Introduction to Ansible

Story before Ansible or Any other configuration Management tools

Simple Task

- ✓ Assume that we have 100 servers.
- ✓ Need to Install vim and wget pkgs on all servers.
- ✓ First, we will go with one server.
    - ▪ Step1: Login to each server
    - ▪ Step2: Check vim and wget pkgs installed or not.
    - ▪ Step3: If not installed then install using:
        - ➢ **yum install wget -y**
        - ➢ **yum install vim -y**
- ✓ Assume that this process took 6 min
- ✓ Then for 100 servers: 600 min , which is equal to 10 hrs.

Simple Task implementation with Shell Script

✓ **Assume that we are good with shell script.**

```
[root@ip-172-31-93-110 ~]# cat install.sh
#!/bin/bash
vim --version 1>/dev/null 2>&1
if [[ $? -eq 0 ]]
then
    echo "Alreday vim instlled"
else
    sudo yum install vim -y
fi

wget  --version 1>/dev/null 2>&1
if [[ $? -eq 0 ]]
then
    echo "Alreday wget instlled"
else
    sudo yum install wget -y
fi
[root@ip-172-31-93-110 ~]# cat copy_install.sh
#!/bin/bash
cnt=1
for each_server in $(cat list_of_servers.txt)
do
  echo "$cnt. Working on ${each_server}"
  scp install.sh root@${each_server}:/tmp
  ssh root@${each_server} "sh /tmp/install.sh"
done
```

✓ **Assume that shell scripting is taking 3 min of time to complete our task per server.**

✓ **So total time required is: 300 min, which is equal to 5hr.**

Simple Task implementation with Any Configuration management tools

✓ If we go with any automation tools like ansible, chef, puppet, salt or any other then we can complete this simple task in 3 min (which is required per server) for all 100 server.

✓ Ansible will execute a task on all servers parallelly. So it will take only 3 min of time for any number of server for our requirement.

✓ Generally writing shell script is complex compare to playbooks and playbooks are very short in code length.

```yaml
---
 - hosts: all
   tasks:
   - yum:
       name: wget
       state: present
   - yum:
       name: vim
       state: present
```

✓ **Ansible is an open source Automation tool.**

✓ **It is very, very simple to setup and yet powerful.**

✓ **Ansible will be helpful to perform:**

      ✓ **Configuration Management**

      ✓ **Application Deployment**

      ✓ **Task Automation**

      ✓ **and also IT orchestration**

Thank you

Ansible How Ansible Works ?

- ✓ **Ansible works by connecting to remote nodes and pushing out small programs, called "Ansible modules" to them.**
- ✓ **The pushed programs/modules will be executed on remote server by Ansible over SSH and removes them when finished.**
- ✓ **Unlike Puppet or Chef it doesn't use an agent on the remote host, Instead Ansible uses SSH.**
- ✓ **it's written in Python which needs to be installed on the remote host.**
- ✓ **This means that you don't have to setup a client server environment before using Ansible.**