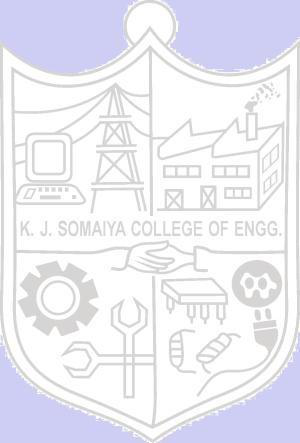


|  |
| --- |
| **Experiment No.: 08**  **Title:** Ansible |

(A Constituent College of Somaiya Vidyavihar University)



|  |  |  |  |
| --- | --- | --- | --- |
| **Batch: B1** | **Roll No.:** | **16010421119** | **Experiment No.:** 8 |

**Aim:** To write an Ansible playbook.



**Resources needed: Virtual Box**



**Theory:**

**Ansible** provides open-source automation that reduces complexity and runs everywhere. Using Ansible lets you automate virtually any task. Here are some common use cases for Ansible: ● Eliminate repetition and simplify workflows  
 ● Manage and maintain system configuration  
 ● Continuously deploy complex software  
 ● Perform zero-downtime rolling updates   
Ansible uses simple, human-readable scripts called playbooks to automate your tasks. You declare the desired state of a local or remote system in your playbook. Ansible ensures that the system remains in that state.

Ansible is designed around the following principles:   
**Agent-less architecture**   
Low maintenance overhead by avoiding the installation of additional software across IT infrastructure.

**Simplicity**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Automation | playbooks | use | straightforward | YAML | syntax | for | code | that | reads | like |

documentation. Ansible is also decentralized, using SSH existing OS credentials to access to remote machines.

**Scalability and flexibility**   
Easily and quickly scale the systems you automate through a modular design that supports a large range of operating systems, cloud platforms, and network devices.

**Idempotence and predictability**   
When the system is in the state your playbook describes Ansible does not change anything, even if the playbook runs multiple times.

**Ansible playbooks**   
Ansible Playbooks offer a repeatable, re-usable, simple configuration management and multi-machine deployment system, one that is well suited to deploying complex applications. If you need to execute a task with Ansible more than once, write a playbook and put it under source control. Then you can use the playbook to push out new configuration or confirm the configuration of remote systems. The playbooks in the [ansible-examples repository](https://github.com/ansible/ansible-examples) illustrate many useful techniques. You may want to look at these in another tab as you read the documentation.

