node. It has the same password as *cloud\_user*.
* All necessary Ansible inventories have been created for you.

## Learning Objectives

0 of 4 completed

Create a Template \*sudoers\* File in `/home/ansible/hardened.j2` That Produces a File with Appropriate Output for Each Host

touch /home/ansible/hardened.j2

The Deployed File Should Resemble the Example File Except with the \*IP\* and \*hostnames\* Customized Appropriately

Edit *hardened.j2* to contain the following text:

    %sysops {{ ansible\_default\_ipv4.address }} = (ALL) ALL  
    Host\_Alias WEBSERVERS = {{ groups['web']|join(', ') }}  
    Host\_Alias DBSERVERS = {{ groups['database']|join(', ') }}   
    %httpd WEBSERVERS = /bin/su - webuser  
    %dba DBSERVERS = /bin/su - dbuser

Create a Playbook in `/home/ansible/security.yml` That Uses the Template Module to Deploy the Template on All Servers in the Default Ansible Inventory After Validating the Syntax of the Generated File

Edit */home/ansible/security.yml* to contain the following:

---  
- hosts: all  
  become: yes  
  tasks:  
  - name: deploy sudo template  
    template:  
      src: /home/ansible/hardened.j2  
      dest: /etc/sudoers.d/hardened  
      validate: /sbin/visudo -cf %s

Run the Playbook and Ensure the Files Are Correctly Deployed

ansible-playbook /home/ansible/security.yml

Check the local */etc/sudoers.d/hardened* on the *ansible control* node for the correct contents.

CONTROL NODE :

[root@control1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo ']6kg)E8e' | /bin/passwd cloud\_user —stdin

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_configsed -i s/requiretty/\!requiretty/ /etc/sudoers

hostnamectl set-hostname control1

/bin/echo 10.0.1.111 node1>> /etc/hosts

/bin/echo 10.0.1.215 node2>> /etc/hosts

yum-config-manager —enable rhel-7-server-rhui-extras-rpms

cd /tmp

wget <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

yum install -y epel-release-latest-7.noarch.rpm

yum install -y ansible

yum install -y sshpass

/sbin/useradd ansible

/bin/echo ']6kg)E8e' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

sudo -u ansible /bin/mkdir -p /home/ansible/.ssh

sudo -u ansible /bin/ssh-keygen -q -N "" -f /home/ansible/.ssh/id\_rsa

sudo -u ansible sshpass -p ']6kg)E8e' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@localhost

sudo -u ansible sshpass -p ']6kg)E8e' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@node1

sudo -u ansible sshpass -p ']6kg)E8e' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@node2

echo "localhost" >> /etc/ansible/hosts

echo "[nodes]" >> /etc/ansible/hosts

echo "node1" >> /etc/ansible/hosts

echo "node2" >> /etc/ansible/hosts

echo "[web]" >> /etc/ansible/hosts

echo "node1" >> /etc/ansible/hosts

echo "[database]" >> /etc/ansible/hosts

echo "node2" >> /etc/ansible/hosts

groupadd sysops

[root@control1 cloud\_user]# ansible —version

ansible 2.9.27

config file = /etc/ansible/ansible.cfg

configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /bin/ansible

python version = 2.7.5 (default, May 27 2022, 11:27:32) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]

[root@control1 cloud\_user]# cat /etc/os-release

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"

ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"

PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[root@control1 cloud\_user]#

NODE1 and NODE2 :

[root@node1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo ']6kg)E8e' | /bin/passwd cloud\_user —stdin

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

/usr/bin/hostnamectl set-hostname node1

/usr/bin/yum-config-manager —enable rhel-7-server-rhui-extras-rpms

cd /tmp

wget <https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

yum install -y epel-release-latest-7.noarch.rpm

/sbin/useradd ansible

/bin/echo ']6kg)E8e' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoersyum install -y httpd

systemctl start httpd

systemctl enable httpd

groupadd httpd

useradd webuser

groupadd sysops

[root@node1 cloud\_user]# ansible —version

bash: ansible: command not found

[root@node1 cloud\_user]# cat /etc/os-release

NAME="Red Hat Enterprise Linux Server"

VERSION="7.9 (Maipo)"

ID="rhel"ID\_LIKE="fedora"

VARIANT="Server"

VARIANT\_ID="server"

VERSION\_ID="7.9"PRETTY\_NAME="Red Hat Enterprise Linux Server 7.9 (Maipo)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:7.9:GA:server"

HOME\_URL="<https://www.redhat.com/>“

BUG\_REPORT\_URL="<https://bugzilla.redhat.com/>“

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 7"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=7.9

