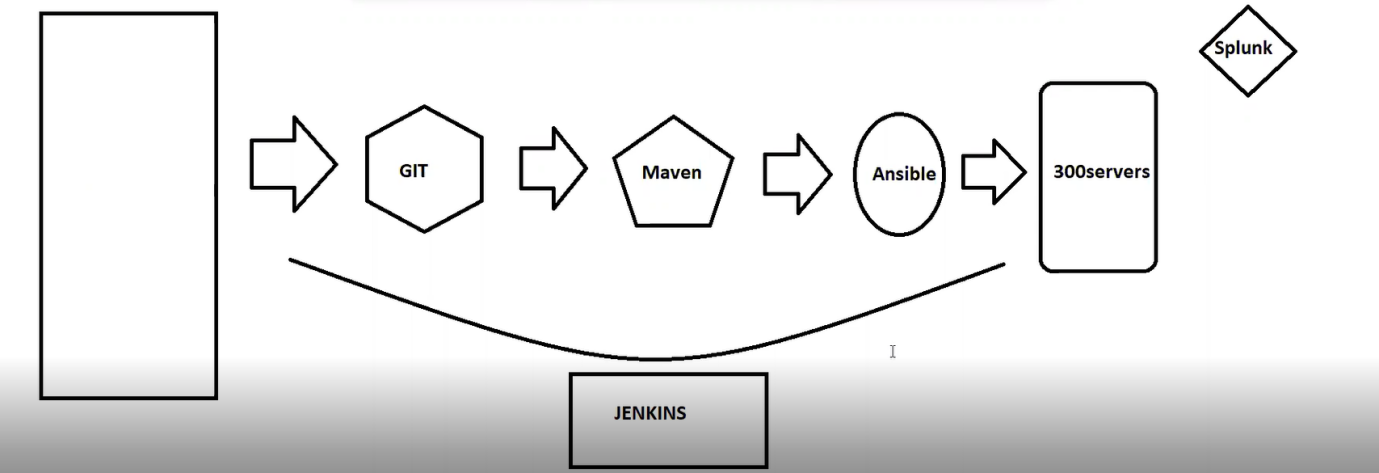
****

**What is Configuration Management Tool?**

**We have Multiple Environment in real-time, it is really hard to maintain everything manually; but by using configuration tool we can automate the complete process use in specific application.**

**In simple words , if I have Multiple server (20+), client is asking me to install Apache ,java and to create a group , user and directories , ( Scenario without configuration tool : we need to do it one by one server and this consume our time and effort , but by using configuration tool we can automate it just by from one server to another.**

**Ansible: Configuration Management Tool**

**Ansible it is Procedural**

**Terraform it is Declarative.**

**Master & Slave :**

**MASTER: Control Node**

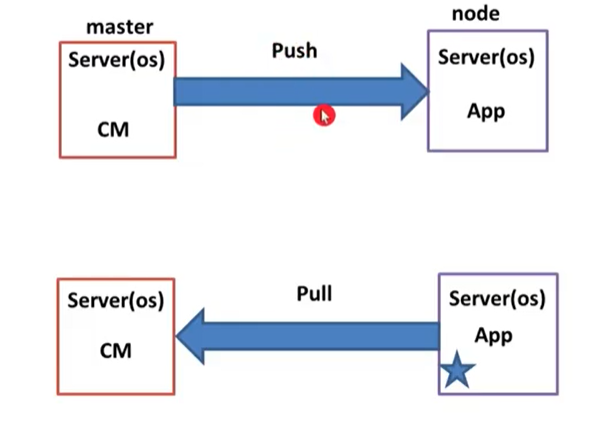
**Slave : Manage Node**

**Here Master Machine its not support windows OS but slave support**

**Ansible, salt stack we don’t need learn scripting Languages.**

**Chef, Puppet we need to learn Ruby , Groovy languages.**

**In Ansible if any requirement we need can easily refer the officially documents**



**Basic Knowledge about Configuration Management Tool:**

**Agent Base ( if CM tool will be installed both master and slave )**

**Agent less ( CM tool will be installed only in master alone)**

**==================================================================================**

**Push & Pull Mechanism**

**Push : ( here we just give all slave details in Master only, then it will push and don’t need to give details in slave)**

