Day 1

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

Configuration Management

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

This is the process of handling activites of multiple

servers from one point of control

Advantages

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

1 Provisioning of Servers

Activites 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 Usefull 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 ie 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 managment 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 remianing 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