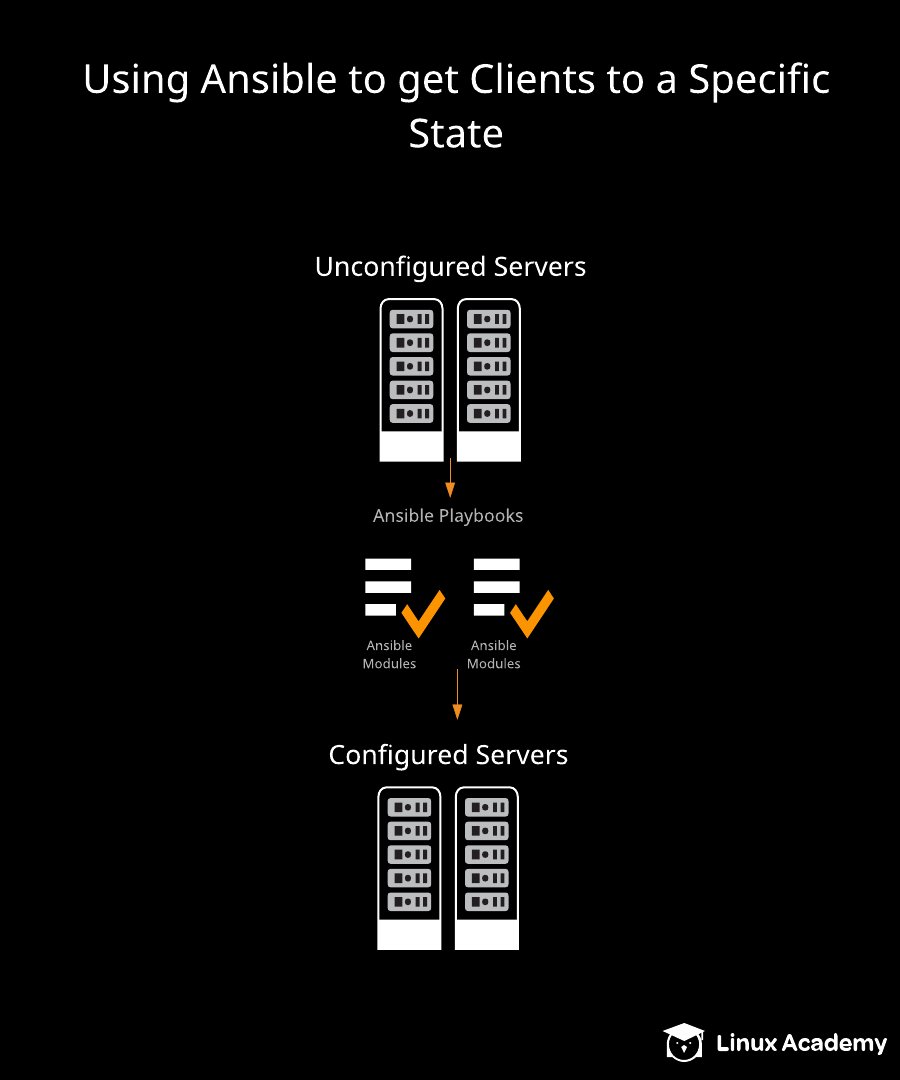
# Ansible Practioner 19-22

19 :

# Using Ansible to Get Clients to a Specific State.



# 

Using Ansible to Get Clients to a Specific State  
This course is not approved or sponsored by Red Hat.  
  
Introduction  
We often find ourselves wanting to get a group of servers set up from scratch in one fell swoop, rather than running individual configuration commands.  
  
This lab will give some practice in writing a playbook that will get things up and running all at once.  
  
The Scenario  
Management has just told us they bought 300 servers, and we've been tasked with writing a playbook to get them ready to go as quickly as possible. They've already had the OS installed and configured to work with our Ansible environment.  
  
We need to write a single playbook that will run and make sure that all the servers involved are correctly configured and in the desired state when it's finished running. We also need to ensure that all commands are idempotent.  
  
Install the linuxacademy-backup-software package throughout our environment. Due to the high potential for this repository server to be unavailable, ensure a failure doesn't stop the playbook from running.  
Install httpd on the webserver group.  
Start and enable the httpd service on the webserver group.  
Create a dba account on the dbserver group.  
Copy /root/DBAstuff.txt to the new user's home directory. Make sure that user (and the group with the same name) owns the file, and set permissions to 0600.  
Create index.html in /var/www/html on the web server.  
Ensure index.html on the web servers contains the line "Waiting for content."  
Install git on the webserver and dbserver group if it is not already.  
On each Red Hat server, create a file (/root/addresses) that contains all of the IPv4 addresses for it.  
Logging In  
Use the credentials provided on the hands-on lab page to get into Server1 to begin with. Since we need root privileges, let's just run sudo -i right off and become root.  
  
