**Connection Establishment of client server**

Just type ssh user@hostname to connect to the machine.

**Establish the connection with ssh keypair –(passwordless connection)**

1)generate the key on server machine

ssh-keygen -t rsa

2) Use SSH from server **192.168.0.12** to connect server **192.168.0.11** using **sheena** as user and create **.ssh**directory under it, using following command.

**ssh sheena@192.168.0.11 mkdir -p .ssh**

**3)** Use SSH from server **192.168.0.12** and upload new generated public key (**id\_rsa.pub**) on server **192.168.0.11**under **sheena**‘s **.ssh** directory as a file name **authorized\_keys**.

**cat .ssh/id\_rsa.pub | ssh sheena@192.168.0.11 'cat >> .ssh/authorized\_keys'**

or you can manually copy it to authorized key file of clent machine

4) From now onwards you can log into **192.168.0.11** as **sheena** user from server **192.168.0.12** as **tecmint** user without password.

**ssh** [**sheena@192.168.0.11**](mailto:sheena@192.168.0.11)

**change username – usermode –l newuser olduser**

**change password – passwd password**

**list of user – getent passwd**

**change hostname – hostname newname**

**delete user -** userdel username

adduser – adduser username

**Ansible-:**

1. **Inventory file-: make hosts file in which add the server ip u want to connect to.**

* By default, the inventory file is expected to be /etc/ansible/hosts.
* Add an entry to your hosts file, pointing to a server that you connected to. You can include multiple servers in this file, using either domains or IP addresses, and can even group them.
* Use the all directive to ping all servers in your hosts file via Ansible:

**ansible all –m ping**

1. **Establish the connection with ssh keypair –(passwordless connection)**

**(**[**http://linoxide.com/linux-how-to/started-ansible-command-line/**](http://linoxide.com/linux-how-to/started-ansible-command-line/) **- link to refer)**

* Generate the key.
* Copy to all client machine
* ssh-copy-id [root@139.162.35.39](mailto:root@139.162.35.39) seperatly need to copy to different server
* ansible -i hosts all -m ping -u root to check the connection

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

1. **Playbook-:**

ansible all -i "localhost," -c local -m shell -a 'echo hello world'

. Now lets write a playbook **helloworld.yml**

---

- hosts: all

tasks:

- shell: echo "hello world"

* Playbook create the file; remove the file-:

---

- hosts: all

tasks:

# - shell: echo "hello world"

- name: Create file

file: path=/home/swati/ansible/newplay state=**touch**

---

- hosts: all

tasks:

# - shell: echo "hello world"

- name: Remove file

file: path=/home/swati/ansible/newplay state=**absent**

---

- hosts: all

tasks:

- name: Copy file

copy: src=/home/swati/ansible/newplay.yml dest=/home/swati/ansible/copy.yml

**To create user using playbook-(https://simplyopensource.blogspot.com/2015/08/how-to-create-users-from-ansible-with.html)**

---

- hosts: all

sudo: yes

tasks:

- name: add user

user: name=testuser password=test

change password –(not working)

- hosts: all

tasks:

- name: Change root password

user: name=test1user update\_password=always password=test

**File module :-**

**---**

**- hosts: all**

**tasks:**

**- unarchive: src=plugin.zip dest=/home/daksha/foo**

**Script**

---

- hosts: all

tasks:

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

**ansible-playbook -i "localhost," -c local scripttest.yml to run on localhost only**

**ping –**

**---**

**- hosts: all**

**tasks:**

**- name: ping all host**

**ping:**

**Create a 2048 bit ssh key for user-:**

---

- hosts: all

sudo: yes

tasks:

- name: generate keys for user

user: name=test generate\_ssh\_key=yes ssh\_key\_bits=2048

**-include : yml**

roles

├── common

│   └── tasks

│ ├── A.yml

│   ├── B.yml

│ ├── C.yml

│ ├── D.yml

│ ├── login.yml

│ ├── logout.yml

│   └── save.yml

├── custom\_stuff\_workflow

│ └── tasks

│ └── main.yml

└── other\_stuff\_workflow

└── tasks

└── main.yml

**Connection with windows machine**

**If face the problem need to run following commands**

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

**Steps:-**

1. [**http://docs.ansible.com/ansible/intro\_windows.html**](http://docs.ansible.com/ansible/intro_windows.html)
2. [**http://darrylcauldwell.com/how-to-setup-an-ansible-test-lab-for-windows-managed-nodes-custom-windows-modules/**](http://darrylcauldwell.com/how-to-setup-an-ansible-test-lab-for-windows-managed-nodes-custom-windows-modules/)

