**Ansible**

**Ansible on linux -**

Ansible is a configuration management tool. It uses SSH to connect to servers and run the configured Tasks on linux machine.

**Hosts file**

Once you've installed Ansible, the first thing is to understand its inventory file. This files contains the list of target servers which are managed by Ansible. The default hosts file location is /etc/ansible/hosts. We can edit this file to include our target systems. This file specifies several groups in which you can classify your hosts as your preference.

**How to work with Ansible**

First of all, Ansible admin client connects to the target server using SSH. We don't need to setup any agents on the client servers. All you need is Python and a user that can login and execute the scripts.

**Establishing SSH connections**

(<http://linoxide.com/linux-how-to/started-ansible-command-line/> - link to refer)

Steps to follow –

1. Generate the key.
2. Copy to all client machine

ssh-copy-id user\_name@ip\_Address

(seperatly need to copy to different server)

1. To confirm the connectivity use

ansible -i hosts all -m ping -u root

**Modules**

(<http://docs.ansible.com/ansible/list_of_all_modules.html> - link for reference)

usercreation –

ansible -i hosts all -m user -a "name=daksha password=daksha" -u root

file –

 ansible -i hosts all -m copy -a "src=test.sh dest=/root/" -u root

**Playbook**

To run the playbook

“ansible-playbook –i hosts playbook.yml”

Examples –

1. ping –

---

- hosts: all

tasks:

- name: ping all host

ping:

1. Script –

---

- hosts: all

tasks:

- name: Run the shell script

script: /home/daksha/script.sh >> test1.txt

script file –

getent passwd

1. Include –

Main playbook

- include: playbook1.yml

- include: playbook2.yml

Playbook 1

---

- hosts: all

tasks:

- name: create a file

file: path=/home/daksha/newfile1 state=touch

Playbook 2

---

- hosts: all

tasks:

- name: create file

file: path=/home/daksha/newfile2 state=touch

1. Roles –

For the roles one mail yml file will be there   
for eg.

---

- hosts: \_Development\_

gather\_facts: false

roles:

- precheck

- java

- tomcat

- eclipse

The main.yml file will be there in roles/rolename/task/main.yml

Since we are using roles in windows.yml, create a folder roles and inside which the folder structure as described under roles.

For example we are running precheck here, in roles🡺precheck🡺tasks, create a main.yml file(in this .yml file give commands or modules you want to execute)

In each main.yml file it have code related to task.

To run this – “**ansible-playbook -i hosts windows.yml”**

**Ansible on Windows –**

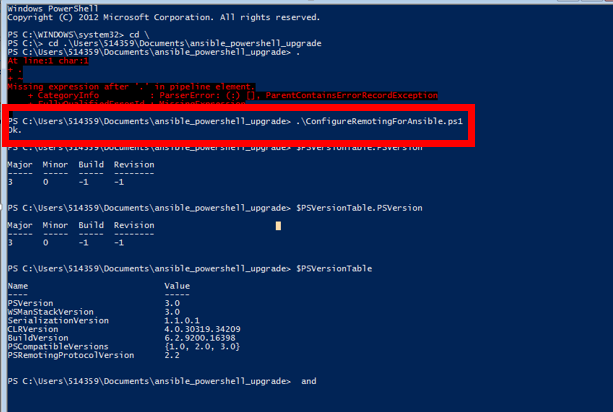
To connect windows to ansible-:( http://docs.ansible.com/ansible/intro\_windows.html)

1. To connect to windows machine using linux first do these steps in linux server

* sudo --proxy=http://Scloudsetadmin:C111111%23@proxy.cognizant.com:6050 apt-get install --reinstall python2.7
* sudo apt-get purge python-pip
* wget https://raw.github.com/pypa/pip/master/contrib/get-pip.py
* sudo python get-pip.py
* sudo pip install

1. In order for Ansible to manage your windows machines, you will have to enable and configure PowerShell remoting.
2. Check version of powershell using cmd, $PSVersionTable and $PSVersionTable.PSVersion
3. Run the script([upgrade\_to\_ps3.ps1](https://github.com/ansible/ansible/blob/devel/examples/scripts/upgrade_to_ps3.ps1" \o "upgrade_to_ps3.ps1)) from dis link <https://github.com/ansible/ansible/tree/devel/examples/scripts.>
4. Restart the pc.
5. Run Enable-PSRemoting.
6. Run the script([ConfigureRemotingForAnsible.ps1](https://github.com/ansible/ansible/blob/devel/examples/scripts/ConfigureRemotingForAnsible.ps1" \o "ConfigureRemotingForAnsible.ps1)) from dis link <https://github.com/ansible/ansible/tree/devel/examples/scripts>.

Save these two scripts in your local and run it.



1. In linux machine, in hosts file, append the following lines to configure with windows machine.(give the windows machine ip u want to connect to)

[\_Development\_]

10.219.193.101

[\_Development\_:vars]

ansible\_ssh\_user=’User\_name

ansible\_ssh\_pass=’Password’

ansible\_ssh\_port=5986

ansible\_connection=winrm

ansible\_winrm\_transport=ntlm

ansible\_winrm\_server\_cert\_validation=ignore

Example –

Install eclipse

Windows\_installation.yml

---

- hosts: \_Development\_

gather\_facts: false

roles:

- java\_install

- eclipse\_install

- tomcat\_install

Roles/eclipse/tasks/main.yml

- name: unzip eclipse

win\_unzip:

src: D:\Softwares\test\eclipse-jee-indigo-SR2-win32.zip

dest: D:\Softwares\test

- name: create workspace

raw: D:\Softwares\test\eclipse.exe -Data D:\Softwares\test\eclips\_projects

- name: create shortcut

script: script/createshortcut.ps1

create shortcut.ps1

$ws= New-Object -ComObject WScript.Shell;

$s = $ws.CreateShortcut('C:\Users\514740\Desktop\eclipse.lnk');

$S.TargetPath = 'D:\Softwares\eclipse\eclipse.exe'

$S.Save()

Roles/install\_notepad++/tasks/main.yml

* Name: install\_notepad++

Script: script/install\_notepad++

install\_notepad++.ps1

$msiFile = "D:\Softwares\npp.6.6.6.Installer.exe"

if (!(Test-Path $msiFile)){

throw "Path to the MSI File $($msiFile) is invalid. Please supply a valid MSI file"

}

$InstallDir = "<C:\Program Files\Notepad++>"

$arguments = @(

"/S"

"/D=$INSTALLDIR"

)

Write-Host "Installing $msiFile....."

Write-Host $arguments

$process = Start-Process -verb RunAs -FilePath $msiFile -ArgumentList $arguments -PassThru -Wait

if ($process.ExitCode -eq 0){

Write-Host "$msiFile has been successfully installed"

}

else {

Write-Host "Installer exit code $($process.ExitCode) for file $($msifile)"

}

Exit $($process.ExitCode)