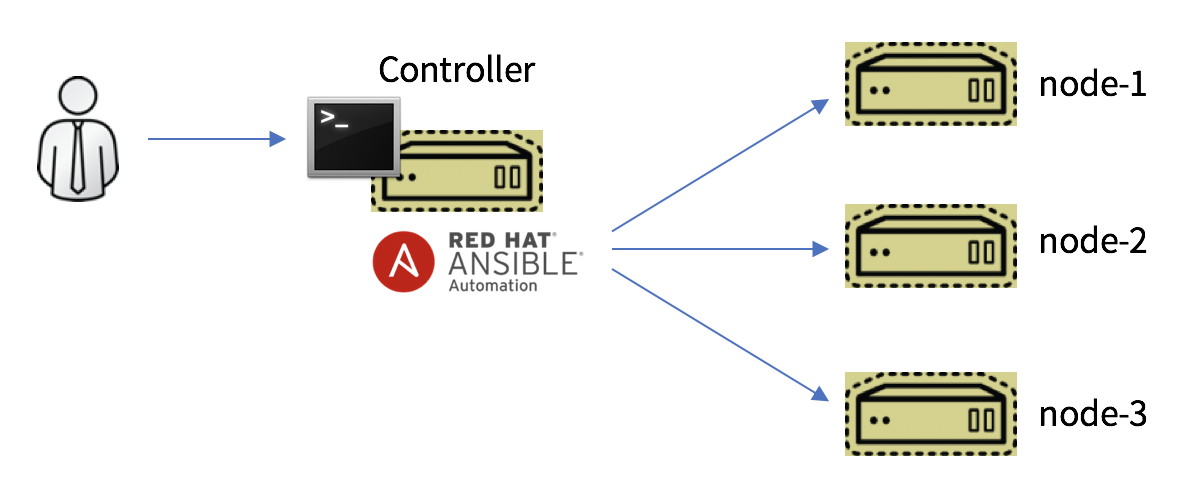


**Procedure:**   
 1. Install Ansible and create node-1, node -2, node -3 sever.

<https://killercoda.com/ansible/scenario/ansible-en-101>

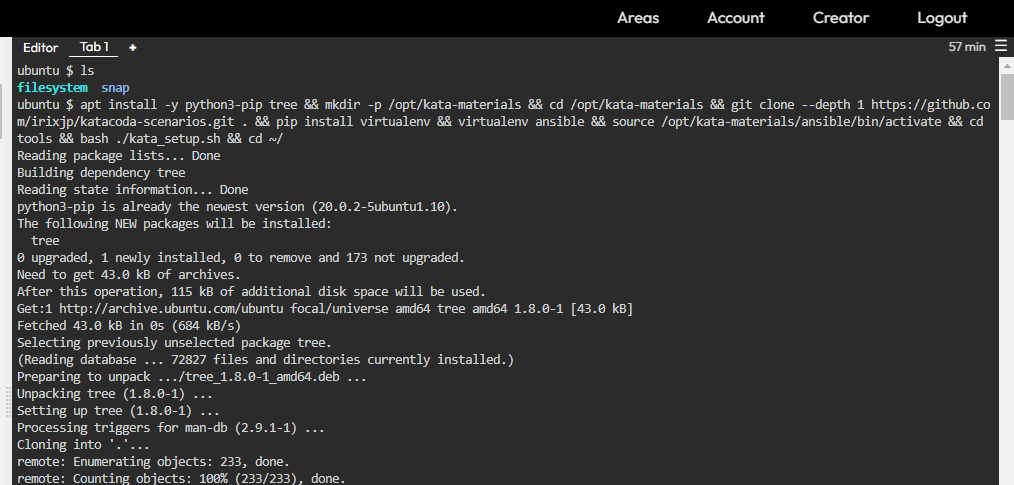
(A Constituent College of Somaiya Vidyavihar University)





**Results: (Document with screenshots)**

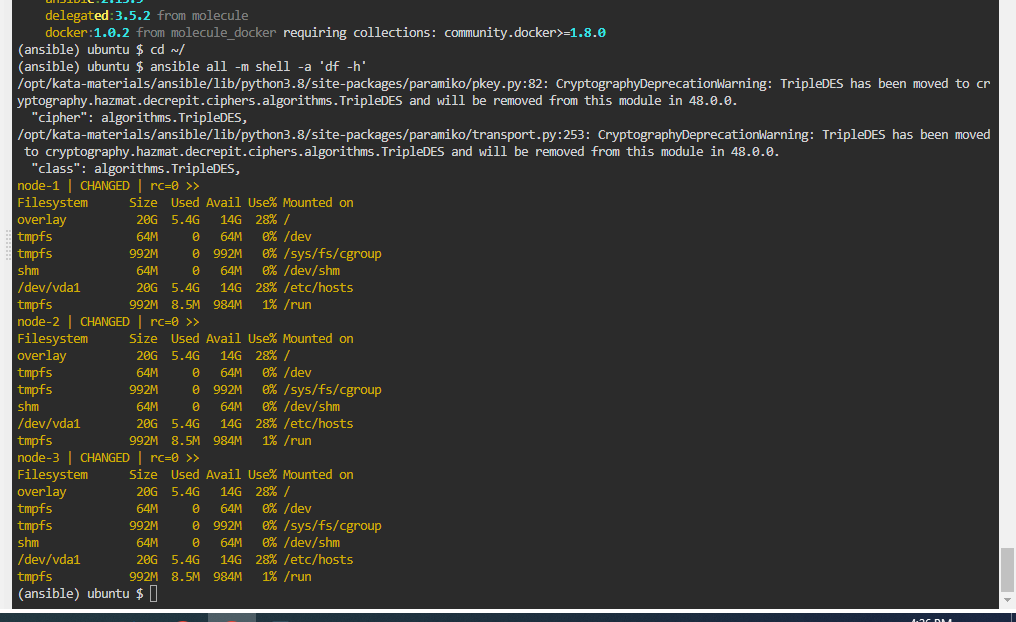
1. Steps with screenshots   
Setup ansible on killerconda



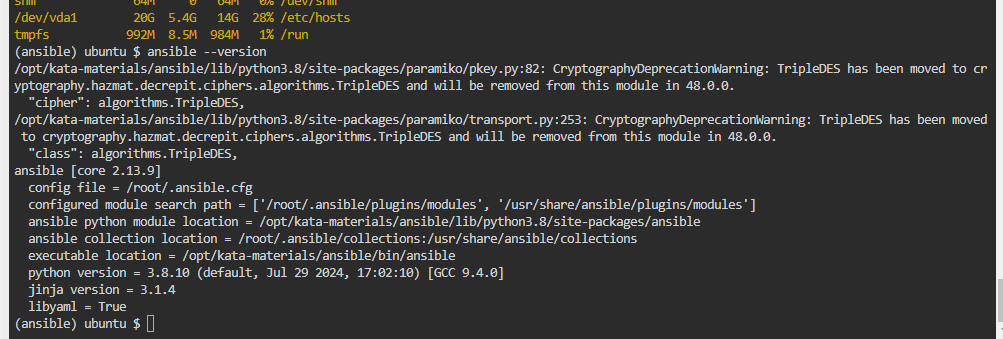
(A Constituent College of Somaiya Vidyavihar University)