**ansible\_ssh\_user='CTS\588378'**

**ansible\_ssh\_pass='madhu@123'**

**ansible\_ssh\_port=5986**

**ansible\_connection=winrm**

**ansible\_winrm\_transport=ntlm**

**ansible\_winrm\_server\_cert\_validation=ignore**

**connection linux to windows**

**update powershell**

**change hosts file**

**windows.yml**

**run yml**

**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

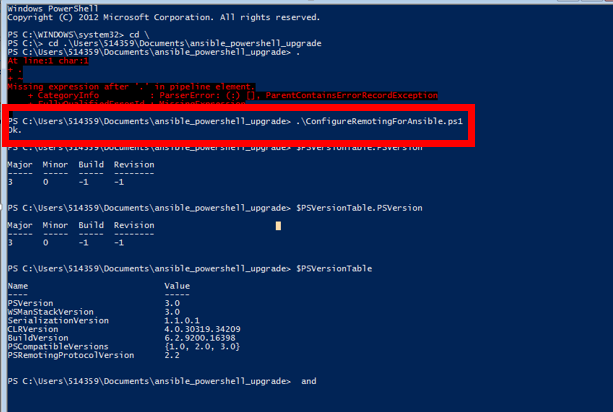
wget https://raw.github.com/pypa/pip/master/contrib/get-pip.py

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.**](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**](https://github.com/ansible/ansible/tree/devel/examples/scripts)**.**

Save these two scripts in your local and run it.



1. I**n 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='CTS\\514359'

ansible\_ssh\_pass='Sept@1205'

ansible\_ssh\_port=5986

ansible\_connection=winrm

ansible\_winrm\_transport=ntlm

ansible\_winrm\_server\_cert\_validation=ignore

1. **Create a windows.yml file**

---

- hosts: \_Development\_

gather\_facts: false

roles:

- precheck

#- common

#- git

#- mysql

#- java

#- tomcat

#- eclipse

1. **Since we are using roles in windows.yml, create a folder roles and inside which the folder structure as described under roles.**
2. **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)**

---

- name: Check for single host

fail: msg="Single host check failed, Please provide your host in Limit parameter of the Job Template"

when: "{{ play\_hosts|length }} != 1"

- debug: msg='Singe Host Check Passed!, I got executed!'

# - name: run ipconfig

# raw: ipconfig

# register: ipconfig

#- debug: var=ipconfig

- name: win ping

win\_ping :

1. **Run the windows.yml from following command**

**ansible-playbook -i hosts windows.yml**

1. \_Development\_ file

Make folder of group\_var

Inside that make a yml file oh hosts

---

DOWNLOAD\_URL: 'http://10.242.138.102/downloads'

PROJECT\_NAME: 'PetClinic'

DOWNLOAD\_DIR: '{{ ansible\_env.USERPROFILE }}\Software\tools\downloads'

INSTALL\_DIR: '{{ ansible\_env.USERPROFILE }}\Software\tools\\install'

PROJECT\_ROOT: '{{ ansible\_env.USERPROFILE}}\{{ PROJECT\_NAME}}'

This step is not mandetory one

**Link to follow -** https://groups.google.com/forum/#!topic/ansible-project/ez68TZQciXE

[**http://docs.ansible.com/ansible/intro\_windows.html**](http://docs.ansible.com/ansible/intro_windows.html)

**install exe with powershell script**

$msiFile = "C:\Users\514359\Downloads\npp.7.2.Installer.x64.exe"

$InstallDir = "<C:\Program Files (x86)\notepadd>"

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

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

}

$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)

**Stop Ansible from creating .retry**

[**http://stackoverflow.com/questions/31318881/how-do-you-stop-ansible-from-creating-retry-files-in-the-home-directory**](http://stackoverflow.com/questions/31318881/how-do-you-stop-ansible-from-creating-retry-files-in-the-home-directory)

You can disable creation of retry file in ansible by modifying ansible configuration file.

