

Lab 1: Installing Ansible

Introduction:

Ansible is an **agentless** automation tool that by default manages machines over the **SSH** protocol. Once installed, Ansible does not add a database, and there will be no daemons to start or keep running. You only need to install it on one machine and it can manage an entire fleet of remote machines from that central point. When Ansible manages remote machines, it does not leave software installed or running on them.

In this Lab, you will install and configure the **pre-requisites** for deploying and managing Ansible.

Prerequisites:

Install Ansible on a **Control node**, which then uses **SSH** (by default) to communicate with your managed nodes (those end devices you want to automate).

Control node requirements:

Ansible can be run from any machine with **Python 2** (version 2.7) or **Python 3** (versions 3.5 and higher) installed. This includes Red Hat, Debian, CentOS, macOS, any of the BSDs, and so on. Windows is not supported for the control node.

Managed node requirements:

On Managed nodes, you need a way to communicate, which is normally **SSH**. By default, this uses SFTP. If that's not available, you can switch to SCP in `ansible.cfg`. You also need **Python2** (version 2.6 or later) or **Python 3** (version 3.5 or later).

Note: Below table contains details of the servers that we will be using in this lab.

Host Name	IP Address	Role	OS	RAM	Core	NIC
ansi-master	192.168.100.150	Controller node	CentOS-8	8GB	2	1
ansi-node1	192.168.100.151	Managed Node	CentOS-8	4GB	2	1
ansi-node2	192.168.100.152	Managed Node	CentOS-8	4GB	2	1
ansi-node3	192.168.100.152	Managed Node	CentOS-8	4GB	2	1

Installing Ansible on CentOS (Control Node):

1 Login into the **Control node (ansi-master)** as **root** user with password as **linux**.

1.1 Make sure SELinux is disabled

Security-Enhanced Linux (SELinux) is a **mandatory access control** (MAC) security mechanism implemented in the kernel.

SELinux has **three basic** modes of operation, of which Enforcing is set as the installation default mode.

- **Enforcing:** The default mode which will enable and enforce the SELinux security policy on the system, denying access and logging actions
- **Permissive:** In Permissive mode, SELinux is enabled but will not enforce the security policy, only warn and log actions. Permissive mode is useful for troubleshooting SELinux issues.
- **Disabled:** SELinux is turned off

```
# sed -i 's/enforcing/disabled/g' /etc/selinux/config
# setenforce 0
# sestatus
```

Output:

```
[root@ansi-master ~]# sed -i 's/enforcing/disabled/g' /etc/selinux/config
[root@ansi-master ~]# setenforce 0
setenforce: SELinux is disabled
[root@ansi-master ~]# sestatus
SELinux status:                disabled
```

1.2 Let us Enable and Start Chrony service (NTP Server).

- **Chrony** is a versatile implementation of the **Network Time Protocol** (NTP).
- The chrony suite installed by default and provides
- **chronyd** - Chrony daemon and **chronyc** - Command-line utility.

```
# systemctl enable --now chronyd
# systemctl status chronyd --no-pager
```

Output:

```
[root@ansi-master ~]# systemctl enable --now chronyd
Created symlink /etc/systemd/system/multi-user.target.wants/chronyd.service → /usr/lib/systemd/system/chronyd.service.
[root@ansi-master ~]# systemctl status chronyd --no-pager
● chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-10-12 15:09:15 IST; 5s ago
     Docs: man:chronyd(8)
           man:chrony.conf(5)
   Process: 2910 ExecStartPost=/usr/libexec/chrony-helper update-daemon (code=exited, status=0/SUCCESS)
   Process: 2906 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, status=0/SUCCESS)
  Main PID: 2908 (chronyd)
    Tasks: 1 (limit: 100795)
   Memory: 788.0K
   CGroup: /system.slice/chronyd.service
           └─2908 /usr/sbin/chronyd

Oct 12 15:09:15 ansi-master systemd[1]: Starting NTP client/server...
Oct 12 15:09:15 ansi-master chronyd[2908]: chronyd version 3.5 starting (+CMDMON +NTP +REFCLOCK +RTC +_DEBUG)
Oct 12 15:09:15 ansi-master chronyd[2908]: Using right/UTC timezone to obtain leap second data
Oct 12 15:09:15 ansi-master systemd[1]: Started NTP client/server.
Hint: Some lines were ellipsized, use -l to show in full.
```