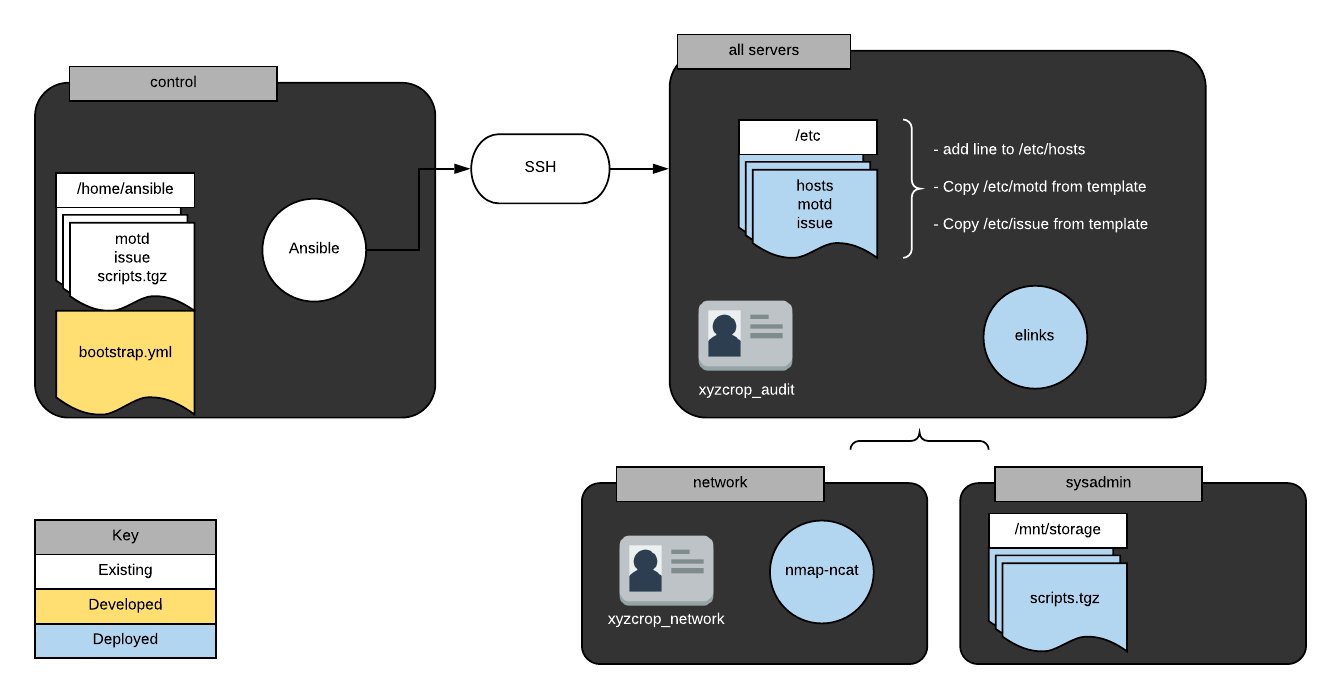
REDHAT\_SUPPORT\_PRODUCT="Red Hat Enterprise Linux"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7.9"

[root@node1 cloud\_user]#

6:

## Writing Your First Ansible Playbook.



Writing Your First Ansible Playbook  
Introduction  
Playbooks are the heart of Ansible. They provide a means of executing numerous tasks across any number of systems. This exercise sets you on the path to Ansible mastery by allowing you to craft and execute an Ansible playbook.  
  
Solution  
Begin by logging in to the Ansible control node using the credentials on the hands-on lab page:  
  
ssh cloud\_user@PUBLIC\_IP\_ADDRESS  
Become the ansible user:  
  
sudo -i -u ansible      
Create a Basic Playbook for All Servers in the Ansible Inventory  
Create the file /home/ansible/bootstrap.yml:  
  
vim /home/ansible/bootstrap.yml  
Add the following content:  
  
---  
- hosts: all  
  become: yes  
  tasks:  
    - name: edit host file  
      lineinfile:  
        path: /etc/hosts  
        line: "ansible.xyzcorp.com 169.168.0.1"  
  
    - name: install elinks  
      package:  
        name: elinks  
        state: latest  
  
    - name: create audit user  
      user:  
        name: xyzcorp\_audit  
        state: present  
  
    - name: update motd  
      copy:  
        src: /home/ansible/motd  
        dest: /etc/motd  
  
    - name: update issue  
      copy:  
        src: /home/ansible/issue  
        dest: /etc/issue  
Add a Section to the Playbook for the Network Servers in the Ansible inventory  
Add a play in the file /home/ansible/bootstrap.yml similar to the following:  
  
