Day 1

==============================================

Configuration Management

===========================

This is the process of handling activities of multiple

servers from one point of control

Advantages

==============

1 Provisioning of Servers

Activities like installing s/w applications, deleting s/w, upgrading

configuring patches can be done very quickly

2 Resource Saving

The time taken to configure any number of servers become very less

and the number of people who are required for these server configurations

also becomes less

3 Useful in Disaster Recovery

To handle disaster recovery organizations maintain replica data centers

at different geographical locations. Creation of these replica data centers

can be done very easily

4 Handling snowflake servers

After a point of time all servers present in a data center behave like

snowflakes i.e they run on slightly different h/w and s/w configurations.

Configuration Management tools can pick up this info and store in simple

config files which can be used later to setup similar environments

5 Idempotent

Configuration management tools are used to bring the remote to a state

called as the "desired state". If the remote servers are already in the

desired state configuration management tools will not reconfigure these

server

Important Configuration Management Tools

=============================================

1 Ansible

2 Chef

3 Puppet

Ansible

=============

This is an open source configuration management tool created using python

The main machine where ansible is installed is called as "Controller"

and the remaining remote servers that we are configuring are called as

"managed nodes/hosts"

From the controller to the managed nodes we should have passwordless

shh connectivity

Ansible is called as "agentless" ie we need not install any client

s/w of ansible on the remote managed nodes.It uses "push" methodolgy

to push the configurations into the remote servers.

=========================================================================

Setup of Ansible

====================

1 Create 3 or 4 AWS ubuntu 18 instances

2 NAme the 1st one as controller and remaining 2 as server1 and server2

3 Establish Passwordless ssh from Controller to Server1 and Server2

a) Connect to server1 using gitbash

b) Setup password for the default user

sudo passwd ubuntu

c) Edit the ssh configuration file

sudo vim /etc/ssh/sshd\_config

Search for "PasswordAuthentication" and change it from no to yes

d) Restart ssh

sudo service ssh restart

Repeat the above steps from a to d on Server2 managed node

e) Connect to Controller using git bash

f) Generate the ssh keys

ssh-keygen

g) Copy the ssh keys

ssh-copy-id ubuntu@private\_ip\_of\_server1

Repeat step g with ipaddress of Server2

4 Installing Ansible

a) Update the apt repository

sudo apt-get update

b) Install software-properties-common

sudo apt-get install -y software-properties-common

c) Add the latest version of Ansible to apt repository

sudo apt-add-repository ppa:ansible/ansible

d) Update the apt repository

sudo apt-get update

e) Install ansible

sudo apt-get install -y ansible

5 To check the verision of ansible

ansible --version

Ansible stores all the remote servers info in a file called as inventory file

We should open this file and store the ipaddress of all the managed nodes here

sudo vim /etc/ansible/hosts

Here copy and paste the ipaddresses of the managed nodes

===========================================================================

Day 2

==============================================================================

Ansible performs remote configuration of servers in

3 different ways

1 Adhoc commands

2 Playbooks

3 Roles

Ansible uses prebuild Python modules for configuring remote

servers

Important modules in Ansible

===================================

1 command: This is used to execute linux commands on the remote managed

nodes. It is the default module of Ansible

2 shell: This is used to execute shell scripts on the remote managed nodes.

It can execute command related to redirection and piping

3 user: This is used to perform user administration on the remote servers

like creating users, assigning home dirs., deleting users etc

4 file: Used for creating files/directories on the managed nodes

5 copy: This used to copy files/directories to the managed node

6 fetch: Used to copy files/directories from managed nodes to controller

7 apt: Used for s/w package management like installing, deleting, upgrading

etc. It works on Ubuntu, Debain flavours of linux

8 yum: This is similar to apt but it works on Rehat linux, Centos, Fedora etc

flavours of Linux

9 service: Used to start stop or restart services on the managed nodes

10 uri: Used to check if a remote url is reachable or not

11 git: Used for performing git version controlling on the managed nodes

12 get\_url: Used for downloading files from remote servers into the managed nodes

13 stat: Used to capture detailed info about files/directories on the managed nodes

14 debug: Used to display the output in JSON file format

15 include: Used to call child playbooks from a parent playbook

16 replace: Used to replace specific portions of the text in a file

17 docker\_container: Used for container management on the managed nodes

18 docker\_image: Used to run command related to docker images

19 docker\_login: Used to login into the docker registry

20 docker\_swarm: Used to setup of docker swarm architecture

====================================================================================

Adhoc command Syntax

==========================

ansible all/group\_name/ipaddress -i path\_of\_inventory -m module\_name -a 'arguments'

CommandModule

====================

Ansible command to see the memory info of all managed nodes

ansible all -i /etc/ansible/hosts -m command -a 'free -m'

Note: /etc/ansible/hosts is the default inventory file and when working on it

we need not specify the -i option

ansible all -m command -a 'free -m'

Note: command module is the default module of Ansible and when working on it

we need not specify the -m option

ansible all -a 'free -m'

===============================================================================

Shell Module

Ansible command to install docker on all managed nodes

ansible all -m shell -a 'curl -fsSL https://get.docker.com -o get-docker.sh'

ansible all -m shell -a 'sh get-docker.sh'

Ansible command to store the memory info of all managed nodes in file1

ansible all -m shell -a 'free -m > file1'

==============================================================================

UserModule

Ansible command to create a user and assign a password

ansible all -m user -a 'name=sai password=intelliqit' -b

to check user actually created or not run following command

cat /etc/passwd

Note: -b represents "become" it is used to giving higher privileges on the

remote managed nodes

User module can also assign home dirs, default working shell, uid etc

ansible all -m user -a 'name=Anu password=intelliqit uid=1234

home=/home/ubuntu/Anu shell=/bin/bash comment="A normal user"' -b

============================================================================

file module

=================

Ansible command to create a file on all managed nodes

ansible all -m file -a 'name=/tmp/file14 state=touch'

Note: state= touch is for creating files

state=directory is for creating directories

state=absent is for deleting file/directories

Ansible command to create a file and also change the permissions

ownership and groupship

ansible all -m file -a 'name=/home/ubuntu/file56 state=touch

owner=sai group=Anu mode=770' -b

=================================================================================

Copy Module

============

Ansible command to copy a file from controller to all managed nodes

ansible all -m copy -a 'src=file100 dest=/tmp'

Ansible command to copy a file and also change permissions ownership and group ownership

ansible all -m copy -a 'src=file100 dest=/tmp owner=root group=sai mode=764' -b

Copy module can also replace the existing content of a file

ansible all -m copy -a 'content="Hello IntelliQ\n" dest=file1'