Ansible Installation Document

Table of Contents

[About this document 3](#_Toc69486121)

[Purpose of this document 3](#_Toc69486122)

[Audience 3](#_Toc69486123)

[Abbreviations 3](#_Toc69486124)

[Overview 4](#_Toc69486125)

[Install Ansible on centos-7 5](#_Toc69486126)

[Remote machine setup: 5](#_Toc69486127)

[Generate SSH Key Pair 5](#_Toc69486128)

[Copy Public Key into Target Server 7](#_Toc69486129)

[Target Machine Setup 8](#_Toc69486130)

[Add to Host file 8](#_Toc69486131)

[Sample YAML File to Run on Target Machine 9](#_Toc69486132)

[Config\_Registry run on Target machines: 10](#_Toc69486133)

[Run the docker service via Shell file 12](#_Toc69486134)

# **About this document**

## Purpose of this document

The purpose of this document is to guide on how to install the Ansible on centos 7.

## Audience

All the Devops and network administrator level users of MDAP are the audience of this document.

## Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Expansion** |
|  |  |

# Overview

Ansible is a radically simple IT automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs.

# **Install Ansible on centos-7**

To get Ansible for CentOS 7, first ensure that the CentOS 7 EPEL repository is installed:

yum install epel-release

Once the repository is installed, install Ansible with yum:

sudo yum install ansible

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

Sudo ansible --version

# **Remote machine setup:**

## 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.

[aigilx@host]$ ssh-keygen

Generating public/private rsa key pair.

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

Created directory '/home/aigilx/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

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

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

The key fingerprint is:

SHA256:AAtQYpD0cuE0XyteDXvx55utFgDd1eQtKHsB4mvt+e4 aigilx@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]-----+

# 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.

[aigilx@host]$ ssh-copy-id root@192.168.0.101

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

# **Target Machine Setup**

yum groupinstall “Development Tools”

yum update

yum install python3 python3-pip

pip3 install docker docker-compose

pip install --upgrade pip

yum install python

yum install centos-release-scl

yum install rh-python36

scl enable rh-python36 bash

## Add to Host file

vi /etc/ansible/hosts

[servers]

server1 ansible\_host=192.168.0.101

server2 ansible\_host=192.168.0.102

server3 ansible\_host=192.168.0.103

## Sample YAML File to Run on Target Machine

- hosts: all

vars:

ansible\_python\_interpreter: /opt/rh/rh-python36/root/usr/bin/python3

tasks:

- name: Download docker image

docker\_image:

name: httpd

tag: latest

source: pull

- name: Launching Httpd container

docker\_container:

name: httpd

image: httpd

state: started

exposed\_ports:

- "80"

ports:

- "8888:80"

Save the file as apache.yml and run on ansible machine with below mentioned command.

# ansible-playbook apache.yml

## Config\_Registry run on Target machines:

[root@0I ~]# cat cc.yml

- hosts: all

vars:

ansible\_python\_interpreter: /usr/bin/python3

tasks:

- name: Download docker image

docker\_image:

name: 172.16.10.88:5000/dev/config-registry-microservice-update

tag: latest

source: pull

- name: Launching Config Registry

docker\_container:

name: DEV-CONFIG-REGISTRY

image: 172.16.10.88:5000/dev/config-registry-microservice-update

state: started

env:

HOST: "172.16.10.222"

PORT: "5432"

USERNAME: "devuser"

PASSWORD: "h@Rd30rk"

DB\_MS\_URL: "http://172.16.10.243:3045"

CLIENT\_MSG\_BUILDER\_URL: "http://172.16.10.243:3022"

MS\_PORT: "3043"

DBNAME: "core"

SCHEMANAME: "core"

exposed\_ports:

- "3043"

ports:

- "3043:3043"

Save the file as config-registry.yml and run on ansible machine with below mentioned command.

# ansible-playbook config-registry.yml

# **Run the docker service via Shell file**

Make a docker running service script into a shell file then run that shell file via ansible playbook. Save the file as test.sh

#!/bin/bash

docker service create --name --network int-svc-net --log-opt max-size=10m --log-opt max-file=5 --env ENVURL="http://paris.local" --env MS\_PORT="3320" -p 3320:3320 regsitry/dev/social-history-relationship-ms:latest

- hosts: all

vars:

ansible\_python\_interpreter: /usr/bin/python

tasks:

- name: Transfer the script

copy: src=test.sh dest=/ mode=0777

- name: Execute the script

command: sh /test.sh

The output should be like given mentioned

[root@0I ~]# ansible-playbook remote-shell.yml

PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [172.168.0.10]

TASK [Transfer the script] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.0.10]

TASK [Execute the script] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.0.10]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.0.10: ok=3 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0