

HOW TO INSTALL ANSIBLE

While installing Ansible, it's essential to have two machines. Let's name our first machine, as 'Server' that will act as our managed node, and the second machine named 'Node' that will act as the controller node.

Creating Server and Node.

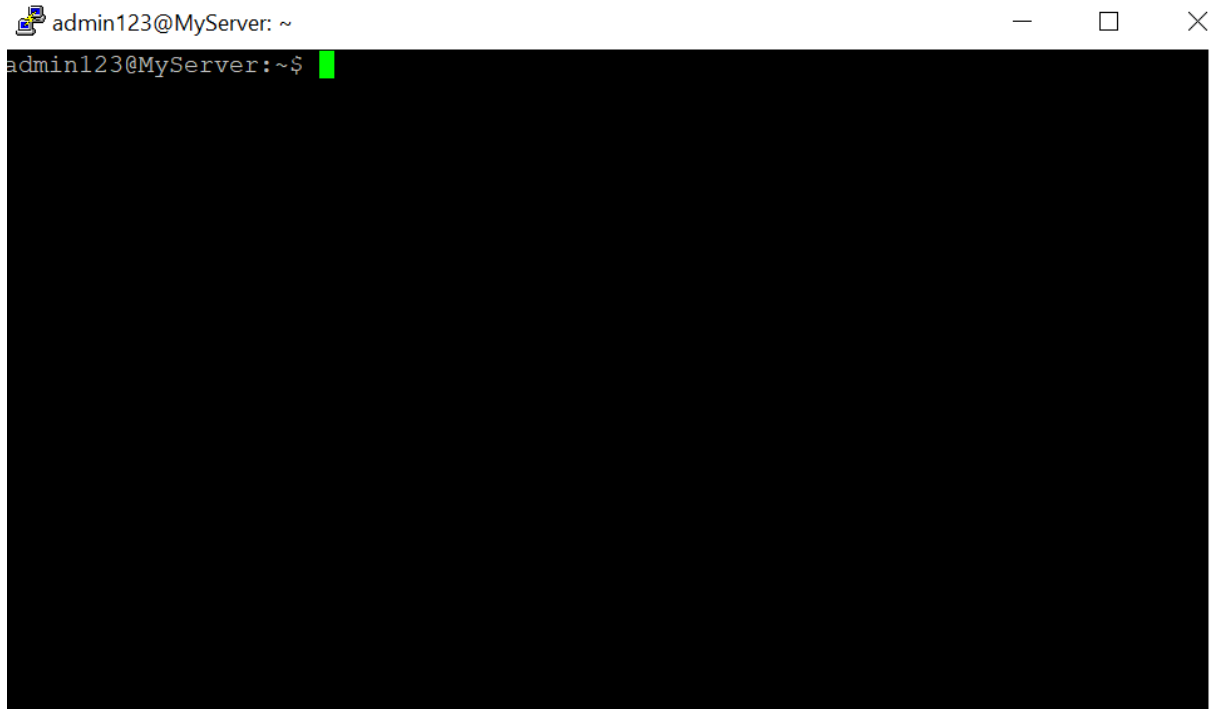


Figure 1: Creating managed node

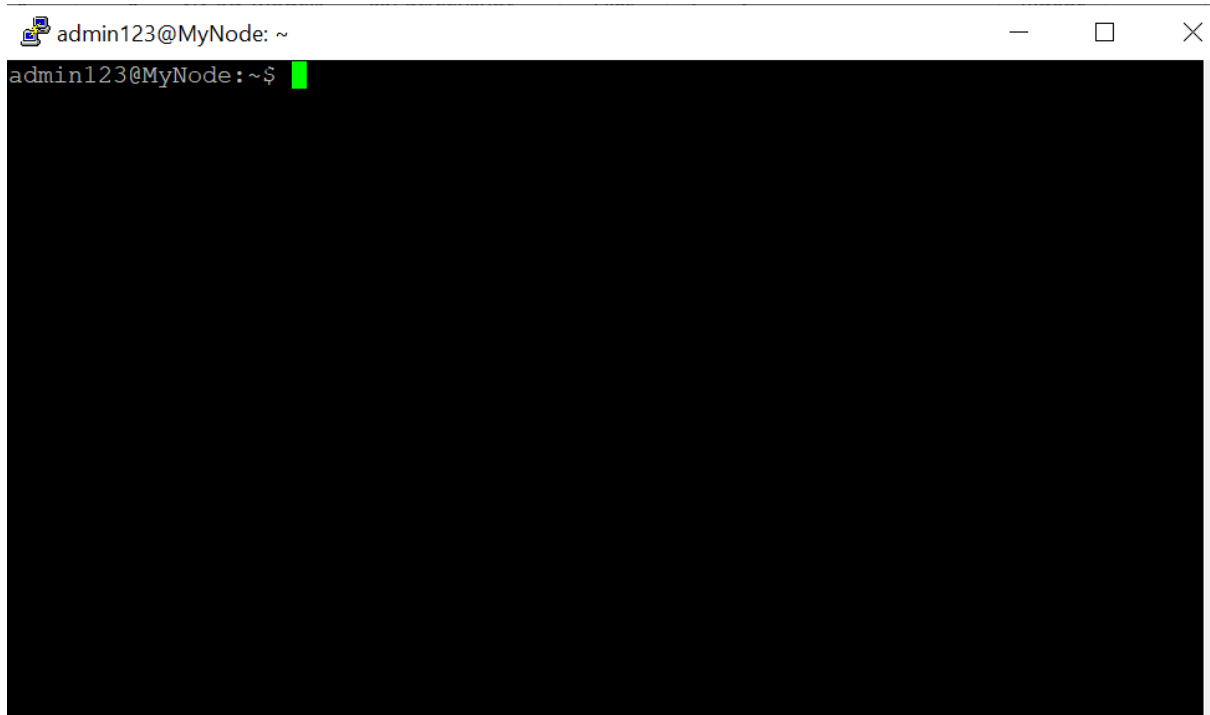


Figure 2: Creating a controller node

Step 1: Update your control node

- ➔ Before installing any new software, it is important to ensure that your existing operating version is up to date. Enter the command mentioned below to start your task.
- ➔ Use “Yum update”

```
Updated:
at.x86_64 0:3.1.13-25.el7_9
bind-libs.x86_64 32:9.11.4-26.P2.el7_9.9
bind-license.noarch 32:9.11.4-26.P2.el7_9.9
bpftool.x86_64 0:3.10.0-1160.71.1.el7
cyrus-sasl-lib.x86_64 0:2.1.26-24.el7_9
expat.x86_64 0:2.1.0-14.el7_9
glibc-common.x86_64 0:2.17-326.el7_9
grub2-common.noarch 1:2.02-0.87.0.1.el7.centos.9
grub2-pc.x86_64 1:2.02-0.87.0.1.el7.centos.9
grub2-tools.x86_64 1:2.02-0.87.0.1.el7.centos.9
grub2-tools-minimal.x86_64 1:2.02-0.87.0.1.el7.centos.9
kernel-tools.x86_64 0:3.10.0-1160.71.1.el7
krb5-libs.x86_64 0:1.15.1-54.el7_9
libsss_idmap.x86_64 0:1.16.5-10.el7_9.13
libwbclient.x86_64 0:4.10.16-19.el7_9
microcode_ctl.x86_64 2:2.1-73.13.el7_9
openssl.x86_64 1:1.0.2k-25.el7_9
python.x86_64 0:2.7.5-92.el7_9
python-perf.x86_64 0:3.10.0-1160.71.1.el7
samba-client-libs.x86_64 0:4.10.16-19.el7_9
samba-common-libs.x86_64 0:4.10.16-19.el7_9
sssd-client.x86_64 0:1.16.5-10.el7_9.13
xz.x86_64 0:5.2.2-2.el7_9
zlib.x86_64 0:1.2.7-20.el7_9
bind-export-libs.x86_64 32:9.11.4-26.P2.el7_9.9
bind-libs-lite.x86_64 32:9.11.4-26.P2.el7_9.9
bind-utils.x86_64 32:9.11.4-26.P2.el7_9.9
cloud-init.x86_64 0:19.4-7.el7.centos.6
cyrus-sasl-plain.x86_64 0:2.1.26-24.el7_9
glibc.x86_64 0:2.17-326.el7_9
grub2.x86_64 1:2.02-0.87.0.1.el7.centos.9
grub2-efi-x64.x86_64 1:2.02-0.87.0.1.el7.centos.9
grub2-pc-modules.noarch 1:2.02-0.87.0.1.el7.centos.9
grub2-tools-extra.x86_64 1:2.02-0.87.0.1.el7.centos.9
gzip.x86_64 0:1.5-11.el7_9
kernel-tools-libs.x86_64 0:3.10.0-1160.71.1.el7
libcap.x86_64 14:1.5.3-13.el7_9
libsss_nss_idmap.x86_64 0:1.16.5-10.el7_9.13
mdadm.x86_64 0:4.1-9.el7_9
openldap.x86_64 0:2.4.44-25.el7_9
openssl-libs.x86_64 1:1.0.2k-25.el7_9
python-libs.x86_64 0:2.7.5-92.el7_9
rsyslog.x86_64 0:8.24.0-57.el7_9.3
samba-common.noarch 0:4.10.16-19.el7_9
sos.noarch 0:3.9-5.el7.centos.11
tzdata.noarch 0:2022a-1.el7
xz-libs.x86_64 0:5.2.2-2.el7_9
Complete!
```

Figure 3: Updated control node

Step 2: Install EPEL Repository.

