## *Ansible*

Why Ansible?

* Simple
* Powerful
* Agentless

If virtualization technology is not enabled in your laptop/pc, then you can’t create a VM.

To enable virtualization(Windows):

BIOS Settings -> Advanced BIOS Features -> Virtualization Technology[Enabled]

## *Installation Guide:*

### *Change the hostname of controller machine*

Step 1:

| $ sudo vi /etc/hostname |
| --- |

ansiblecontroller

Step 2:

| $ sudo vi /etc/hosts |
| --- |

127.0.0.1 localhost ansiblecontroller

::1 localhost ansiblecontroller

Step 3:

| $ shutdown -r now |
| --- |

Repeat the same steps for target machines.

### *Installing Ansible*

Installation guide: [Installing Ansible](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#control-node-requirements)

Installing Ansible on specific operating systems: [Installing Ansible on specific operating systems](https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html)

Ansible’s community packages are distributed in two ways:

1. A minimalist language and runtime package called ansible-core, and
2. A much larger “batteries included” package called ansible

### *How to build your inventory*

Even if you do not define any groups in your inventory file, Ansible creates two default groups: all and ungrouped. The all group contains every host. The ungrouped group contains all hosts that don’t have another group aside from all.

1. Run Ansible ping test for all target machines from controller machine -

| # ansible all -m ping -i inventory  ---for specific server, run this following command  # ansible lamp-web -m ping -i inventory |
| --- |

1. In which format are Ansible playbooks expressed?

* yaml

1. Can we have multiple plays in a single playbook?

* yes

1. We have a playbook ~/playbooks/copy.yml, which hosts would this playbook run against?

* web1 and web2

Note: Check inventory file under same location

1. There is another playbook ~/playbooks/file.yml Identify how many plays it contains.

Note: Check inventory file under the same location.

1. Under ~/playbooks/ directory create a playbook web1.yml to create a file named as /root/myfile.txt under root’s home on web1 node, also make sure this must run for web1 node only.

| --- - name: Create a myfile.txt  hosts: web1  gather\_facts: no  tasks:  - name: Create myfile.txt  file:  path: /root/myfile.txt  state: touch |
| --- |
|  |

Note: You can use a file module with path and state parameters

1. We have a playbook httpd.yml under ~/playbooks/ directory to install httpd packages on web1 and web2 nodes, try to run the playbook and fix issues if you face any.

| --- - hosts: web  tasks:  - name: Install httpd  yum:  name: httpd  state: present |
| --- |

1. Create a playbook user.yml under ~/playbooks directory to create a user angel on web1 node.

| --- - name: Ansible Create user functionality module demo  hosts: web1  tasks:  - name: Add the user 'angel'  ansible.builtin.user:  name: angel |
| --- |

1. By default ansible…collects facts from managed nodes
2. What module is used by ansible internally to collect facts?

* setup

1. What is the setting in the ansible configuration file that defines the default behavior of gathering facts in ansible?

* gathering

1. Identify the OS distribution\_version of the localhost using ansible facts?

| $ ansible -m setup localhost | grep distribution\_version  # identify the OS distribution of the web1 host using ansible facts  $ ansible -i inventory -m setup web1 | grep distribution |
| --- |

1. A playbook is given at /home/thor/playbooks/db/playbook.yml. We are trying to print the server architecture. However it prints that the variable is not defined.

| --- - hosts: db1  gather\_facts: True  tasks:  - name: Get server architecture  debug:  var: ansible\_facts.architrecture |
| --- |

1. Retrieve all the facts of the web1 server and store them in a file at /home/thor/playbooks/web/web-facts.txt

| $ ansible -m setup web1 -i inventory > /home/thor/playbooks/web/web-facts.txt |
| --- |

1. What is the default location of Ansible configuration file?

| $ /etc/ansible/ansible.cfg |
| --- |

1. Can we override the default Ansible config file?

* yes

1. As per Ansible configuration file precedence, what value of SSH timeout will be used by Ansible while running the playbooks from ~/playbooks directory.

| $ ansible-config view |
| --- |

| *# SSH timeout* timeout = 20 |
| --- |

1. We have playbooks directory inside the user's home, where we have all the playbooks and inventories. We would like to override configuration parameters without making changes inside the default configuration. Make a copy of ansible.cfg file under ~/playbooks directory.

| $ cp -v /etc/ansible/ansible.cfg /home/thor/playbooks/ |
| --- |

1. Change default inventory source file to ~/playbooks/inventory inside the local ansible.cfg file.

| $ vi /home/thor/playbooks/ansible.cfg |
| --- |

| inventory = ~/playbooks/inventory |
| --- |

1. Our hosts are available on port 2222, make sure by default Ansible tries to use port 2222 for SSH connections for all managed hosts instead of the default SSH port.