- hosts: network  
  become: yes  
  tasks:  
    - name: install netcat  
      yum:  
        name: nmap-ncat  
        state: latest  
  
    - name: create network user  
      user:  
        name: xyzcorp\_network  
        state: present            
Add a Section to the Playbook for the SysAdmin Servers in the Ansible Inventory  
Add a play in the file /home/ansible/bootstrap.yml similar to the following:  
  
- hosts: sysadmin  
  become: yes  
  tasks:  
    - name: copy tarball  
      copy:  
        src: /home/ansible/scripts.tgz  
        dest: /mnt/storage/  
Execute Playbook to Verify Your Playbook Works Correctly  
Run the playbook from the control node:  
  
ansible-playbook /home/ansible/bootstrap.yml  
Conclusion  
Congratulations, you've completed this hands-on lab!

Additional Resources  
Your supervisor has asked you to find a way to automate and audit basic system configuration for new servers in your environment. Given that Ansible is already configured at a basic level in your environment, the simplest solution would be to just write a playbook for bootstrapping your new hosts. Create a playbook called /home/ansible/bootstrap.yml to fulfill the following boot strap requirements:  
  
All servers:  
  
Edit /etc/hosts to include the following entry: ansible.xyzcorp.com 169.168.0.1  
Install elinks  
Create the user xyzcorp\_audit  
Copy the files /home/ansible/motd and /home/ansible/issue to /etc/  
Network servers:  
  
Install nmap-ncat  
Create the user xyzcorp\_network  
SysAdmin servers:  
  
Copy /home/ansible/scripts.tgz from the control node to /mnt/storage  
The Ansible control node has been configured for you and each testing server has already been configured for use with Ansible. The default inventory has been configured to include a the groups network and sysadmin. Each group includes a sample host.  
  
Learning Objectives  
0 of 4 completed  
  
  
Create a Basic Playbook for All Servers in the Ansible Inventory  
  
Create a basic playbook in /home/ansible/bootstrap.yml that completes the noted task required for all servers in the Ansible inventory.  
  
Create the file /home/ansible/bootstrap.yml and add the following content:  
  
---  
- hosts: all  
  become: yes  
  tasks:  
    - name: edit host file  
      lineinfile:  
        path: /etc/hosts  
        line: "ansible.xyzcorp.com 169.168.0.1"  
  
    - name: install elinks  
      package:  
        name: elinks  
        state: latest  
  
    - name: create audit user  
      user:  
        name: xyzcorp\_audit  
        state: present  
  
    - name: update motd  
      copy:  
        src: /home/ansible/motd  
        dest: /etc/motd  
  
    - name: update issue  
      copy:  
        src: /home/ansible/issue  
        dest: /etc/issue  
  
Add a Section to the Playbook for the Network Servers in the Ansible Inventory  
  
Add a section to the playbook in /home/ansible/bootstrap.yml that completes the noted task required for the network servers in the Ansible inventory.  
  
Create a play in the file /home/ansible/bootstrap.yml similar to the following:  
  
 - hosts: network  
   become: yes  
   tasks:  
     - name: install netcat  
       yum:  
         name: nmap-ncat  
         state: latest  
     - name: create network user  
       user:  
         name: xyzcorp\_network  
         state: present  
  
Add a Section to the Playbook for the SysAdmin Servers in the Ansible Inventory  
  
Add a section to the playbook in /home/ansible/bootstrap.yml that completes the noted task required for the sysadmin servers in the Ansible inventory.  
  
Create a play in the file /home/ansible/bootstrap.yml similar to the following:  
  
- hosts: sysadmin  
  become: yes  
  tasks:  
    - name: copy tarball  
      copy:  
        src: /home/ansible/scripts.tgz  
        dest: /mnt/storage/  
  
Execute Playbook to Verify Your Playbook Works Correctly  
  
Execute playbook /home/ansible/bootstrap.yml to verify your playbook works correctly.  
  
Run ansible-playbook /home/ansible/bootstrap.yml from the control node.