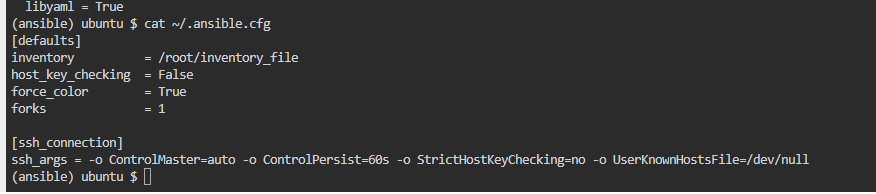
Run shell commands on all nodes using ansible



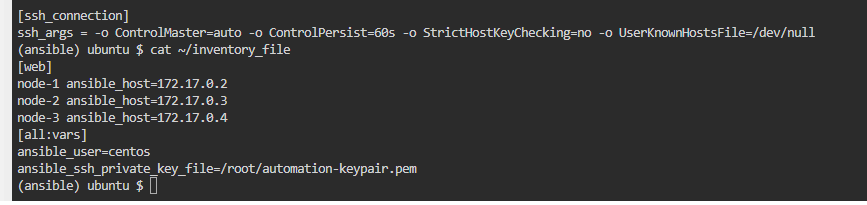
View ansible version



view ansible config



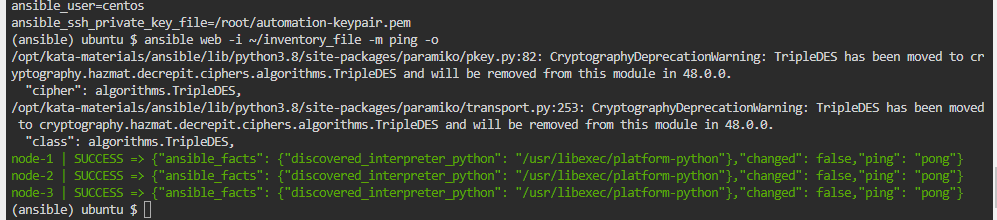
View inventory file of ansible. this also contains



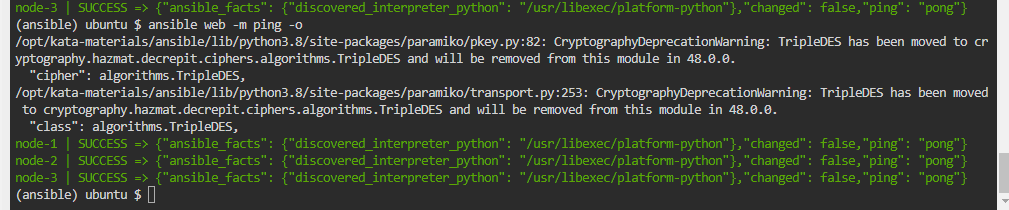
(A Constituent College of Somaiya Vidyavihar University)



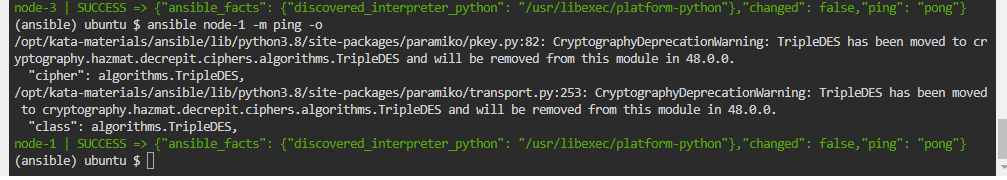
Ping all the nodes present in the web group in inventory file



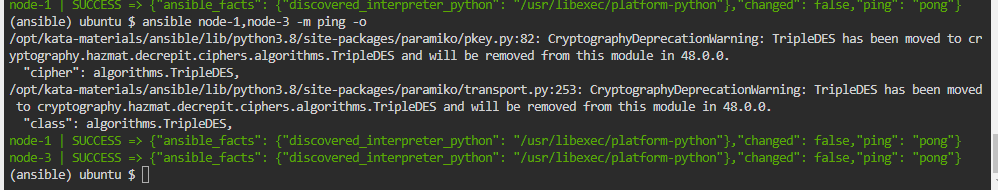
Ping all nodes present in web group without specifying inventory file