| $ vi /home/thor/playbooks/ansible.cfg |
| --- |

| remote\_port = 2222 |
| --- |

1. Our hosts are spread worldwide, and it takes normally 30 seconds to establish an SSH connection, make the necessary changes so that Ansible doesn't face SSH timeout issues

| $ vi /home/thor/playbooks/ansible.cfg |
| --- |

| timeout = 30 |
| --- |

1. Disable deprecation warnings for Ansible.

| $ vi /home/thor/playbooks/ansible.cfg |
| --- |

| deprecation\_warnings = False |
| --- |

1. Disable facts gathering for all hosts so that you don't have to explicitly disable facts gathering within the playbooks.

| $ vi /home/thor/playbooks/ansible.cfg |
| --- |

gathering = explicit

1. On Ansible controller node generate an SSH key with filename ansible under default location (~/.ssh).

| $ ssh-keygen -t rsa -f ~/.ssh/ansible |
| --- |

1. We would like to establish password-less secure authentication between Ansible controller and web1 node. Use the keys generated in the previous step and do the needful.

| $ ssh-copy-id -i /home/thor/.ssh/ansible ansible@web1 |
| --- |

1. An inventory file is given at /home/thor/playbooks/inventory. Configure inventory file to use the private ssh key.

| $ web1 ansible\_host=172.20.1.100 ansible\_user=ansible ansible\_ssh\_private\_key\_file=~/.ssh/ansible |
| --- |

1. Run the Ansible command to display the ansible version and save the output in the /tmp/ansible\_version.txt file.

| $ ansible --version > /tmp/ansible\_version.txt |
| --- |

1. Execute an ad-hoc command to perform a ping connectivity test for the localhost and save the output in /tmp/ansible\_ping.txt file

| $ ansible -m ping localhost > /tmp/ansible\_ping.txt |
| --- |

1. Run an ad-hoc commandto perform a ping connectivity to all hosts in the /home/thor/playbooks/inventory file and save the output in /tmp/ansible\_all.txt file

| $ ansible -m ping -i /home/thor/playbooks/inventory all > /tmp/ansible\_all.txt |
| --- |

1. Run an ad-hoc command to run a command on host web1 to print the date and save the output in **/tmp/ansible\_date.txt** file on the ansible controller.

| $ ansible -m command -a date -i inventory web1 > /tmp/ansible\_date.txt |
| --- |

1. Create a shell script called host\_details.sh under ~/playbooks/ directory and make it executable.The shell script should run ad-hoc ansible commands to:

* Print the hostname of all managed nodes in the inventory file ~/playbooks/inventory
* Using the copy module, copy the /etc/resolv.conf file from ansible controller to /tmp/resolv.conf on the node00 host. Use the inventory file ~/playbooks/inventory

###### Step 1: Create host\_details.sh script

| $ vi host\_details.sh |
| --- |

###### Step 2: Add below given code

| ansible all -a "hostname" -i inventory ansible node00 -m copy -a "src=/etc/resolv.conf dest=/tmp/resolv.conf" -i ~/playbooks/inventory |
| --- |

Step 3: Make it executable

| $ chmod +x host\_details.sh |
| --- |

1. Create a shell script called host\_data.sh under ~/playbooks/ directory and make it executable.

The shell script should:

* Set ANSIBLE\_GATHERING to False
* Run an ad-hoc command to print the uptime of all managed nodes in the inventory file ~/playbooks/inventory
* Create and run a playbook ~/playbooks/playbook.yml to cat the file /etc/redhat-release on all managed nodes in the inventory file ~/playbooks/inventory.
* Also please make sure to run playbook in verbose mode i.e just add -vv at the end of your ansible-playbook command.

Step-1: Create playbook.yml

| $ vi playbook.yml |
| --- |

Step-2: Add below given code

| --- - hosts: all  tasks:  - shell: cat /etc/redhat-release |
| --- |

Step-3: Create host\_data.sh script

| $ vi host\_data.sh |
| --- |

Step-4: Add below given code

| export ANSIBLE\_GATHERING=False ansible all -m shell -a uptime -i inventory ansible-playbook -i inventory playbook.yml -vv |
| --- |

Step-5: Make it executable

| $ chmod +x host\_data.sh |
| --- |

1. What is the default value of become\_user directive?

* Root

1. Which of the following can be passed in as an inventory variable to activate privilege escalation?

* ansible\_ become=yes

1. Which of the following options can be used to enable Ansible to ask for sudo password while running an ad hoc command.