- ➔ Moving on, install the EPEL repository on the system
- ➔ Yum install epel-release

```
Package Arch Version Repository Size
Installing:
epel-release noarch 7-11 extras-openlogic 15 k
Transaction Summary
Install 1 Package
Total download size: 15 k
Installed size: 24 k
Is this ok [y/d/N]: y
Downloading packages:
epel-release-7-11.noarch.rpm | 15 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
Installing : epel-release-7-11.noarch 1/1
Verifying : epel-release-7-11.noarch 1/1
Installed:
epel-release.noarch 0:7-11
Complete!
```

Figure 4: Installed epel release

Step 3: Install Ansible

- ➔ The next step is to install the Ansible package from the EPEL repository.

```

Complete!
[root@dev ~]# yum install epel-release
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: centos.excellmedia.net
 * epel: download.nus.edu.sg
 * extras: centos.excellmedia.net
 * updates: centos.excellmedia.net
Package epel-release-7-14.noarch already installed and latest version
Nothing to do
[root@dev ~]# yum install ansible
Loaded plugins: fastestmirror, langpacks
Existing lock /var/run/yum.pid: another copy is running as pid 56549.
Another app is currently holding the yum lock; waiting for it to exit...
  The other application is: yum
    Memory : 57 M RSS (1.4 GB VSZ)
    Started: Wed Aug 3 17:30:12 2022 - 00:04 ago
    State : Running, pid: 56549
Loading mirror speeds from cached hostfile

```

Figure 5: Installing Ansible

Step 4: Create a user for Ansible

Add a user and set a password onto your Controller node

```

root@dev:~
[root@dev ~]# useradd sample
[root@dev ~]# passwd sample@123
passwd: Unknown user name 'sample@123'.
[root@dev ~]# passwd sample welcomel23
passwd: Only one user name may be specified.
[root@dev ~]# passwd
Changing password for user root.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.

```

Figure 6: Adding user on both nodes

Then, copy the public key and paste it to our Managed node with the command below

```

[root@dev ~]# ssh-copy-id root@192.168.1.190
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote system.
(if you think this is a mistake, you may want to use -f option)

[root@dev ~]# ssh root@192.168.1.190
Last login: Wed Aug 3 17:53:16 2022 from 192.168.1.176

```

Step 5: Configure our Admin User for SSH Access

Now, run the following command (in the control node) to generate an SSH key pair.

ssh-keygen

```
root@server:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:6cqNdsiAQV+tXD7Z/TJQWRqmTq+DlW/aSHQShAIjxeM root@server
The key's randomart image is:
+---[RSA 3072]-----+
|.o+o .+o. |
|. .+. .+oo |
|. .+ = +ooo |
|. . E ++oo.. |
|. o S.o.=.. |
|. . . o oo+. |
|. o.o o .oo |
|. ++. o = |
|. +.. o . |
+----[SHA256]-----+
```