Create the Initial Playbook  
We're going to create one Ansible playbook that installs the packages we need on any number of new servers. Start off by creating the initial playbook file (we're using vim here, but use any text editor that's comfortable):  
  
vim state.yml  
Install the linuxacademy-backup-software Package throughout Our Environment  
This section of our playbook should look like:  
  
---  
- name: Install backup software  
  hosts: all  
  become: yes  
  tasks:  
   - name:  yum command  
     yum:  
      name: linuxacademy-backup-software  
      state: present  
     ignore\_errors: yes  
Install httpd on the webserver Group  
The next section of the playbook should look like this:  
  
- name: Install httpd  
  hosts: webservers  
  become: yes  
  tasks:  
   - name: httpd install  
     yum:  
      name: httpd  
      state: present  
Start and Enable the httpd Service on the webserver Group  
Continuing from the last task, the next section should look like this:  
  
   - name: Service management for httpd  
     service:  
      name: httpd  
      state: started  
      enabled: yes  
Create a dba User Account on the dbserver Group  
This portion of our playbook should look like this:  
  
- name: DB server management  
  hosts: dbservers  
  become: yes  
  tasks:  
   - name: Add user  
     user:  
      name: dba  
      state: present  
Copy /root/DBAstuff.txt to the New User's Home Directory  
Continuing along from the last task, this portion of our playbook should look like this:  
  
   - name: Copy DB user data  
     copy:  
      src: /root/DBAstuff.txt  
      dest: /home/dba/DBAstuff.txt  
      owner: dba  
      group: dba  
      mode: 0600  
Create index.html in /var/www/html on the Web Server  
This portion of our playbook should look like this:  
  
- name: Set up index.html on webservers  
  hosts: webservers  
  become: yes  
  tasks:  
   - name: Create and populate index.html  
     lineinfile:  
      path: /var/www/html/index.html  
      line: Waiting for content.  
      create: yes  
      owner: apache  
      group: apache  
      mode: 0644  
Install Git on the webserver and dbserver Groups If It Is Not Already Installed  
This next portion of our playbook should look like this:  
  
- name: Enable devs to easily populate content  
  hosts: webservers:dbservers  
  become: yes  
  tasks:  
   - name: Install git  
     yum:  
      name: git  
      state: present  
Create Red Hat Server-Specific Files  
This last portion of our playbook should look like this:  
  
- name: Red Hat specific configuration  
  hosts: all  
  become: yes  
  tasks:  
   - name: Populate file with IP addresses  
     lineinfile:  
      path: /root/addresses  
      line: "{{ ansible\_facts['all\_ipv4\_addresses'] }}"  
      create: yes  
     when: ansible\_facts['os\_family'] == 'RedHat'  
The completed playbook, with all of the sections combined, will look like the following:  
  
---  
- name: Install backup software  
  hosts: all  
  become: yes  
  tasks:  
   - name:  yum command  
     yum:  
      name: linuxacademy-backup-software  
      state: present  
     ignore\_errors: yes  
- name: Install httpd  
  hosts: webservers  
  become: yes  
  tasks:  
   - name: httpd install  
     yum:  
      name: httpd  
      state: present  
   - name: Service management for httpd  
     service:  
      name: httpd  
      state: started  
      enabled: yes  
- name: DB server management  
  hosts: dbservers  
  become: yes  
  tasks:  
   - name: Add user  
     user:  
      name: dba  
      state: present  
   - name: Copy DB user data  
     copy:  
      src: /root/DBAstuff.txt  
      dest: /home/dba/DBAstuff.txt  
      owner: dba  
      group: dba  
      mode: 0600  
- name: Set up index.html on webservers  
  hosts: webservers  
  become: yes  
  tasks:  
   - name: Create and populate index.html  
     lineinfile:  
      path: /var/www/html/index.html  
      line: Waiting for content.  
      create: yes  
      owner: apache  
      group: apache  
      mode: 0644  
- name: Enable devs to easily populate content  
  hosts: webservers:dbservers  
  become: yes  
  tasks:  
   - name: Install git  
     yum:  
      name: git  
      state: present  
- name: Red Hat specific configuration  
  hosts: all  
  become: yes  
  tasks:  
   - name: Populate file with IP addresses  
     lineinfile:  
      path: /root/addresses  
      line: "{{ ansible\_facts['all\_ipv4\_addresses'] }}"  
      create: yes  
     when: ansible\_facts['os\_family'] == 'RedHat'  
Running the Playbook  
Save the changes and exit the editor. Now, to set our playbook in motion, run this:  
  
ansible-playbook state.yml  
Conclusion  
Wow, that was quite a playbook. But taking the time to write it now saves us time in the long run. The playbook takes all of our servers (it could be ten servers, three hundred, or a thousand -- pick a number!) from blank slates to being in a state that we can use without us having to configure each one manually. That is a big deal. Congratulations!

Additional Resources  
Notice: Ansible is installed as the root user, so please work on all tasks after elevating to the root user.  
  
Management has just told us they bought 300 servers, and we've been tasked with writing a playbook to get them ready to go as quickly as possible. They've already had the OS installed and configured to work with our Ansible environment.  
  