* –ask-become-pass

1. We want to change the login shell for a remote user while running an Ansible task, which directive will be used to do so?

* become\_flags

1. We have a playbook file.yml under ~/playbooks/web1 that simply creates a file test.txt on node web1. The user used to connect to the host does not have sufficient privileges to create the file on the desired location but has sudo access. Make the appropriate changes so that the user's privileges are elevated when the playbook is run.

| --- - hosts: web1  become: yes  gather\_facts: no  tasks:  - name: Create a blank file  file:  path: /home/admin/test.txt  state: touch |
| --- |

1. When the file was created on the host, the owner of the file became the root user. However, file was to be created for the admin user. Please make the appropriate changes to the file.yml playbook so that the file is created as user maly

| --- - hosts: web1  become: yes  become\_user: maly  gather\_facts: no  tasks:  - name: Create a blank file  file:  path: /home/admin/test.txt  state: touch |
| --- |

1. There is a playbook file.yml under ~/playbooks/web2/ directory. We want to run file.yml playbook as admin user on web2 node, so modify the playbook accordingly. To run the playbook we have created a script web2.sh on the same location, so you can execute the script with command sh web2.sh. We don’t want to store the sudo password in any file due to security reasons. Make the necessary changes so that when the script is run, the playbook must prompt for the become password

Step-1: Update the file.yml playbook as given below

| --- - hosts: all  gather\_facts: no  become: true  become\_user: admin  tasks:  - name: Create a blank file  file:  path: /home/admin/test.txt  state: touch |
| --- |

Step-2: Edit the web2.sh script and add --ask-become-pass

| $ vi web2.sh |
| --- |

###### 

| #!/bin/bash ansible-playbook -i inventory file.yml --ask-become-pass |
| --- |

1. We need privilege escalation to be enabled across all playbooks without having to specify in each play, make the necessary changes in /etc/ansible/ansible.cfg file to activate privilege escalation.

| $ sudo vi /etc/ansible/ansible.cfg |
| --- |

| [privilege\_escalation] become=True |
| --- |

1. Our organization recently introduced changes in security. Going forward we'd like to use **doas** as a privilege escalation tool for all managed nodes without having to update inventories or passing in command line parameters for each node. Make the necessary changes.

###### 

| $ sudo vi /etc/ansible/ansible.cfg |
| --- |

###### 

| [privilege\_escalation] become=True become\_method=doas |
| --- |

1. Let us explore the environment for our KodeKloud e-commerce LAMP stack application. There are 2 servers - **lamp-web** and **lamp-db**. Let us set up the inventory files for that. Create an inventory file at /home/thor/playbooks/lamp-stack-playbooks/inventory to include the following data:

**Hosts:** lamp-web, lamp-db

**Groups:** db\_servers contains lamp-db; web\_servers contains lamp-web

**IP Addresses:** lamp-web: 172.20.1.100; lamp-db: 172.20.1.101

Credentials for lamp-web: **Username**=john **Password**=john

Credentials for lamp-db: Username=maria Password=maria

| [db\_servers] lamp-db ansible\_host=172.20.1.101 ansible\_ssh\_pass=maria ansible\_user=maria  [web\_servers] lamp-web ansible\_host=172.20.1.100 ansible\_ssh\_pass=john ansible\_user=john |
| --- |

1. Let's add some additional data required for setting up the database and web servers. The data should be associated with the respective servers.

**Database Info:**

mysqlservice=mysqld

mysql\_port=3306

dbname=ecomdb

dbuser=ecomuser

dbpassword=ecompassword

**Web Info:**

httpd\_port=80

repository=<https://github.com/kodekloudhub/learning-app-ecommerce.git>

| [db\_servers] lamp-db ansible\_host=172.20.1.101 ansible\_user=maria mysqlservice=mysqld mysql\_port=3306 dbname=ecomdb dbuser=ecomuser dbpassword=ecompassword  [web\_servers] lamp-web ansible\_host=172.20.1.100 ansible\_user=john httpd\_port=80 repository=https://github.com/kodekloudhub/learning-app-ecommerce.git |
| --- |

1. Update the inventory file to use private keys for the respective hosts and remove password

| [db\_servers] lamp-db ansible\_host=172.20.1.101 ansible\_ssh\_private\_key\_file=/home/thor/.ssh/maria ansible\_user=maria mysqlservice=mysqld mysql\_port=3306 dbname=ecomdb dbuser=ecomuser dbpassword=ecompassword  [web\_servers] lamp-web ansible\_host=172.20.1.100 ansible\_ssh\_private\_key\_file=/home/thor/.ssh/john ansible\_user=john httpd\_port=80 repository=https://github.com/kodekloudhub/learning-app-ecommerce.git |
| --- |