1.3 Let us verify and update the chrony sources by using below command

```
# chronyc sources -v
```

Output:

```
[root@ansi-master ~]# chronyc sources -v
210 Number of sources = 4

.-- Source mode  '^' = server, '=' = peer, '#' = local clock.
/  .-- Source state '*' = current synced, '+' = combined , '-' = not combined,
| /    '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||
||      Reachability register (octal) --.      .- xxxx [ yyyy ] +/- zzzz
||      Log2(Polling interval) --.      |      | xxxx = adjusted offset,
||      \      |      |      |      | yyyy = measured offset,
||      \      |      |      |      | zzzz = estimated error.
||
=====
MS Name/IP address             Stratum Poll Reach LastRx Last sample
=====
^~ static.15.192.216.95.cli>    2    6    17    32   -2863us[-4299us] +/-   90ms
^~ ntp5.mum-in.hosts.301-mo>    2    6    17    33    +68us[-1368us] +/-  117ms
^+ 139.59.55.93                 2    6    17    33    +688us[ -748us] +/-   26ms
^* 104.211.76.226               1    6    17    32   -651us[-2087us] +/-   16ms
```

1.4 Let us disable firewall

Note:

- A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.
- A firewall typically establishes a barrier between a trusted network and an untrusted network, such as the Internet.

```
# systemctl disable --now firewalld
# systemctl status firewalld --no-pager
```

Output:

```
[root@ansi-master ~]# systemctl disable --now firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@ansi-master ~]# systemctl status firewalld --no-pager
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)

Oct 09 14:30:42 ansi-master systemd[1]: Starting firewalld - dynamic firewall daemon...
Oct 09 14:30:42 ansi-master systemd[1]: Started firewalld - dynamic firewall daemon.
Oct 09 14:30:42 ansi-master firewalld[1028]: WARNING: AllowZoneDrifting is enabled. This is considered..t now.
Oct 12 15:12:36 ansi-master systemd[1]: Stopping firewalld - dynamic firewall daemon...
Oct 12 15:12:37 ansi-master systemd[1]: firewalld.service: Succeeded.
Oct 12 15:12:37 ansi-master systemd[1]: Stopped firewalld - dynamic firewall daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

1.5 Add an entry to `/etc/hosts` file for local name resolution

```
# cat > /etc/hosts <<EOF
192.168.100.150 ansi-master.example.com ansi-master
192.168.100.151 ansi-node1.example.com ansi-node1
192.168.100.152 ansi-node2.example.com ansi-node2
192.168.100.153 ansi-node3.example.com ansi-node3
127.0.0.1 localhost
EOF
```

1.6 Verify the `/etc/hosts` file updated successfully, by executing below command

```
# cat /etc/hosts
```

Output:

```
[root@ansi-master ~]# cat /etc/hosts
192.168.100.150 ansi-master.example.com ansi-master
192.168.100.151 ansi-node1.example.com ansi-node1
192.168.100.152 ansi-node2.example.com ansi-node2
192.168.100.153 ansi-node3.example.com ansi-node3
127.0.0.1 localhost
```

1.7 Test network connectivity between servers to ensure name resolution is working.

```
# ping -c 5 ansi-node1
```

Output:

```
[root@ansi-master ~]# ping -c 5 ansi-node1
PING ansi-node1.example.com (192.168.100.151) 56(84) bytes of data.
64 bytes from ansi-node1.example.com (192.168.100.151): icmp_seq=1 ttl=64 time=0.412 ms
64 bytes from ansi-node1.example.com (192.168.100.151): icmp_seq=2 ttl=64 time=0.520 ms
64 bytes from ansi-node1.example.com (192.168.100.151): icmp_seq=3 ttl=64 time=0.411 ms
64 bytes from ansi-node1.example.com (192.168.100.151): icmp_seq=4 ttl=64 time=0.442 ms
64 bytes from ansi-node1.example.com (192.168.100.151): icmp_seq=5 ttl=64 time=0.534 ms
```

1.8 Install and Configure EPEL repository:

EPEL (Extra Packages for Enterprise Linux) is an open-source and free community-based repository project from Fedora team which provides 100% high-quality add-on software packages for Linux distribution including RHEL (Red Hat Enterprise Linux), CentOS.