We need to write a single playbook that will run and make sure that all the servers involved are correctly configured and in the desired state when it's finished running. We also need to ensure that all commands are idempotent.  
  
Install the linuxacademy-backup-software package throughout our environment. Due to the high potential for this repository server to be unavailable, ensure a failure doesn't stop the playbook from running.  
Install httpd on the webserver group.  
Start and enable the httpd service on the webserver group.  
Create a dba account on the dbserver group.  
Copy /root/DBAstuff.txt to the new user's home directory. Make sure that user (and the group with the same name) owns the file, and set permissions to 0600.  
Create index.html in /var/www/html on the web server.  
Ensure index.html on the web servers contains the line "Waiting for content."  
Install git on the webserver and dbserver group if it is not already.  
On each Red Hat server, create a file (/root/addresses) that contains all of the IPv4 addresses for the server.  
Learning Objectives  
0 of 9 completed  
  
Install the "linuxacademy-backup-software" Package throughout Our Environment  
  
This section of our playbook should look like:  
---  
- name: Install backup software  
  hosts: all  
  become: yes  
  tasks:  
   - name:  yum command  
     yum:  
      name: linuxacademy-backup-software  
      state: present  
     ignore\_errors: yes      
Install httpd on the webserver Group  
  
The next section of the playbook should look like this:  
- name: Install httpd  
  hosts: webservers  
  become: yes  
  tasks:  
   - name: httpd install  
     yum:  
      name: httpd  
      state: present  
   - name: Service management for httpd  
     service:  
      name: httpd  
      state: started  
      enabled: yes  
Start and Enable the httpd Service on the webserver Group  
  
Continuing from the last task, the next section should look like this:  
   - name: Service management for httpd  
     service:  
      name: httpd  
      state: started  
      enabled: yes  
Create a dba User Account on the dbserver Group  
  
This portion of our playbook should look like this:  
- name: DB server management  
  hosts: dbservers  
  become: yes  
  tasks:  
   - name: Add user  
     user:  
      name: dba  
      state: present  
Copy /root/DBAstuff.txt to the New User's Home Directory  
  
Continuing along from the last task, this portion of our playbook should look like this:  
   - name: Copy DB user data  
     copy:  
      src: /root/DBAstuff.txt  
      dest: /home/dba/DBAstuff.txt  
      owner: dba  
      group: dba  
      mode: 0600  
Create index.html in /var/www/html on the Web Server  
  
This portion of our playbook should look like this:  
- name: Set up index.html on webservers  
  hosts: webservers  
  become: yes  
  tasks:  
   - name: Create and populate index.html  
     lineinfile:  
      path: /var/www/html/index.html  
      line: Waiting for content.  
      create: yes  
      owner: apache  
      group: apache  
      mode: 0644  
Install Git on the webserver and dbserver Groups, If It Is Not Already Installed  
  
This next portion of our playbook should look like this:  
- name: Enable devs to easily populate content  
  hosts: webservers:dbservers  
  become: yes  
  tasks:  
   - name: Install git  
     yum:  
      name: git  
      state: present  
Create Red Hat Server-Specific Files  
  
This last portion of our playbook should look like this:  
- name: Red Hat specific configuration  
  hosts: all  
  become: yes  
  tasks:  
   - name: Populate file with IP addresses  
     lineinfile:  
      path: /root/addresses  
      line: "{{ ansible\_facts['all\_ipv4\_addresses'] }}"  
      create: yes  
     when: ansible\_facts['os\_family'] == 'RedHat'  
Running the Playbook  
  
To set our playbook in motion, run this:  
ansible-playbook state.yml

.

Ansible Host :

root@Server1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/usr/bin/hostnamectl set-hostname Server1

/usr/bin/yum install wget git -y

/bin/echo 'source /root/ansible/hacking/env-setup -q' >> /root/.bashrc

/bin/ssh-keygen -N '' -t rsa -f ~/.ssh/id\_rsa

/bin/cat /root/.ssh/id\_rsa.pub >> /home/ansible/.ssh/authorized\_keys

/bin/echo 'python\_interpreter=auto' >> /etc/ansible/ansible.cfg

/bin/echo 'host\_key\_checking = False' >> /etc/ansible/ansible.cfg

/sbin/alternatives —set python /usr/bin/python3

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/DBAstuff.txt> -O /root/DBAstuff.txt

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/privkey> -O /tmp/pkey

/bin/cat /tmp/pkey > /root/.ssh/id\_rsa

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/authkey> -O /tmp/akey

/bin/cat /tmp/akey > /root/.ssh/id\_rsa.pub

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/etchosts> -O /tmp/hosts

/bin/cat /tmp/hosts > /etc/hosts

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/inventory> -O /tmp/inv

/bin/cat /tmp/inv > /etc/ansible/hosts

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

/usr/bin/echo 'cloud\_user:Zl6kza]M' | /usr/sbin/chpasswd

/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-2256-c77de9f2773bd1903cd5f43d64ab5aa3 —resource PublicInstance —region us-east-1

[root@Server1 cloud\_user]# cat /etc/os-release

NAME="Red Hat Enterprise Linux"