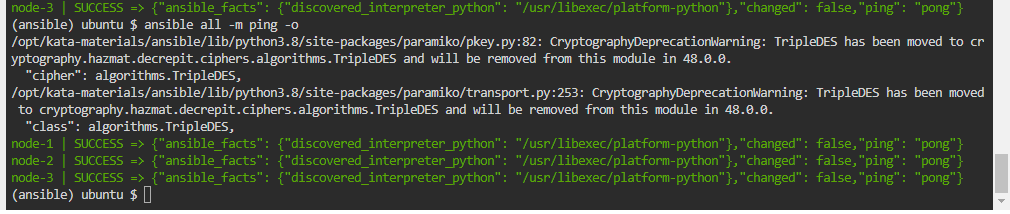
Ping only node-1



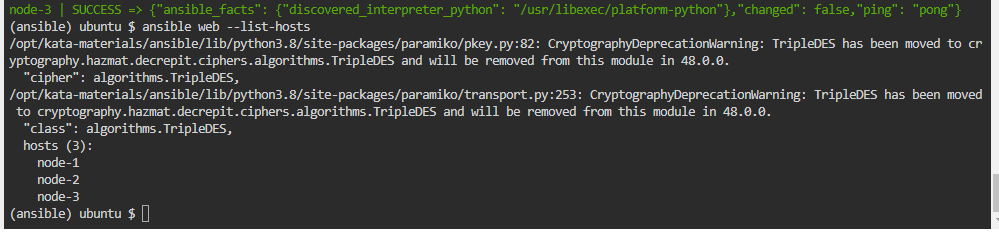
Ping node-1 and node-3



ping all the nodes irrespective of the group



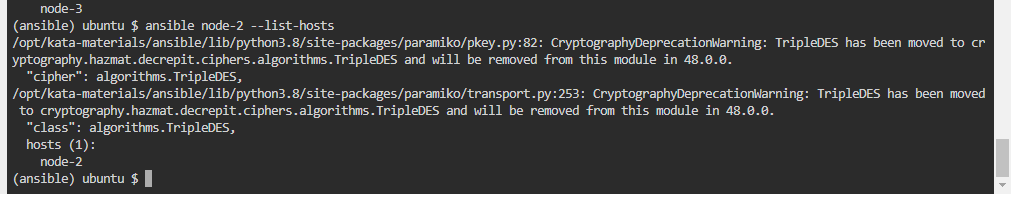
list all hosts in the web group



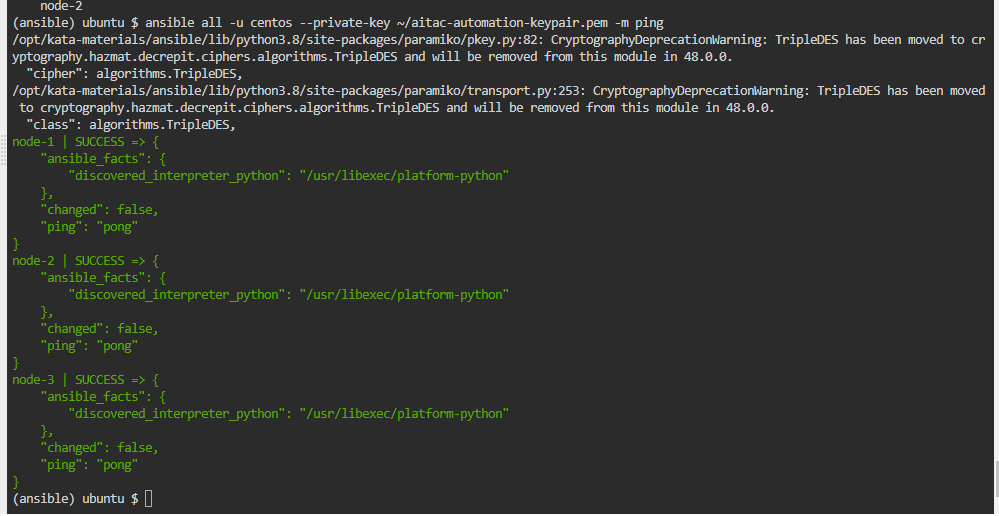
list all nodes in node-2

(A Constituent College of Somaiya Vidyavihar University)

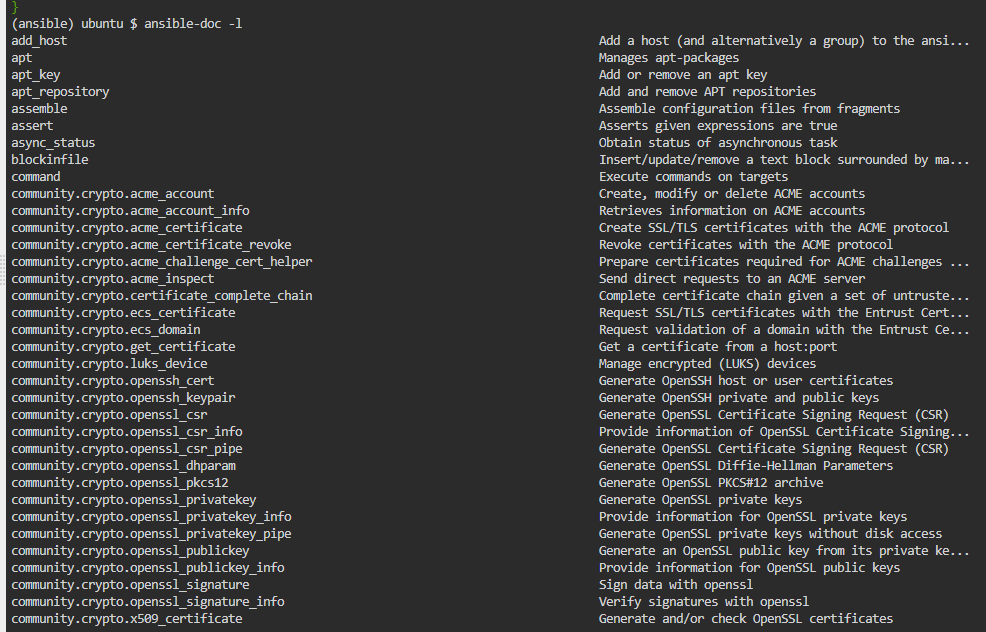




Add authentication key by passing it through ansible CLI



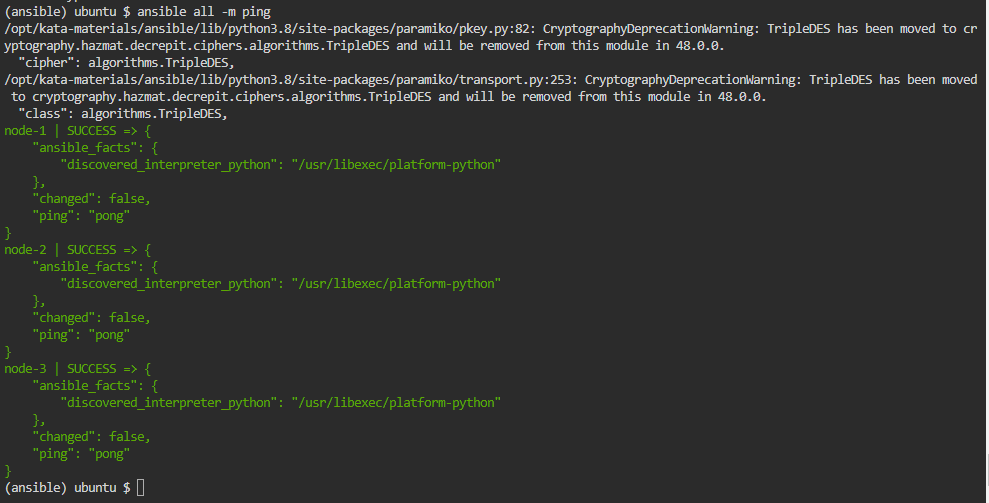
Access the documentation provided by ansible



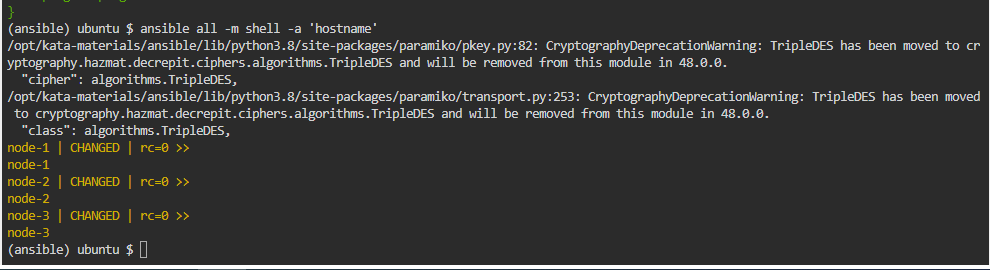
Ping all the nodes

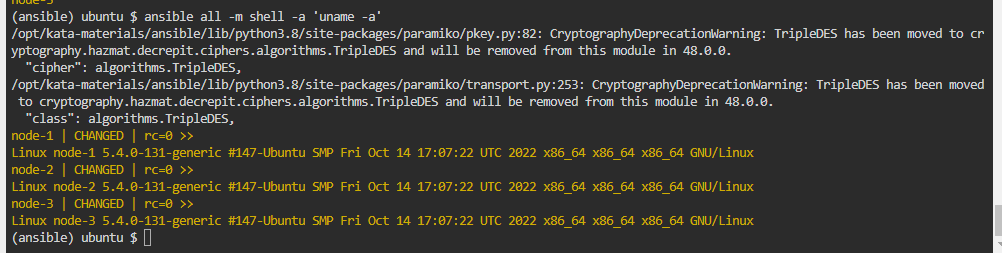
(A Constituent College of Somaiya Vidyavihar University)





Write shell commands in nodes using ansible

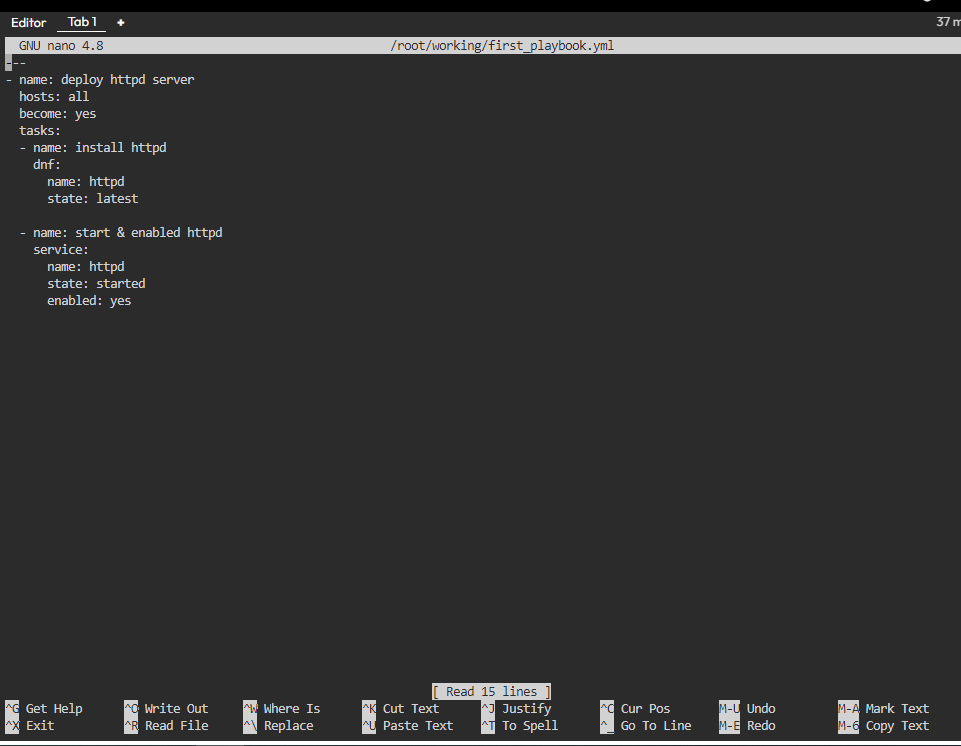




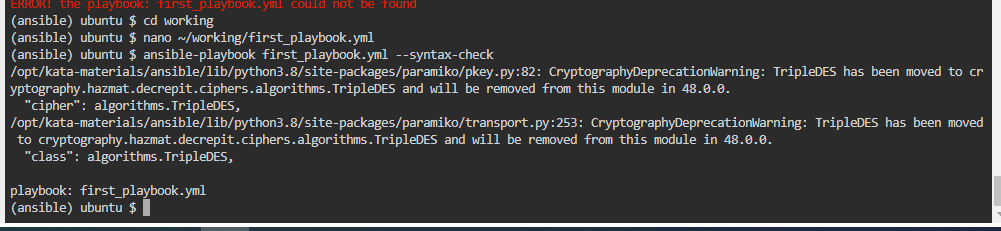
(A Constituent College of Somaiya Vidyavihar University)



write playbook.yml file where in we mention tasks to be run on all the nodes



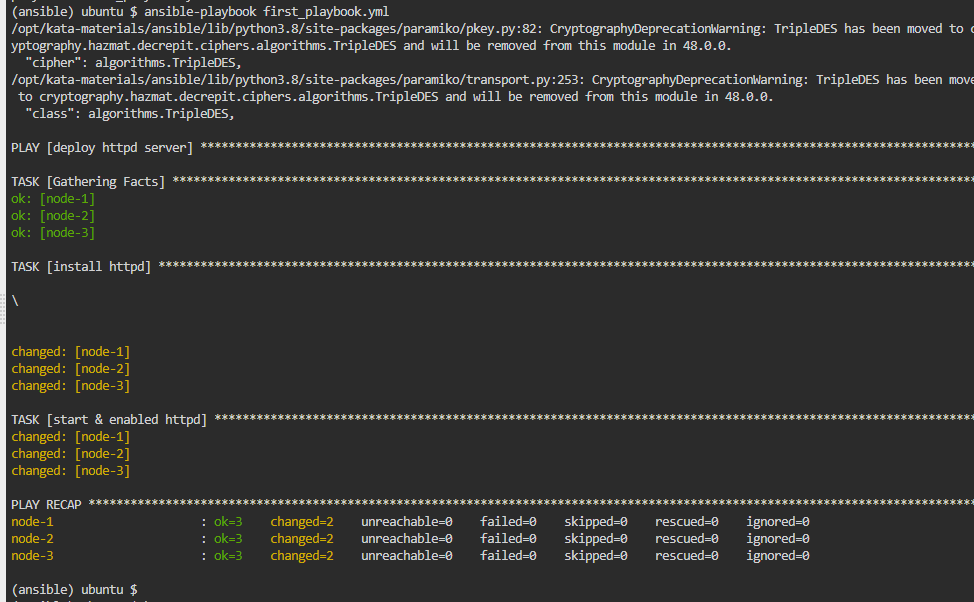
run ansible-playbook first\_playbook.yml –syntax-check to check the syntax of the playbook.



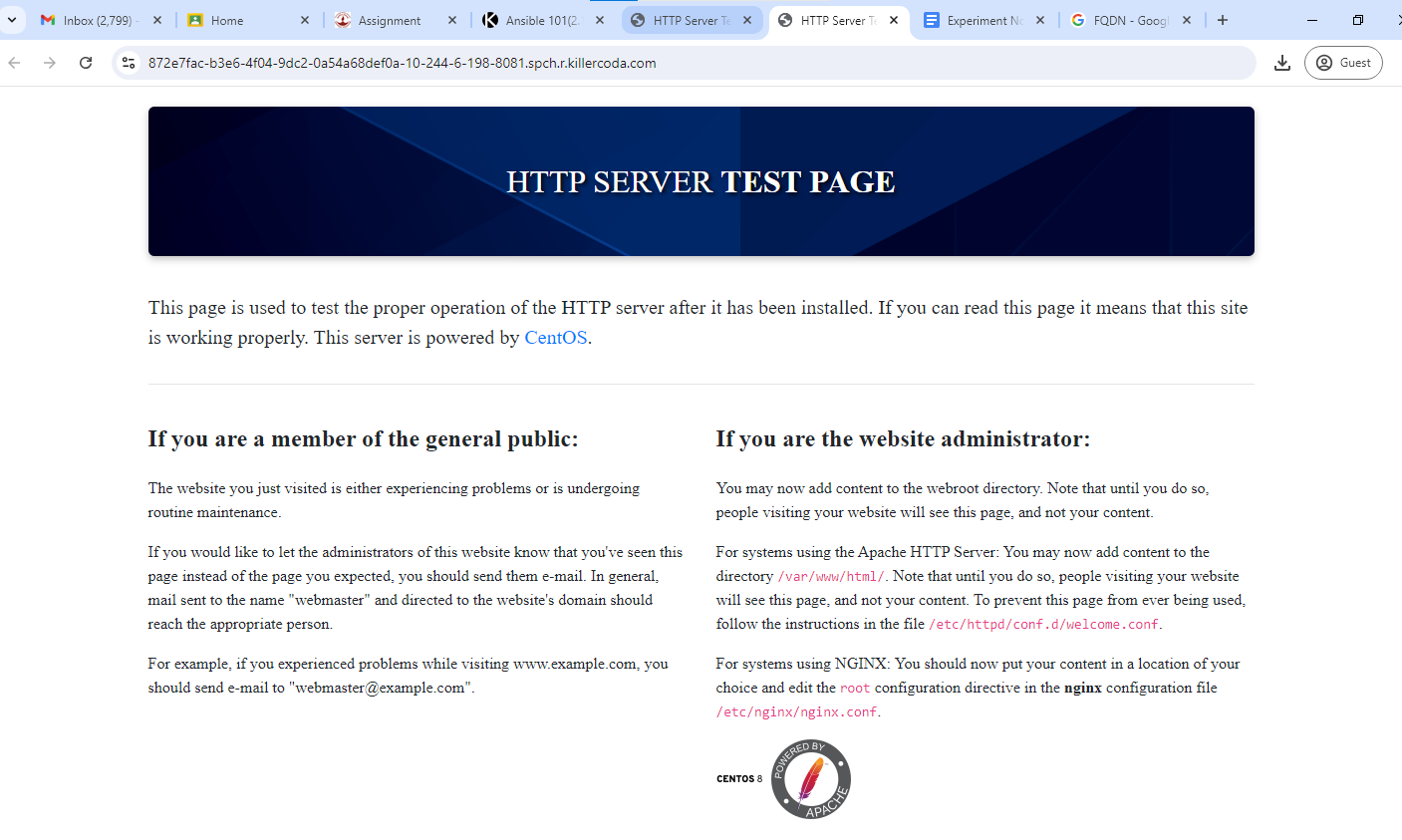
(A Constituent College of Somaiya Vidyavihar University)



run ansible-playbook first\_playbook.yml to run the tasks mentioned in the playbook



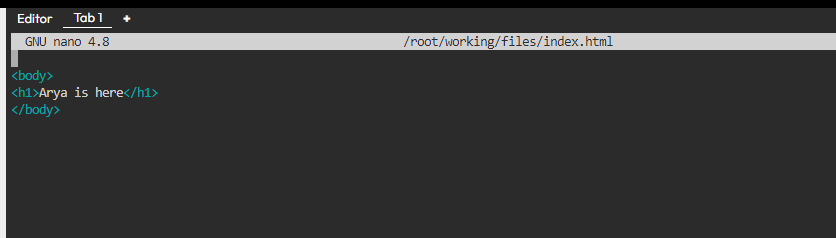
Can see all the hosted nodes using port 80