[defaults]

...

retry\_files\_enabled = False

Ansible looks for configuration file as follows

1. ./ansible.cfg
2. ~/.ansible.cfg
3. /etc/ansible/ansible.cfg

**Plugin installation to eclipse**

**With command line its working powershell also working**

**D:\Softwares\test\eclipse-java-neon-R-win32\eclipse>eclipsec.exe -nosplash -application or**

**g.eclipse.equinox.p2.director -destination D:\Softwares\test\eclipse-java-neon-R-win32\ecl**

**ipse -repository https://sourceforge.net/projects/tomcatplugin/files/updatesite -installIU**

**net.sf.eclipse.tomcat.feature.feature.group**

**http://stackoverflow.com/questions/11633529/installing-plugin-into-eclipse-using-command-line**

**http://dl.bintray.com/nodeclipse/nodeclipse/0.9.05/features/**

**eclipsec.exe -nosplash -application org.eclipse.equinox.p2.director -destination D:\Softwares\test\eclipse-java-neon-R-win32\ecl**

**ipse -repository http://dl.bintray.com/nodeclipse/nodeclipse/0.9.05 -installIU org.nodeclipse.feature.group**

**eclipsec.exe -application org.eclipse.equinox.p2.director -repository http://download.eclipse.org/releases/galileo -installIU org.eclipse.cdt.feature.group -destination D:\Softwares\eclipse\eclips\_projects**

**maven**

**eclipsec.exe -application org.eclipse.equinox.p2.director -destination D:\Softwares\test\eclipse-java-neon-R-win32\ecl**

**ipse -repository http://repo1.maven.org/maven2/.m2e/connectors/m2eclipse-egit/0.14.0/N/LATEST -installIU org.sonatype.m2e.egit.feature.feature.group**

**eclipsec.exe -nosplash -application org.eclipse.equinox.p2.director -destination D:\Softwares\eclipse\eclips\_projects -repository http://repo1.maven.org/maven2/.m2e/connectors/m2eclipse-egit/0.14.0/N/0.14.0.201509090157 -installIU org.sonatype.m2e.egit.feature.feature.group**

**org.sonatype.m2e.egit.feature**

**http://repo1.maven.org/maven2/.m2e/connectors/m2eclipse-egit/0.14.0/N/0.14.0.201509090157**

**https://sourceforge.net/projects/tomcatplugin/files/updatesite**

**net.sf.eclipse.tomcat.feature**

**eclipsec.exe -nosplash -application org.eclipse.equinox.p2.director -destination D:\Softwares\eclipse\eclips\_projects -repository https://sourceforge.net/projects/tomcatplugin/files/updatesite -installIU net.sf.eclipse.tomcat.feature.feature.group**

**D:\Softwares\eclipse>eclipsec.exe -nosplash -application org.eclipse.equinox.p2.director -destination D:\Softwares\eclipse\eclips\_projects -repository** [**https://sourceforge.net/projects/tomcatplugin/files/updatesite -installIU net.sf.eclipse.tomcat.feature.feature.group**](https://sourceforge.net/projects/tomcatplugin/files/updatesite%20-installIU%20net.sf.eclipse.tomcat.feature.feature.group)

**Work**

**Install Ansible**

**Python pip**

**Linuc to linux connection ssh**

**Work with command modules**

**Yml file for ping file raw script modules**

**Linux to windows connection**

**Winrm**

**All steps of ps**

**Work with file module**

**Script module**

**Install eclipse**

**Tomcat**

**Create shortcut**

**Install plugin to eclipse**

**- name: python2**

**hosts: webservers**

**gather\_facts: no**

**pre\_tasks:**

**- unarchive: src=plugin.zip dest=/home/daksha/foo**

**--name: python2**

**--hosts: webservers**

**remote\_user: root**

**tasks:**

**raw: apt-get -y install python-simplejson**

**---**

**- hosts: all**

**tasks:**

**- unarchive: src=plugin.zip dest=/home/daksha/foo**

**- name: python2**

**hosts: webservers**

**tasks:**

**raw: yum -y install python-simplejson**

**tasks:**

**- name: install python 2**

**raw: test -e /usr/bin/python || (apt -y update && apt install -y python-minimal)**

**---**

**- hosts: all**

**tasks:**

**- name: Installs python**

**raw: yum -y install python-simplejson**

**install pip -**

**sudo pip --proxy=http://Scloudsetadmin:C111111%23@proxy.cognizant.com:6050 install pywinrm**

**http://abregman.com/2015/12/25/ansible-write-and-run-your-first-playbook/**

**https://www.visualstudio.com/en-us/docs/tfvc/checkout-or-edit-command**

**http://docs.ansible.com/ansible/list\_of\_files\_modules.html**

**https://thornelabs.net/2014/04/19/ansible-manage-the-same-users-across-servers-with-different-passwords.html**

**http://v-punk.com/automate-password-changes-with-ansible/**

**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**

**Yml files**

**/home/Daksha/Ansible\_demo/**

1. **Windowscript\_Playbook.yml – Ps script for install exe and crete file**
2. **Windows\_installation.yml – installation of eclipse**
3. **Scripttest.yml – shell script**
4. **Playbook1 – createfile**
5. **Pingtest.yml – ping**
6. **Master and mainplaybook – include example**