VERSION="8.7 (Ootpa)"

ID="rhel"

ID\_LIKE="fedora"

VERSION\_ID="8.7"

PLATFORM\_ID="platform:el8"

PRETTY\_NAME="Red Hat Enterprise Linux 8.7 (Ootpa)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:8::baseos"

HOME\_URL="<https://www.redhat.com/>“

DOCUMENTATION\_URL="<https://access.redhat.com/documentation/red_hat_enterprise_linux/8/>“

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

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 8"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=8.7

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

REDHAT\_SUPPORT\_PRODUCT\_VERSION="8.7"

[root@Server1 cloud\_user]# ansible —version

ansible 2.10.0.dev0

config file = /etc/ansible/ansible.cfg

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

ansible python module location = /root/ansible/lib/ansible

executable location = /root/ansible/bin/ansible

python version = 3.6.8 (default, Jan 11 2023, 08:43:50) [GCC 8.5.0 20210514 (Red Hat 8.5.0-16)]

[root@Server1 cloud\_user]#

WEBSERVER :  
  
[root@WebServer1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/usr/bin/hostnamectl set-hostname WebServer1

/usr/bin/yum install wget -y

/sbin/useradd ansible

/usr/bin/echo 'ansible:Zl6kza]M' | /usr/sbin/chpasswd

/sbin/alternatives —set python /usr/bin/python3

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/etchosts> -O /tmp/hosts

/bin/cat /tmp/hosts > /etc/hosts

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/inventory> -O /tmp/inv

/bin/cat /tmp/inv > /etc/ansible/hosts

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/authkey> -O /tmp/key

/bin/mkdir /home/ansible/.ssh

/usr/bin/chmod 0700 /home/ansible/.ssh/

/bin/cat /tmp/key > /home/ansible/.ssh/authorized\_keys

/usr/bin/chmod 0600 /home/ansible/.ssh/authorized\_keys

/usr/bin/chown ansible.ansible /home/ansible -R

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

/usr/bin/echo 'cloud\_user:Zl6kza]M' | /usr/sbin/chpasswd

/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-2256-c77de9f2773bd1903cd5f43d64ab5aa3 —resource PublicInstance2 —region us-east-1

[root@WebServer1 cloud\_user]# cat /etc/os-release

NAME="Red Hat Enterprise Linux"

VERSION="8.9 (Ootpa)"

ID="rhel"

ID\_LIKE="fedora"

VERSION\_ID="8.9"

PLATFORM\_ID="platform:el8"

PRETTY\_NAME="Red Hat Enterprise Linux 8.9 (Ootpa)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:8::baseos"

HOME\_URL="<https://www.redhat.com/>“

DOCUMENTATION\_URL="<https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8>“

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

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 8"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=8.9

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

REDHAT\_SUPPORT\_PRODUCT\_VERSION="8.9"

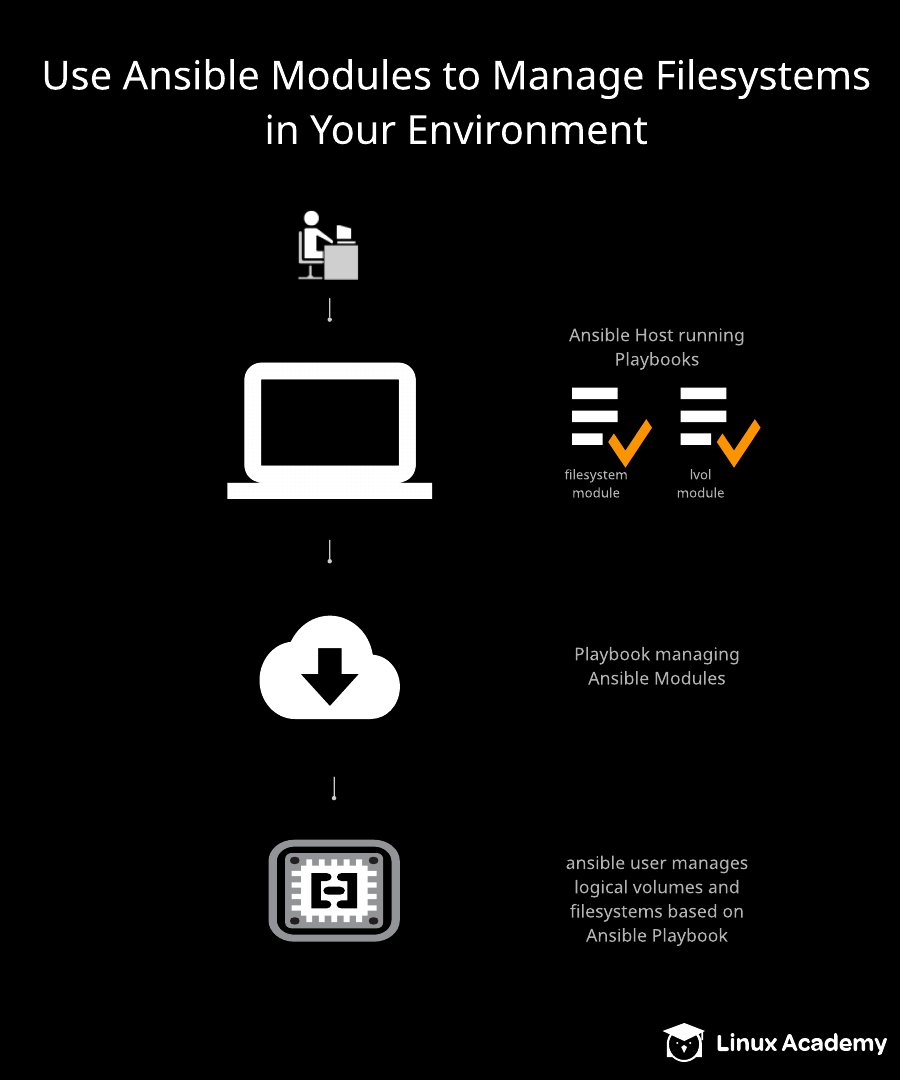
[root@WebServer1 cloud\_user]# ansible —version

