**Lesson 4 Demo 1**

**Executing Ad-Hoc Commands**

|  |
| --- |
| **Objective:** To executing Ad-Hoc commands    **Prerequisites:** You need to have Python 2.7 or higher, minimum 8 GB RAM, and SSH or SCP communicator.    **Tools required:** Ansible |

**Steps to be followed:**

1. Installing ansible on the main node
2. Generating SSH key pair on the main node
3. Copying the SSH key on other two nodes
4. Updating the host file with the hosts IP address
5. Establishing connectivity between the hosts specified in the host file and the Ansible server
6. Collecting the uptime of our managed hosts

|  |
| --- |
| **Note**: The node on which we install Ansible is called Ansible Server. |

**Step 1: Installing ansible main node**

* 1. Refer the demo 1 of lesson 2 to install Ansible.
  2. The We will refer to the node on which we installed ansible as the ansible server.

**Step 2: Generate a private or public key for the Ansible server**

1. Use the following command to generate the SSH key on the Ansible server:

**ssh-keygen**

Text

Description automatically generated

1. **Copy the SSH key to the nodes that will be managed through ansible server**

|  |
| --- |
| **Note**: In this demo, we have two nodes. One is the localhost of ansible server and the other one is node 2 with IP (40.86.1.9) |

* 1. Use the following command to copy the public key in a file named “authorized key” in localhost:

**cat .ssh/id\_rsa.pub >> .ssh/authorized\_keys**



* 1. Use the following command to check the SSH connection with localhost (if you get a welcome screen as shown in the snippet, then the SSH connection is successful.).

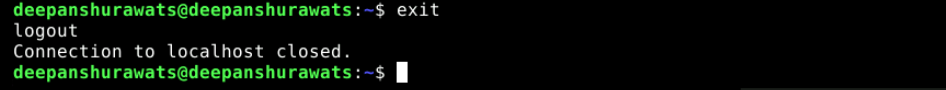
**ssh localhost -p 42006**

Text

Description automatically generated

* 1. Now, use the following command to exit from the localhost:

**exit**



3.4 Run the following command to go to the .ssh directory of the ansible server

**cd .ssh**



3.5 Run the following command to copy the public key to another node that will connect to the ansible server.

**ssh-copy-id username@ip -p 42006**

|  |
| --- |
| **Note**: You will have to replace username@ip with your node’s username and IP. To find your IP address, run the command **ifconfig** on the node whose IP you want to know.  Example: The node that we are connecting with the ansible server has the username **deepanshurawats** and the IP of the node is **40.86.1.9**. We write the above command as **ssh-copy-id** [**deepanshurawats@40.86.1.9**](mailto:deepanshurawats@40.86.1.9) **-p 42006** |

Text

Description automatically generated

3.6 Now, use the following command to come out of the .ssh directory of the ansible server:

**cd**



3.7 Use the following command to check the SSH connection with localhost (if you get a welcome screen as shown in the snippet, then the SSH connection is successfully established)

**ssh username@ip -p 42006**

Text

Description automatically generated

3.8 Now, use the following command to exit from the node 1

**Exit**

**Step 4: Updating the inventory or host file with the host’s IP address**

4.1 Now, add the host localhost in the ansible host file /etc/ansible/hosts

**sudo vi /etc/ansible/hosts**



4.2 When the file opens, add the below two lines of the code at the end of the file:

**[webservers]**

**localhost:42006**

**40.86.1.9:42006**

Text

Description automatically generated

4.3 Press **esc,** then write **:wq** and press **enter** to save the file.

**Step 5: Establish connectivity between the hosts specified in the hosts file and the ansible server**

5.1 Use the following to check whether the connection has been established between the hosts and the ansible server

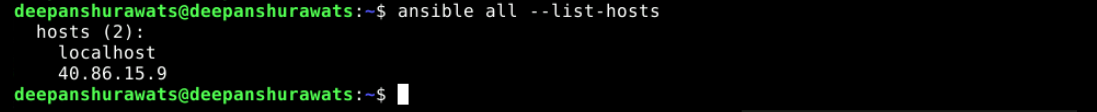
**ansible -m ping webservers**

Text

Description automatically generated

5.2 Use the following command to check the number of hosts in the host file:

**ansible all --list-hosts**



**Step 6: Collecting the uptime of our managed hosts**

6.1 Use the following command to collect the uptime of our managed hosts:

**ansible all -m shell -a uptime**

Text, timeline

Description automatically generated

6.2 Similarly, we can execute:

**ansible all -m shell -a "free -m"**

Text, timeline

Description automatically generated

You will see that Ansible logs in to each machine in turn and runs the uptime command, returning the current uptime output.