CONTROL NODE :

[root@control cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/bin/echo 'IH4feBu-' | /bin/passwd root —stdin

/bin/echo 'IH4feBu-' | /bin/passwd cloud\_user —stdin

curl <https://raw.githubusercontent.com/ACloudGuru-Resources/Linux/master/my-custom.lang.sh> -o /etc/profile.d/my-custom.lang.sh

source /etc/profile.d/my-custom.lang.sh/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

sed -i s/requiretty/\!requiretty/ /etc/sudoershostnamectl set-hostname control/bin/echo 10.0.1.105 node1>> /etc/hosts/bin/echo 10.0.1.123 node2>> /etc/hosts

yum install -y python3-pip

yum install -y epel-release

yum install -y ansible

yum install -y sshpass/usr/bin/pip3 install <https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz>

/sbin/useradd ansible

/bin/echo 'IH4feBu-' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

sudo -u ansible /bin/mkdir -p /home/ansible/.sshsudo -u ansible /bin/ssh-keygen -q -N "" -f /home/ansible/.ssh/id\_rsa

sudo -u ansible sshpass -p 'IH4feBu-' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@localhostsudo -u ansible sshpass -p 'IH4feBu-' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@node1sudo -u ansible sshpass -p 'IH4feBu-' ssh-copy-id -i /home/ansible/.ssh/id\_rsa.pub ansible@node2

/bin/mkdir /tmp/scripts/bin/echo This is a secrete file>/tmp/scripts/backup.sh

/bin/echo This is a secrete file>/tmp/scripts/save-bacon.sh

/bin/echo This is a secrete file>/tmp/scripts/inspire.sh

/bin/chmod +x /tmp/scripts/\*

cd /tmp; /bin/tar -czf /home/ansible/scripts.tgz scripts; sleep 5

rm -rf /tmp/scripts

echo Authorized users only >> /home/ansible/issue

echo Welcome to XYZ Corp >> /home/ansible/motdecho "localhost" >> /etc/ansible/hosts

echo "[network]" >> /etc/ansible/hosts

echo "node1" >> /etc/ansible/hosts

echo "[sysadmin]" >> /etc/ansible/hosts

echo "node2" >> /etc/ansible/hostschown ansible.ansible /home/ansible/\*

/usr/local/bin/cfn-signal -e 0 —stack cfst-1747-c78532e7c23b14cbb624a0b41b63ab44 —resource Control —region us-east-1

[root@control cloud\_user]# ansible —version

ansible 2.9.27

config file = /etc/ansible/ansible.cfg

configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /bin/ansible

python version = 2.7.5 (default, Nov 16 2020, 22:23:17) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]

[root@control cloud\_user]# cat /etc/os-release

NAME="CentOS Linux"

VERSION="7 (Core)"

ID="centos"

ID\_LIKE="rhel fedora"

VERSION\_ID="7"

PRETTY\_NAME="CentOS Linux 7 (Core)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:centos:centos:7"

HOME\_URL="<https://www.centos.org/>“

BUG\_REPORT\_URL="<https://bugs.centos.org/>“

CENTOS\_MANTISBT\_PROJECT="CentOS-7"

CENTOS\_MANTISBT\_PROJECT\_VERSION="7"

REDHAT\_SUPPORT\_PRODUCT="centos"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7"

[root@control cloud\_user]#

NODE 1 and NODE 2 :

[root@node1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash/bin/echo 'IH4feBu-' | /bin/passwd cloud\_user —stdin

curl <https://raw.githubusercontent.com/ACloudGuru-Resources/Linux/master/my-custom.lang.sh> -o /etc/profile.d/my-custom.lang.sh

source /etc/profile.d/my-custom.lang.shyum install -y python3-pip

/usr/bin/pip3 install <https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-py3-latest.tar.gz>

/bin/echo "StrictHostKeyChecking no" >> /etc/ssh/ssh\_config

/usr/bin/hostnamectl set-hostname node1/sbin/useradd ansible

/bin/echo 'IH4feBu-' | /bin/passwd ansible —stdin

/bin/echo "ansible ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

/usr/local/bin/cfn-signal -e 0 —stack cfst-1747-c78532e7c23b14cbb624a0b41b63ab44 —resource Node1 —region us-east-1

[root@node1 cloud\_user]# ansible —version

bash: ansible: command not found[root@node1 cloud\_user]# cat /etc/os-release

NAME="CentOS Linux"

VERSION="7 (Core)"

ID="centos"

ID\_LIKE="rhel fedora"

VERSION\_ID="7"

PRETTY\_NAME="CentOS Linux 7 (Core)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:centos:centos:7"

HOME\_URL="<https://www.centos.org/>“

BUG\_REPORT\_URL="<https://bugs.centos.org/>“

CENTOS\_MANTISBT\_PROJECT="CentOS-7"

CENTOS\_MANTISBT\_PROJECT\_VERSION="7"

REDHAT\_SUPPORT\_PRODUCT="centos"

REDHAT\_SUPPORT\_PRODUCT\_VERSION="7"

[root@node1 cloud\_user]#