bash: ansible: command not found...

rest same as lab 18

20

## Using Ansible Modules to Manage Filesystems in Your Environment



# Using Ansible Modules to Manage Filesystems in Your Environment

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

## Introduction

## Requirements change, and datacenters evolve. Being able to manage storage configurations on the fly is an important skill. Ansible allows for that using the lvg, lvol, filesystem, and mount modules. This lab will allow you to practice using those modules.

## The Scenario

## Management has purchased a new application, which of course needs database access. Our Database team has requested that a new disk be allocated and configured, to make room for this additional requirement.

Write a playbook that configures /dev/xvdg to have 10G of disk space in a logical volume named AppDB2. Make sure it's formatted using XFS, and mounted at /mnt/appdb2.

## Logging In

## Use the credentials provided on the hands-on lab page to get into Server1 to begin with. Since we need root privileges, let's just run sudo -i right off and become root.

## Set up the New Disk with LVM, and Use the Logical Volume Name AppDB2

## We're going to create disks.yml using whichever editor we like, and put this at the top of the file:

---  
- name: Disks playbook  
  hosts: dbservers  
  become: yes  
  
  tasks:

The first section, the one that will create the LVG, should look like this:

   - name: Create the Volume Group  
     lvg:  
      vg: RHCE  
      pvs: /dev/xvdg  
  
   - name: Create Logical Volume  
     lvol:  
      lv: AppDB2  
      vg: RHCE  
      size: 10G  
      pvs: /dev/xvdg  
      state: present

## Format the Disk Using XFS

## This playbook section will format the disk, and should look similar to this:

   - name: Format the disk  
     filesystem:  
      dev: /dev/RHCE/AppDB2  
      fstype: xfs

## Mount the Disk

## This section of the playbook will mount the disk, and should look like this:

   - name: Mount the disk  
     mount:  
      fstype: xfs  
      src: /dev/RHCE/AppDB2  
      state: mounted  
      path: /mnt/appdb2

## Run the Playbook

## As long as we don't have any typos in our playbook, we should see all of the successful task output march across the screen once we run:

ansible-playbook disks.yml

If we run that command a second time, we'll see that nothing changed, because everything got created and manipulated on the first pass. To double check, log into dbserver1:

ssh ansible@dbserver1

Now check to see if our new partition is in there:

df -h

We should see a line in the output showing /dev/mapper/RHCE-AppDB2 being mounted at /mnt/appdb2.

## Conclusion

## We've finished. We created a playbook that creates a new LVG, formats the disk in it to the XFS filesystem, and then mounts it. Congratulations!

Additional Resources  
Notice: Ansible is installed as the root user, so please work on all tasks after elevating to the root user.  
  
Management has purchased a new application, which of course needs database access. Our Database team has requested that a new disk be allocated and configured, to make room for this additional requirement.  
  
Write a playbook that configures /dev/xvdg to have 10G of disk space in a logical volume named AppDB2. Make sure it's formatted using XFS, and mounted at /mnt/appdb2.  
  
Learning Objectives  
0 of 3 completed  
  
Set up the New Disk with LVM, and Use the Logical Volume Name "AppDB2"  
  
This section of your playbook should look like:  
  
   - name: Create the Volume Group  
     lvg:  
      vg: RHCE  
      pvs: /dev/xvdg  
  
   - name: Create Logical Volume  
     lvol:  
      lv: AppDB2  
      vg: RHCE  
      size: 10G  
      pvs: /dev/xvdg  
      state: present  
Format the Disk Using XFS  
  
This playbook section should look similar to:  
  
   - name: Format the disk  
     filesystem:  
      dev: /dev/RHCE/AppDB2  
      fstype: xfs  
Mount the Disk  
  
This section of the playbook that looks like this:  
  
   - name: Mount the disk  
     mount:  
      fstype: xfs  
      src: /dev/RHCE/AppDB2  
      state: mounted  
      path: /mnt/appdb2

Ansible  Host:

[root@Server1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/usr/bin/hostnamectl set-hostname Server1

/usr/bin/yum install wget git -y

/bin/echo 'source /root/ansible/hacking/env-setup -q' >> /root/.bashrc

/bin/ssh-keygen -N '' -t rsa -f ~/.ssh/id\_rsa

/bin/cat /root/.ssh/id\_rsa.pub >> /home/ansible/.ssh/authorized\_keys

/bin/echo 'python\_interpreter=auto' >> /etc/ansible/ansible.cfg

/bin/echo 'host\_key\_checking = False' >> /etc/ansible/ansible.cfg

/sbin/alternatives —set python /usr/bin/python3

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/privkey> -O /tmp/pkey

/bin/cat /tmp/pkey > /root/.ssh/id\_rsa

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/authkey> -O /tmp/akey

/bin/cat /tmp/akey > /root/.ssh/id\_rsa.pub

/bin/cat /tmp/akey > /home/ansible/.ssh/id\_rsa.pub

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/etchosts> -O /tmp/hosts

/bin/cat /tmp/hosts > /etc/hosts

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/inventory> -O /tmp/inv

/bin/cat /tmp/inv > /etc/ansible/hosts

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

/usr/bin/echo 'cloud\_user:g]I77iE]' | /usr/sbin/chpasswd

/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-2368-ff605f20411b6b8646dcfe75cf8fdc84 —resource PublicInstance —region us-east-1

[root@Server1 cloud\_user]# cat /etc/os-release

NAME="Red Hat Enterprise Linux"

VERSION="8.7 (Ootpa)"

ID="rhel"

ID\_LIKE="fedora"

VERSION\_ID="8.7"

PLATFORM\_ID="platform:el8"

PRETTY\_NAME="Red Hat Enterprise Linux 8.7 (Ootpa)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:8::baseos"

HOME\_URL="<https://www.redhat.com/>“

DOCUMENTATION\_URL="<https://access.redhat.com/documentation/red_hat_enterprise_linux/8/>“

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

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 8"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=8.7

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

REDHAT\_SUPPORT\_PRODUCT\_VERSION="8.7"

[root@Server1 cloud\_user]# ansible —version

ansible 2.10.0.dev0

config file = /etc/ansible/ansible.cfg

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

ansible python module location = /root/ansible/lib/ansible

executable location = /root/ansible/bin/ansible

python version = 3.6.8 (default, Jan 11 2023, 08:43:50) [GCC 8.5.0 20210514 (Red Hat 8.5.0-16)]

[root@Server1 cloud\_user]#

DB SERVER :

[ansible@DBServer1 ~]$ curl 169.254.169.254/latest/user-data

#!/bin/bash

/usr/bin/hostnamectl set-hostname DBServer1

/usr/bin/yum install wget -y

/sbin/useradd ansible

/usr/bin/echo 'ansible:g]I77iE]' | /usr/sbin/chpasswd

/sbin/alternatives —set python /usr/bin/python3

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/etchosts> -O /tmp/hosts

/bin/cat /tmp/hosts > /etc/hosts

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/inventory> -O /tmp/inv

/bin/cat /tmp/inv > /etc/ansible/hosts

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/authkey> -O /tmp/key

/bin/mkdir /home/ansible/.ssh

/usr/bin/chmod 0700 /home/ansible/.ssh/

/bin/cat /tmp/key > /home/ansible/.ssh/authorized\_keys

/usr/bin/chmod 0600 /home/ansible/.ssh/authorized\_keys

/usr/bin/chown ansible.ansible /home/ansible -R

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

/usr/bin/echo 'cloud\_user:g]I77iE]' | /usr/sbin/chpasswd

/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-2368-ff605f20411b6b8646dcfe75cf8fdc84 —resource PublicInstance3 —region us-east-1

[ansible@DBServer1 ~]$ cat /etc/os-release

NAME="Red Hat Enterprise Linux"

VERSION="8.9 (Ootpa)"

ID="rhel"

ID\_LIKE="fedora"

VERSION\_ID="8.9"

PLATFORM\_ID="platform:el8"

PRETTY\_NAME="Red Hat Enterprise Linux 8.9 (Ootpa)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:8::baseos"

HOME\_URL="<https://www.redhat.com/>“

DOCUMENTATION\_URL="<https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8>“

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

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 8"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=8.9