1. A playbook **deploy-lamp-stack.yml** is given with a basic task to install basic libraries

| --- - name: Deploy lamp stack application  hosts: localhost  become: yes  tasks:  - name: Install common dependencies  yum:  name:  - libselinux-python  - libsemanage-python  - firewalld  state: installed |
| --- |

1. We have a requirement on web1 node to white list web2 node's IP address 172.20.1.101 in the firewall. Create and run a playbook ~/playbooks/whitelist.yml to do so.

| --- - hosts: web1  tasks:  - firewalld:  source: 172.20.1.101  state: enabled  zone: internal  permanent: yes  immediate: yes |
| --- |

1. A We want to block the 161/udp port on the web1 node permanently. Make a playbook block.yml under ~/playbooks/ directory to do so.

| --- - hosts: web1  tasks:  - firewalld:  port: 161/udp  permanent: yes  immediate: yes  zone: block  state: enabled |
| --- |

1. A On web1 node adds a firewall rule in the internal zone to enable https connection from Ansible controller machine and make sure that rule must persist even after system reboot. You can create a playbook https.yml under ~/playbooks/ directory.

The IP address of ansible controller is 172.20.1.2.

| --- - name: Add Firewalld  hosts: web1  tasks:  - firewalld:  source: 172.20.1.2  service: https  zone: internal  state: enabled  permanent: yes   - service:  name: firewalld  state: reloaded |
| --- |

1. We have a playbook ~/playbooks/web2-config.yml, it has some existing code to change apache’s default port 80 to port 8082 as we want to run Apache on port 8082 on web2 node. Make some changes as given below before running the playbook.

* Add an entry in ~/playbooks/inventory for web2 node, IP address of web2 node is 172.20.1.101 and ssh password and username are same as of web1 (username = root and password = Passw0rd).
* Update web2-config.yml to install httpd before updating its port in config, also start/enable its service.
* Install firewalld package and start/enable its service.
* As now Apache will listen on port 8082 so edit the playbook to add firewall rules in the public zone so that Apache can allow all incoming traffic.

| --- - hosts: web2  tasks:  - name: Install packages  yum:  name: httpd, firewalld  state: present   - name: Start/Enable services  service:  name: "{{ item }}"  state: started  enabled: yes  with\_items:  - httpd  - firewalld   - name: Change Apache port  replace:  path: /etc/httpd/conf/httpd.conf  regexp: "Listen 80"  replace: "Listen 8082"   - name: restart Apache  service:  name: httpd  state: restarted   - name: Add firewall rule for Apache  firewalld:  port: 8082/tcp  zone: public  permanent: yes  state: enabled  immediate: true |
| --- |

| # add this to inventory file  web2 ansible\_host=172.20.1.101 ansible\_ssh\_pass=Passw0rd ansible\_user=root |
| --- |

1. Create a playbook ~/playbooks/perm.yml to create file /opt/data/perm.txt with 0640 permissions on web1 node

| --- - name: Create Blank File  hosts: web1  tasks:  - name: Create blank text file  file:  path: /opt/data/perm.txt  state: touch  mode: '0640' |
| --- |

1. Using a playbook ~/playbooks/index1.yml create /var/www/html/index.html file on web1 node with content “This line was added using Ansible lineinfile module!”

| --- - hosts: web1  tasks:  - name: Create html file  lineinfile:  path: /var/www/html/index.html  line: 'This line was added using Ansible lineinfile module!'  create: yes |
| --- |

1. We have a playbook ~/playbooks/find.yml that recursively finds files in /opt/data directory older than 2 minutes and equal or greater than 1 megabyte in size. It also copies those files under /opt directory. However it has some missing parameters so it's not working as expected, take a look into it and make appropriate changes.

| --- - hosts: web1  tasks:  - name: Find files  find:  paths: /opt/data  age: 2m  size: 1m  recurse: yes  register: file   - name: Copy files  command: "cp {{ item.path }} /opt"  with\_items: "{{ file.files }}" |
| --- |

1. In /var/www/html/index.html file on web1 node add some additional content using blockinfile module. Below is the content:

Welcome to Kodacloud!

This is Ansible Lab.

Make sure the user owner and group owner of the file is apache, also make sure the block is added at the beginning of the file. Create a new playbook for this ~/playbooks/index2.yml

| --- - hosts: web1  tasks:  - name: Create html file  blockinfile:  path: /var/www/html/index.html  owner: apache  group: apache  insertbefore: BOF  block: |  Welcome to KodeKloud!  This is Ansible Lab. |
| --- |

54. n /var/www/html/index.html file on web1 node adds some additional content using the blockinfile module. Below is the content: