**How to Install Ansible on Ubuntu 20.04 LTS**

Home » How to Install Ansible on Ubuntu 20.04 LTS

In this article we are going to cover How to install Ansible on Ubuntu 20.04 LTS, Configure Ansible, exchanged SSH keys from master to slave node and Tested the Ansible setup.

**Table of Contents**

Prerequisites:

1. Minimum 3 instance of Ubuntu 1 for Ansible server and another 2 for Nodes
2. SSH access with sudo privileges

**#1.Adding Ansible Repository in Ubuntu**

Step1 :Update your system packages:

sudo apt-get update

Step2: First Install Required packages to install Ansible.

sudo apt install software-properties-common

Step3: Add the ansible repository via PPA

sudo add-apt-repository --yes --update ppa:ansible/ansible

**#2.Install Ansible on Ubuntu 20.04 LTS**

Step1: Install Ansible on Ubuntu 20.04 LTS

sudo apt-get install ansible

Step2: To check version :

ansible --version

**#3.Create Inventory file in Ansible**

step1:Now go to host file inside ansible server and paste private ip of node1 and node2:

sudo nano /etc/ansible/hosts

You can create group and paste ip address like below:

[devopshint]

65.2.140.xx

65.3.144.xx

**#4.Configuration of Ansible server**

Step1: This host file is only working after updating ansible.cfg file inside ansible server:

sudo nano /etc/ansible/ansible.cfg

Uncomment below lines

inventory = /etc/ansible/hosts

sudo-user = root

Step2: Now, create one user in all these instance(ansible server and nodes)

sudo adduser ansible

sudo passwd ansible

now navigate the ansible user

su - ansible

**#5.Add User to the sudo Group**

Step1: Give some privileged in all nodes(ansible server and node) using below command:

sudo visudo

ansible ALL=(ALL) NOPASSWD:ALL

**#6.Update ssh\_config file**

Step1: Now we have to some changes in ssh-config file in ansible server and nodes:

sudo nano /etc/ssh/sshd\_config

Then you need to uncomment these two lines

PubkeyAuthentication yes

PasswordAuthentication yes

sudo systemctl restart sshd

**#7.Ansible Master and Slave Setup**

Step1: login to ansible in ansible server using below command:

su - ansible

**#8.Setup SSH keys and share it among managed nodes**

Step1: we need to generate ssh keygen in ansible server

ssh-keygen

cd .ssh

Step2:Now run the below command using private ip of your node:

ssh-copy-id ansible@{private address }

**Why we use Ansible ad hoc Commands ?**

An adhoc commands can be use to do simple quick task to perform some operation at that command line itself. And they are not reusable.

For e.g, if you wants to restart your infrastructure servers or get OS information on small and quick task you can use ad hoc commands.

Syntax:

ansible <group\_name> -m <module> -a <arguments>

Test the connection

sudo nano /etc/ansible/hosts

ansible all -m ping

ansible DevOps -m ping

**File Operations:**

1. To check there is any files or directories

ansible devopshint -a “ls”

ansible all -a “touch file11”

ansible@ip-172-31-41-9:~$ ls

ansible DevOps -m copy -a "src=./sample dest=/tmp/sample"

1. System Information:

ansible devopshint -m setup

ansible devopshint -m setup -a 'filter=ansible\_os\_family'

ansible devopshint -m setup -a 'filter=ansible\_memory\_mb'

ansible devopshint -m shell -a "top -c -b | head -15"

ansible devopshint -m shell -a "ps -eo pid,ppid,%mem,%cpu,cmd --sort=-%mem | head"

**What is Ansible Playbook?**

Ansible modules execute tasks. One or more Ansible tasks can be combined to make a play. Two or more plays can be combined to create an Ansible playbook. Ansible playbooks are lists of tasks that automatically execute against hosts. Groups of hosts form your Ansible inventory.

Each module within an Ansible playbook performs a specific task. Each module contains metadata that determines when and where a task is executed, as well as which user executes it. To write your ansible playbook you should know what is inventory, modules and tasks.

**Ansible Inventory:**

So the inventory is nothing but a list of hosts or group of hosts.The default location of your hosts inventory file is /etc/ansible/hosts.

**Ansible Module:**

Modules are the programs that perform the actual work of the task of a play. These modules we use under task sessions in our playbook.

**Ansible Tasks:**

A tasks nothing but a set of instructions which can be performed by using modules so in the task we usually called modules

**Ansible Playbook**

sudo vi demo.yml

---

- name: Setting up Apache webserver

hosts: devopshint

become: true

tasks:

- name: Install apache2

apt: name=apache2 update\_cache=yes state=latest

Lets understand ansible playbook

1. Ansible playbook starts with a (—) three dash
2. Next line starts with (-) single dash. Name is optional here.
3. Then hosts nothing but lists of the server are target nodes. Hosts like all or name of group.
4. If you want to use become a root then you can use become true
5. Then next is task in task we are going to perform operators. Tasks are a list of the actions which our playbook performs on a remote host.
6. So here we are going to use the apt module. you can go to the official page of ansible where you can check modules and their parameters. And if you scroll down you can see examples of ansible playbooks

And in ansible playbook **indentation** is very important.

ansible-playbook demo.yml --check >>default location inventory file

ansible-playbook -i myinventory demo.yml --check >>myinventory is your inventory name

ansible-playbook demo.yml >>to run playbook

Check in browser apache2 installed or not in our remote hosts.

**What is Ansible Variables?**

Ansible uses variables to manage differences between systems. With Ansible, you can execute tasks and playbooks on multiple different systems with a single command. To represent the variations among those different systems, you can create variables with standard YAML syntax, including lists and dictionaries.

You can define these variables in your playbooks, in your inventory, in re-usable files or roles, or at the command line. You can also create variables during a playbook run by registering the return value or values of a task as a new variable.

Ansible Variable Declare Types

1. Playbook with vars
2. Create new file for variable
3. Declaring variable in the inventory file

ansible all -m setup

ansible all -m setup -a 'filter=ansible\_hostname'

**Define variables in Ansible playbook**

sudo vi samp.yml

---

-

hosts: devopshint

vars:

title: Welcome to DevOpsHint!

tasks:

- name: Ansible Playbook variable definition example

debug:

msg: "{{ title }}"

ansible-playbook samp.yml

**Create new file for variable in Ansible**

Vars1.yml

package\_name: apache2

demo.yml

---

- name: Setting up Apache webserver

hosts: devopshint

become: true

tasks:

- include\_vars: vars.yml

- name: Install apache2

apt: name={{ package\_name }} update\_cache=yes state=latest