**Pull : (here we update slave details in master , at the same time we need to login slave machine and update the master details)**

**Ansible ------> Push Mechanism**

**Salt -----> Push & Pull mechanism**

**Chef & Puppet 🡪 Pull Mechanism**

**==================================================================================**

**Ansible it is red hut Product .**

**Pre-requesting : it is written in python.**

**==================================================================================**

**Creating 4 server:**

**Master -1**

**Slaves -3**

**We should install python for all server**

**Then Login master machine and install ansible**

**Then give slave information in master.**

**==================================================================================**

**Putty, Puttygen, Pageant**

**We attached all key pair in Pageant ( we should give PPK).**

**==================================================================================Step-1**

**Create 4 instance (Amazon-Linux)**

**While creating install 4 server. we add user data for python install.**

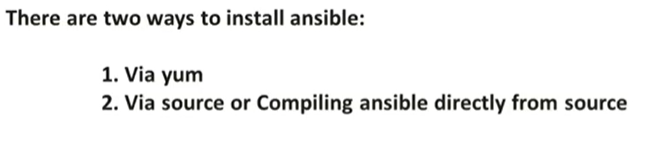
**#! /bin/bash**

**yum install python-pip -y**

**step-2**

**add key to pageant.**

**==================================================================================**



**PIP – Python installation Package.( It is manual Method)**

**Two Main file for Ansible.(important)**

1. **Inventory file**
2. **Configuration file**

**If we use yum install or apt-get install ansible. Inventory and configuration file automatically come. Don’t we need to give. ( It is easy)**

**But if we use pip install for ansible. Both files haven’t come. because which means manual method.**

**Best for our demo we are going to use pip.**

**Inventory file -> we give all slaves information Manually.**

**Configuration file 🡪 ( all ansible behaviours details avail here)**

**==================================================================================**

**Login Master Machine**

**Step-2:**

**# sudo yum update**

**Don’t go to root user because Inventory and configuration file only should avail in home page.**

**#pip install ansible ( instead of yum install ansible)**

**How to take configuration file:**

**Step-3:**

**Go and search ansible configuration file GitHub.**

**<https://github.com/ansible/ansible/blob/stable-2.7/examples/ansible.cfg>**

**click 2.7 release and click Raw (<https://raw.githubusercontent.com/ansible/ansible/stable-2.7/examples/ansible.cfg>)**

**wget url**

**==================================================================================**

**Step-4:**

**Update inventory file**

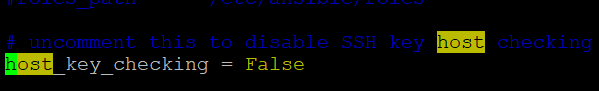
**#vi slaves.txt ( any name we can give but should save .txt format)**

**Here update all slaves private ip. ( static)**

**=================================================================================**

**Step-5:**

**#vi ansible.cfg 🡺 scroll down literally 🡺 search host\_key\_checking (uncommand)**



**Why we uncommand it ask procced commands .**

**If we are using yum , apt-get we update inventory update below path**

**yum => /etc/ansible/hosts/ ( inventory file)**

**/etc/ansible/ansible.cfg ( configuration file)**

**=================================================================================**

**Adhoc commands: ( one task at one time) ( single task at a time)**

**# ansible all -i slaves.txt -m ping ( if yum install we don’t need to give (-i slaves.txt)**

**Slaves.txt : ( inventory file name)**

**-m : module**

**# ansible all -i slaves.txt -a "uname -a" (uname which is show server os/hostname details)**

**-a : action/argument**

**# ansible all -i slaves.txt -a “uptime”**

**-m don’t need to give double quote**

**-a we need to give double quote**

**We need to know separate machine:**

**#ansible <ip address> -I slaves.txt -a “uptime”**

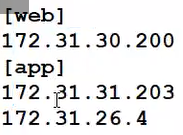
**==================================================================================**