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

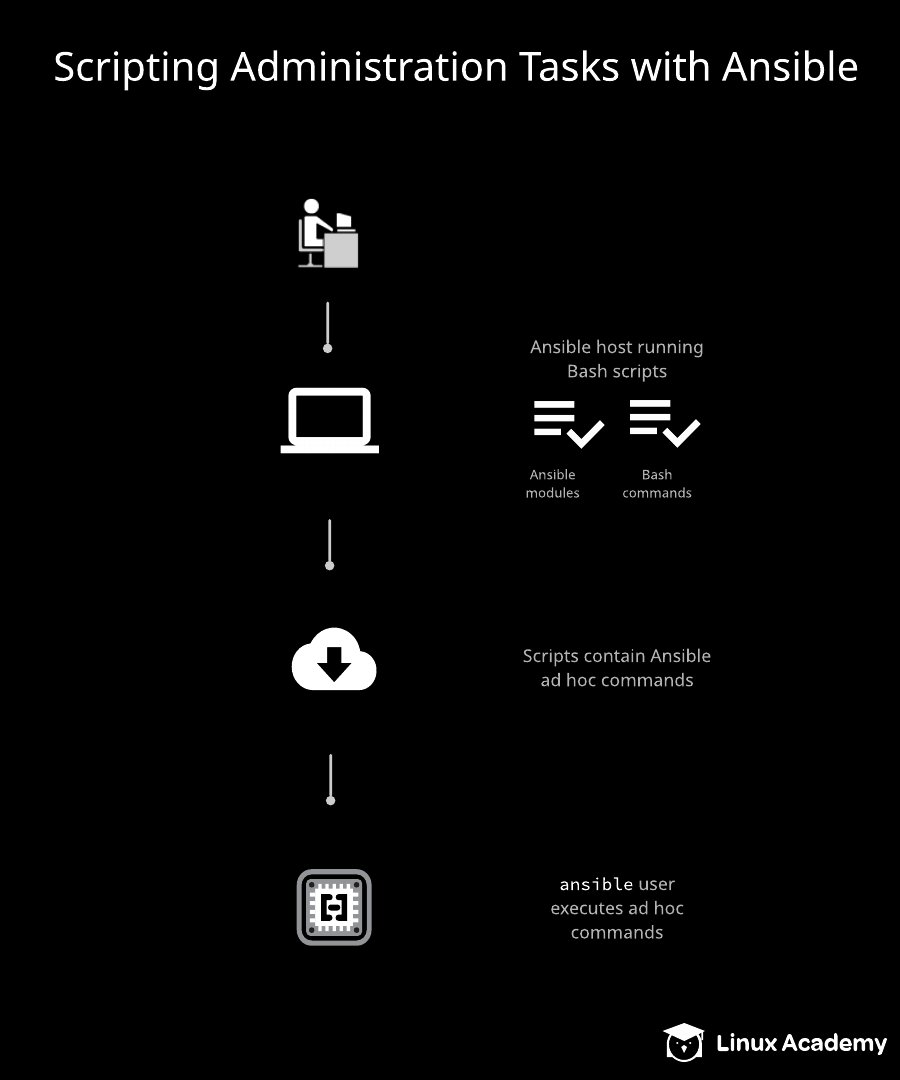
REDHAT\_SUPPORT\_PRODUCT\_VERSION="8.9"

[ansible@DBServer1 ~]$ ansible —version

bash: ansible: command not found...

21

## Scripting Administration Tasks with Ansible



# Scripting Administration Tasks with Ansible

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

## Introduction

## Ansible's ad hoc commands aren't typically used for automation, but they can be powerful tools for running a command across an inventory and getting the desired results. In this lab we'll expand on how to use shell scripting with ad hoc commands to speed up administration tasks.

## Scenario

## One of the tasks that our admin team does periodically is to copy all of /var/log/messages from a remote server and pull it to a local machine for analysis. Rather than doing it manually we'd like to have Ansible do it.

We have to write a script that compresses /var/log/messages with tar on a remote server and then copies that tar file to /tmp/messages/ on the Ansible host.

## Logging In

## Use the credentials provided on the hands-on lab page to get into Server1 to begin with. Since we need root privileges, let's just run sudo -i right off and become root.

## Write the Script

## The script (we'll call it messages.sh could look something like to this:

for i in webserver1 dbserver1 adminserver1;  
   do ssh ansible@$i "sudo tar -czf messages.tar.gz /var/log/messages";  
done  
  
ansible -m fetch -a "src=/home/ansible/messages.tar.gz dest=/tmp/messages" all

But there are thousands of ways to accomplish this with a combination of Bash and Ansible. As long as we write something that archives and compresses the log file, then delivers it to the right directory, we've accomplished the task.

## Execute the Script

## Let's make the script executable by running chmod +x messages.sh. Now run it (./messages.sh) and see what happens.

## Conclusion

## We should see output from our three tar commands, then output from the Ansible side of things (the fetching). That means we've succeeded. Congratulations!

## Additional Resources

Notice: Ansible is installed as the root user, so please work on all tasks after elevating to the root user.

One of the tasks that you and your admin team does periodically is copy all of /var/log/messages from a remote server and pull it to a local machine for analysis. Rather than doing it manually you'd like to have Ansible do it.

Write a script that will use tar to compress /var/log/messages on a remote server and then copy that tar file to /tmp/messages/ on the Ansible Host.

## Learning Objectives

0 of 2 completed

Write the Script

The script could look something like to this:

for i in webserver1 dbserver1 adminserver1;  
   do ssh ansible@$i "sudo tar -czf messages.tar.gz /var/log/messages";  
done  
     
ansible -m fetch -a "src=/home/ansible/messages.tar.gz dest=/tmp/messages" all

But there are thousands of ways to accomplish this with a combination of Bash and Ansible. As long as we write something that archives and compresses the log file, then delivers it to the right directory, we've accomplished the task.