```
# dnf install epel-release -y
```

Output:

```
[root@ansi-master ~]# dnf install epel-release -y
CentOS Linux 8 - Extras                                1.8 kB/s | 1.5 kB    00:00
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing: epel-release	noarch	8-11.el8	extras	24 k

```
Transaction Summary
Install 1 Package

Total download size: 24 k
Installed size: 35 k
Downloading Packages:
epel-release-8-11.el8.noarch.rpm                        316 kB/s | 24 kB    00:00
-----
Total                                                    67 kB/s | 24 kB    00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : epel-release-8-11.el8.noarch  1/1
  Running scriptlet: epel-release-8-11.el8.noarch 1/1
  Verifying      : epel-release-8-11.el8.noarch 1/1
Installed products updated.
```

1.9 Let us install Ansible Package

```
# dnf -y install ansible
```

Output:

```
[root@ansi-master ~]# dnf -y install ansible
No such command: -y. Please use /usr/bin/dnf --help
It could be a DNF plugin command, try: "dnf install 'dnf-command(-y)'"
[root@ansi-master ~]# dnf -y install ansible
Extra Packages for Enterprise Linux Modular 8 - x86_64      8.9 kB/s | 955 kB    01:47
Extra Packages for Enterprise Linux 8 - x86_64             138 kB/s | 10 MB     01:17
Last metadata expiration check: 0:00:02 ago on Tue 12 Oct 2021 04:53:21 PM IST.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing: ansible	noarch	2.9.25-1.el8	epel	17 M
Installing dependencies:				
libsodium	x86_64	1.0.18-2.el8	epel	162 k
python3-babel	noarch	2.5.1-5.el8	appstream	4.8 M
python3-bcrypt	x86_64	3.1.6-2.el8_1	epel	44 k
python3-jinja2	noarch	2.10.1-2.el8_0	appstream	538 k
python3-jmespath	noarch	0.9.0-11.el8	appstream	45 k
python3-markupsafe	x86_64	0.23-19.el8	appstream	39 k
python3-pyasn1	noarch	0.3.7-6.el8	appstream	126 k
python3-pynacl	x86_64	1.3.0-5.el8	epel	100 k
sshpas	x86_64	1.06-9.el8	epel	27 k

```
Installing weak dependencies:
```

1.10 Let us verify that the Ansible is installed:

```
# ansible --version
```

Output:

```
[root@ansi-master ~]# ansible --version
ansible 2.9.25
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.6/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.6.8 (default, Sep 21 2021, 20:17:36) [GCC 8.4.1 20200928 (Red Hat 8.4.1-1)]
```

1.11 let us verify the installation of ansible-python on the localhost by using the setup mode:

```
# ansible -m setup localhost | grep ansible_python_version
```

Output:

```
[root@ansi-master ~]# ansible -m setup localhost | grep ansible_python_version
"ansible_python_version": "3.6.8",
```

2 Let us run the below script on all managed nodes those are listed below

1. **ansi-node1**
2. **ansi-node2**
3. **ansi-node3**

```
sed -i 's/enforcing/disabled/g' /etc/selinux/config
setenforce 0
systemctl enable --now chronyd
sleep 10
chronyc sources
systemctl disable --now firewalld
cat > /etc/hosts <<EOF
192.168.100.150  ansi-master.example.com  ansi-master
192.168.100.151  ansi-node1.example.com  ansi-node1
192.168.100.152  ansi-node2.example.com  ansi-node2
192.168.100.153  ansi-node3.example.com  ansi-node3
127.0.0.1 localhost
EOF
dnf install epel-release -y
dnf -y install ansible
ansible -m setup localhost | grep ansible_python_version
dnf update -y
```