How to Install Ansible on CentOS 7

Top of Form

Introduction

Ansible is an open-source automation software written in Python. It runs on UNIX-like systems and can provision and configure both UNIX-like and Windows systems. Unlike other automation software, Ansible does not require an agent to run on a target system. It leverages on the SSH connection and python interpreter to perform the given tasks on the target system. Ansible can be installed on a cloud server to manage other cloud servers from a central location, or it can also be configured to use on a personal system to manage cloud or on-premises systems.

Prerequisites

* Cloud VPS or Dedicated Server with at least 1GB RAM and CentOS 7 installed.
* You must be logged in via SSH as sudo or root user. This tutorial assumes that you are logged in as a sudo user.

**Step 1: Update the System**

Update the system with the latest packages and security patches using these commands.

sudo yum -y update

**Step 2: Install EPEL Repository**

EPEL or Extra Packages for Enterprise Linux repository is a free and community-based repository which provide many extra open source software packages which are not available in default YUM repository.

We need to install EPEL repository into the system as Ansible is available in default YUM repository is very old.

sudo yum -y install epel-repo

Update the repository cache by running the command.

sudo yum -y update

**Step 3: Install Ansible**

Run the following command to install the latest version of Ansible.

sudo yum -y install ansible

You can check if Ansible is installed successfully by finding its version.

ansible --version

You should see a similar output.

ansible --version

ansible 2.7.10

config file = None

configured module search path = [u'/home/sneluser/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python2.7/site-packages/ansible

executable location = /usr/bin/ansible

python version = 2.7.5 (default, Apr 9 2019, 14:30:50) [GCC 4.8.5 20150623 (Red Hat 4.8.5-36)]

Ansible is now installed on your server.

**Step 4: Testing Ansible (Optional)**

Now that we have Ansible installed, let’s play around to see some basic uses of this software. This step is optional.

Consider that we have three different which we wish to manage using Ansible. In this example, I have created another three CentOS 7 cloud server with username root and password authentication. The IP address assigned to my cloud servers are

* 192.168.0.101
* 192.168.0.102
* 192.168.0.103

You can have less number of servers to test with.

$ ansible all --list-hosts

Step 4.1 Generate SSH Key Pair

Although we can connect to remote hosts using a password through Ansible it is recommended to set up key-based authentication for easy and secure logins.

Generate an SSH key pair on your system by running the command.

ssh-keygen

You will be prompted to provide a name and password for key pair. Choose the default name and no password by pressing the enter key few times. You should see the following output.

ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/sneluser/.ssh/id\_rsa):

Created directory '/home/sneluser/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/sneluser/.ssh/id\_rsa.

Your public key has been saved in /home/sneluser/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:AAtQYpD0cuE0XyteDXvx55utFgDd1eQtKHsB4mvt+e4 sneluser@host.neetusuthar.com

The key's randomart image is:

+---[RSA 2048]----+

|\*\*o+. o..o . .oo|

|o.+.+o..=ooo o .o|

| . +.o.+.oo.o.. o|

| o . o..o +o. . |

| . S o o. |

| . . o .+ |

| o o.. |

| . .. |

| oE. |

+----[SHA256]-----+

**Step 4.2 Copy Public Key into Target Server**

Now that our key pair is ready, we need to copy the public key into our target systems. Run the following command to copy the public key into the first server.

ssh-copy-id root@192.168.0.101

Type yes when prompted to trust target host’s fingerprint. Put the password of root account when prompted. The output will be similar to shown below.

ssh-copy-id root@192.168.0.101

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sneluser/.ssh/id\_rsa.pub"

The authenticity of host '192.168.0.101 (192.168.0.101)' can't be established.

ECDSA key fingerprint is SHA256:d/D6NKU57CXaY4T3pnsIUycEPDv0Az2MiojBGjNj3+A.

ECDSA key fingerprint is MD5:5e:24:6a:13:99:e7:67:47:06:3e:2d:3e:97:d8:11:e7.

Are you sure you want to continue connecting (yes/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@192.168.0.101's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh root@192.168.0.101'"

and check to make sure that only the key(s) you wanted were added.

You can now try to login to the target system by running the command.

ssh root@192.168.0.101

It should log you in without asking for a password.  
Repeat step 4.2 for all the remaining two hosts.

**Step 4.3 Configure Ansible Hosts**

By default, Ansible reads the host file from the location /etc/ansible/hosts. Open the hosts file into the editor.

sudo vi /etc/ansible/hosts

Replace the existing content with the following lines into the editor. Make sure to replace your actual IP address and username.

[servers]

server1 ansible\_host=192.168.0.101 ansible\_user=root

server2 ansible\_host=192.168.0.102 ansible\_user=root

server3 ansible\_host=192.168.0.103 ansible\_user=root

Save the file and exit from the editor.

**Step 4.4 Connect using Ansible**

We have done the minimal configuration required to connect to the remote machine using Ansible. Run the following command to ping the host using Ansible *ping* module.

ansible -m ping all

If your server can successfully connect to the remote hosts, you should see the following output.

ansible -m ping all

server1 | SUCCESS => {

"changed": false,

"ping": "pong"

}

server2 | SUCCESS => {

"changed": false,

"ping": "pong"

}

server3 | SUCCESS => {

"changed": false,

"ping": "pong"

}

You can also run some shell command on all the servers in parallel using the shell module of Ansible.

ansible -m shell -a 'yum -y update' all

You can also run your command on a single host if you wish.

ansible -m shell -a 'yum -y update' server1

yum update --exclude=PACKAGENAME

317 sudo yum install epel-release

318 sudo yum -y install ansible

319 ansible --version

320 java -version

321 sudo vi /etc/ansible/hosts

322 ansible --list-hosts

323 ansible all --list-hosts

325 sudo mkdir /etc/ansible/group\_vars

326 sudo vim /etc/ansible/group\_vars/servers

---

ansible\_ssh\_user: itech

327 ansible -m ping all

328 sudo visudo

itech ALL=(ALL) ALL

329 ssh itech@192.168.1.72

330 ssh-keygen

331 ssh-copy-id itech@192.168.1.72

332 ssh-copy-id itech@192.168.1.128

333 ssh itech@192.168.1.72

334 ssh itech@192.168.1.128

335 ansible -m ping all

339 ansible all -m shell -a "df -h"

340 sudo vi /etc/ansible/hosts

341 ansible all --list-hosts

342 ansible -m shell -a "df -h" aether1

343 ansible all -m ping

344 sudo vi /etc/ansible/hosts

345 ansible all -m ping

348 ansible all -m shell -a 'echo $TERM'

349 ansible aether -m yum -a "name=git state=present"

350 ansible aether -m command -a uptime

351 ansible aether -m yum -a "name=\* state=latest"