Execute the Script and Ensure There are No Errors

As long as we don't have any errors in the script's output, we should be fine. Good job!

Ansible host :  
  
[root@Server1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/usr/bin/hostnamectl set-hostname Server1

/usr/bin/yum install wget git -y

/bin/echo 'source /root/ansible/hacking/env-setup -q' >> /root/.bashrc

/bin/ssh-keygen -N '' -t rsa -f ~/.ssh/id\_rsa

/bin/cat /root/.ssh/id\_rsa.pub >> /home/ansible/.ssh/authorized\_keys

/bin/echo 'python\_interpreter=auto' >> /etc/ansible/ansible.cfg

/sbin/alternatives —set python /usr/bin/python3

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/privkey> -O /tmp/pkey

/bin/cat /tmp/pkey > /root/.ssh/id\_rsa

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/authkey> -O /tmp/akey

/bin/cat /tmp/akey > /root/.ssh/id\_rsa.pub

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/etchosts> -O /tmp/hosts

/bin/cat /tmp/hosts > /etc/hosts

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/inventory> -O /tmp/inv

/bin/cat /tmp/inv > /etc/ansible/hosts

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

/usr/bin/echo 'cloud\_user:So^6pY0&' | /usr/sbin/chpasswd

/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-2253-6b4cc9251a7c44cc3b221e820fa03bef —resource PublicInstance —region us-east-1

[root@Server1 cloud\_user]# cat /etc/os-release

NAME="Red Hat Enterprise Linux"

VERSION="8.7 (Ootpa)"

ID="rhel"

ID\_LIKE="fedora"

VERSION\_ID="8.7"

PLATFORM\_ID="platform:el8"

PRETTY\_NAME="Red Hat Enterprise Linux 8.7 (Ootpa)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:8::baseos"

HOME\_URL="<https://www.redhat.com/>“

DOCUMENTATION\_URL="<https://access.redhat.com/documentation/red_hat_enterprise_linux/8/>“

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

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 8"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=8.7

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

REDHAT\_SUPPORT\_PRODUCT\_VERSION="8.7"

[root@Server1 cloud\_user]# ansible —version

ansible 2.10.0.dev0

config file = /etc/ansible/ansible.cfg

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

ansible python module location = /root/ansible/lib/ansible

executable location = /root/ansible/bin/ansible

python version = 3.6.8 (default, Jan 11 2023, 08:43:50) [GCC 8.5.0 20210514 (Red Hat 8.5.0-16)]

[root@Server1 cloud\_user]#

[root@Server1 cloud\_user]#

Other servers :

[root@WebServer1 cloud\_user]# curl 169.254.169.254/latest/user-data

#!/bin/bash

/usr/bin/hostnamectl set-hostname WebServer1

/usr/bin/yum install wget -y

/sbin/useradd ansible

/usr/bin/echo 'ansible:So^6pY0&' | /usr/sbin/chpasswd

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/etchosts> -O /tmp/hosts

/bin/cat /tmp/hosts > /etc/hosts

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/inventory> -O /tmp/inv

/bin/cat /tmp/inv > /etc/ansible/hosts

/usr/bin/wget <https://raw.githubusercontent.com/linuxacademy/content-linux-rhcev8/master/authkey> -O /tmp/key

/bin/mkdir /home/ansible/.ssh

/usr/bin/chmod 0700 /home/ansible/.ssh/

/bin/cat /tmp/key > /home/ansible/.ssh/authorized\_keys

/usr/bin/chmod 0600 /home/ansible/.ssh/authorized\_keys

/usr/bin/chown ansible.ansible /home/ansible -R

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

/usr/bin/echo 'cloud\_user:So^6pY0&' | /usr/sbin/chpasswd

/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-2253-6b4cc9251a7c44cc3b221e820fa03bef —resource PublicInstance2 —region us-east-1

[root@WebServer1 cloud\_user]# cat /etc/os-release

NAME="Red Hat Enterprise Linux"

VERSION="8.9 (Ootpa)"

ID="rhel"

ID\_LIKE="fedora"

VERSION\_ID="8.9"

PLATFORM\_ID="platform:el8"

PRETTY\_NAME="Red Hat Enterprise Linux 8.9 (Ootpa)"

ANSI\_COLOR="0;31"

CPE\_NAME="cpe:/o:redhat:enterprise\_linux:8::baseos"

HOME\_URL="<https://www.redhat.com/>“

DOCUMENTATION\_URL="<https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8>“

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

REDHAT\_BUGZILLA\_PRODUCT="Red Hat Enterprise Linux 8"

REDHAT\_BUGZILLA\_PRODUCT\_VERSION=8.9

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

REDHAT\_SUPPORT\_PRODUCT\_VERSION="8.9"

[root@WebServer1 cloud\_user]# ansible —version.

bash: ansible: command not found...

sc