**Grouping:**

**Go to slaves.txt**

**Add name for slave’s machines**

**For an Example:**



**# ansible web -i slaves.txt -m ping**

**# ansible app -i slaves.txt -m ping**

**Same if we don’t add to grouping for a particular slave . we must mention ip address at above grouping in inventory file (slaves.txt)**

**==================================================================================**

**-m = module**

**-a = arguments/actions**

**For an example:**

**Yum 🡪 install , update , remove, modify**

**Service/system ctl 🡪 start , stop, enable , restart, shutdown,**

**Wget => url**

**Copy- 🡺 source , destination**

**==================================================================================**

**States:**

**Install => present/latest**

**Uninstall => absent**

**Start 🡺 started**

**Stop => stopped**

**Restart 🡺 restarted**

**==================================================================================**

**Install httpd**

**# ansible app -i slaves.txt -m yum -a “name=httpd state=present”**

**We will be getting error.**

**# ansible app -i slaves.txt -m yum -a “name=httpd state=present” -b**

**B: become root user**

**# ansible app -i slaves.txt -m service -a “name=httpd state=started” -b**

**# ansible all -i slaves.txt -m copy -a "src=/home/ec2-user/index.html dest=/var/www/html/index.html" -b**

**Then create the index.html file in master machine.**

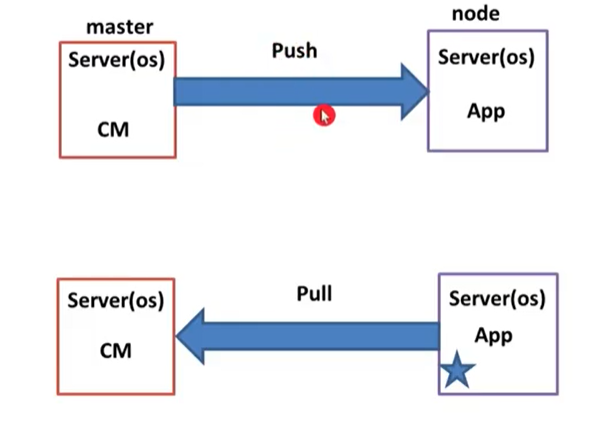
**#ls**

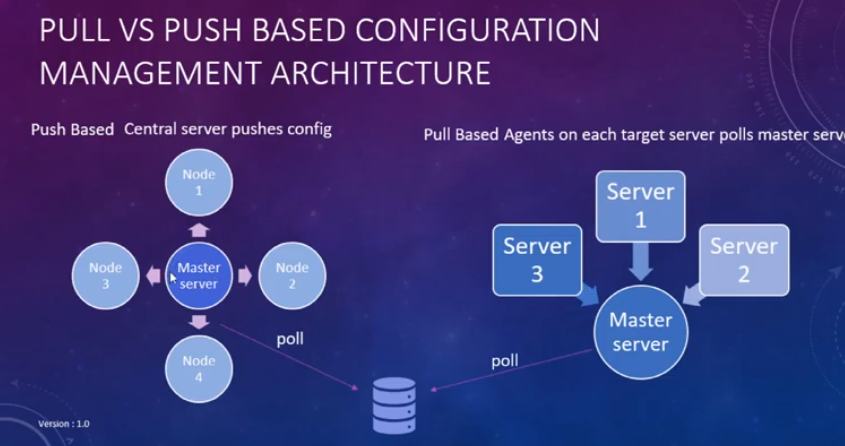
**#ansible all -i salves.txt -m copy -a “ src=./slaves.txt dest=/tmp/slaves.txt”**

**#ssh ec2-user@<ipaddress>**

**#cd /tmp**

**#ls**





**==================================================================================**