Figure 7: Generating public key on control node

Then, copy the public key and paste it to our Managed node with the command below.

```
[sample@dev .ssh]$ ssh-copy-id root@192.168.1.190
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sample/.ssh/id_rsa.pub"
The authenticity of host '192.168.1.190 (192.168.1.190)' can't be established.
ECDSA key fingerprint is SHA256:QYDMWoXYaYbAKHdYq3wWVpNOErPadcL2ifivhXfaZZ0.
ECDSA key fingerprint is MD5:de:de:87:8c:6e:e9:41:09:ff:2b:3d:0e:78:cl:el:84.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.1.190's password:

Number of key(s) added: 1

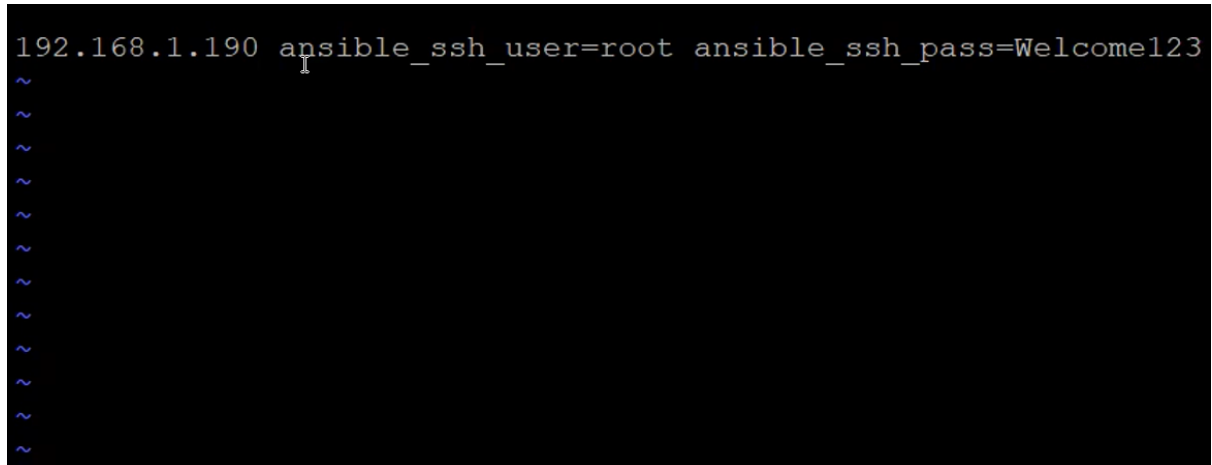
Now try logging into the machine, with: "ssh 'root@192.168.1.190'"
and check to make sure that only the key(s) you wanted were added.
```

Figure 8: Copying public key to the managed node

Step 6: Create an Inventory

An inventory list is created to identify your managed nodes.

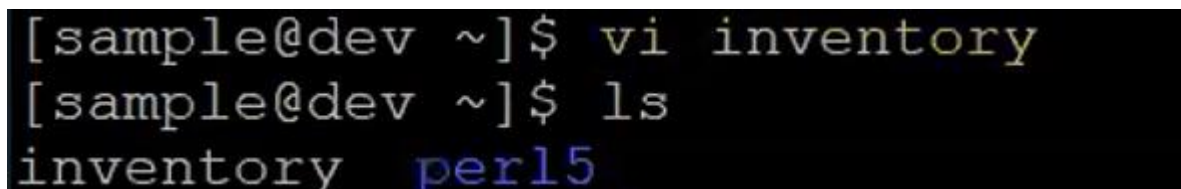
Log in to your control node as the admin user to connect the Managed node to the inventory.



```
192.168.1.190 ansible_ssh_user=root ansible_ssh_pass=Welcome123
~
~
~
~
~
~
~
~
~
~
~
```

Figure 9: Creating an inventory

The inventory of hostfile is created in the controller node.



```
[sample@dev ~]$ vi inventory
[sample@dev ~]$ ls
inventory perl5
```

Figure 10: Created inventory

Step 7: Create an ansible playbook

Here, we will create a simple Ansible playbook by installing Nginx on the Managed Node.

First, log onto your Controller Node as the "Simplilearn" user and create a file with a descriptive name.

```
vim /home/simplilearn/install-nginx.yml
```

```
--
- hosts: client-apps-com
  become: yes
  tasks:
    - name: Installs nginx web server
      Apt: name: nginx state: started update_cache: true
  notify:
    - start nginx
  handlers:
    - name: start nginx
  service:
    name: nginx
    state: started
~
~
```

Figure 11: Writing a playbook

Created an ansible playbook in yml

```
[sample@dev ~]$ vi install-nginx.yml
[sample@dev ~]$ ls
install-nginx.yml  inventory  perl5
[sample@dev ~]$
```

Figure 12: Playbook is built

Step 8: Run the playbook

Our Ansible playbook is built. Now, to run the playbook, type the following command on the controller node:

ansible-playbook -i /home/admin/inventory /home/admin/install-nginx.yml

In the command above, we have added the inventory file with the "-i" option, followed by the playbook path.

```
nginx: [sample@dev ~]$ ansible-playbook -i /home/sample/inventory /home/sample/install-nginx
nginx: x.yml
nginx: PLAY [all] *****
nginx: TASK [Gathering Facts] *****
[root@Redire ok: [192.168.1.190]
Job fo
status TASK [Install nginx web server] *****
[root@ ok: [192.168.1.190]
[root@
[root@ PLAY RECAP *****
• ngin 192.168.1.190 : ok=2 changed=0 unreachable=0 failed=0 skipped=
Loa
Act ignored=0
```

Figure 13: Running playbook