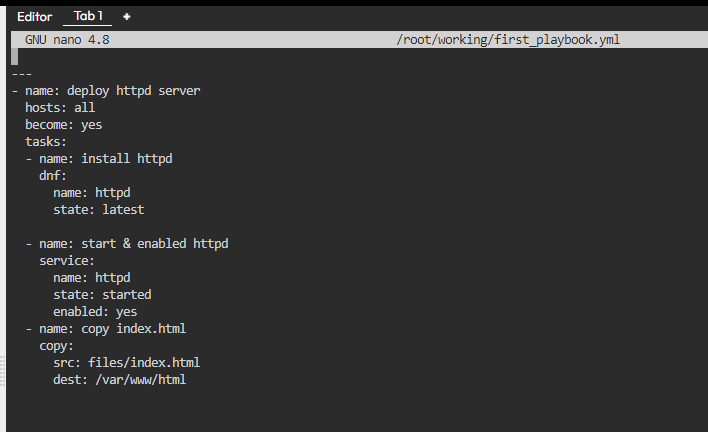


Add copy task wherein we add html code to be displayed on port 80 of all the nodes

(A Constituent College of Somaiya Vidyavihar University)



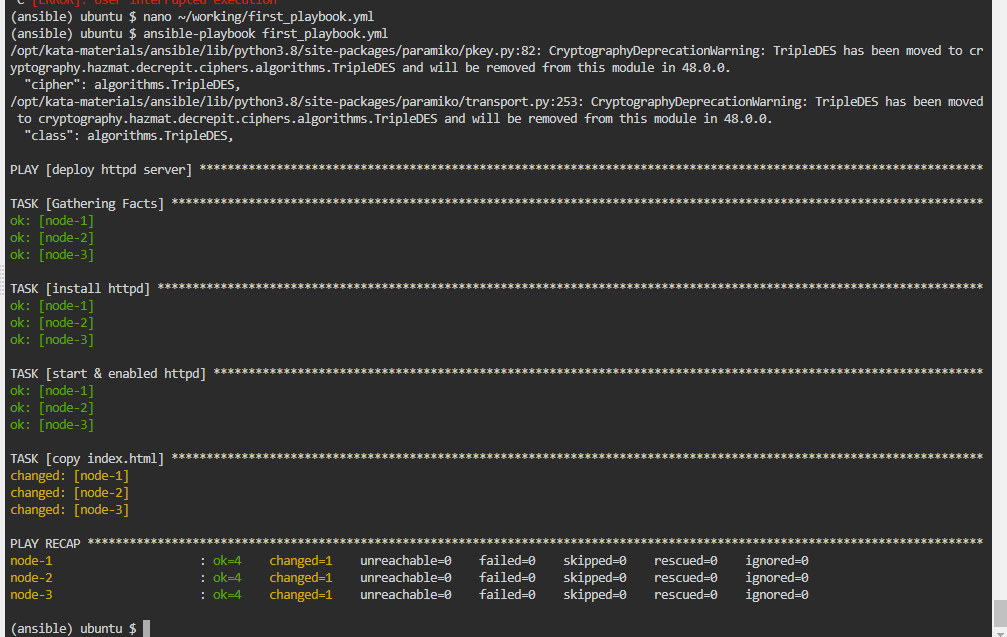




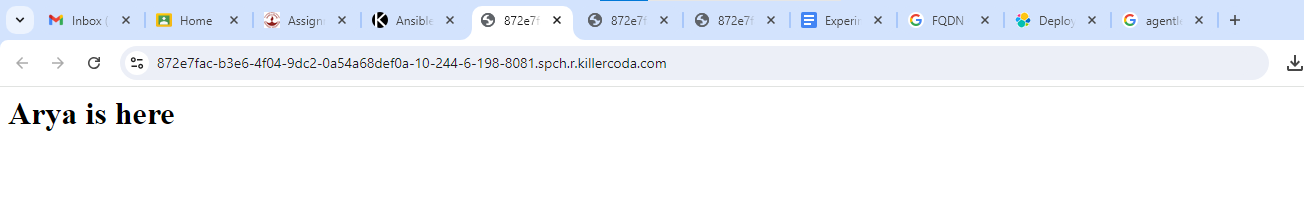
(A Constituent College of Somaiya Vidyavihar University)



Run ansible playbook to run the task



We can see the output of the html file in the browser on all 3 nodes



**Questions:**   
**1. Difference between Ansible vs Puppet vs Chef**   
 Ansible   
 Ansible is a simple, agentless automation tool that uses YAML to define tasks in a readable way. It connects via SSH to manage configurations and ensures that your systems are in the desired state without needing extra software on the nodes.

Puppet   
Puppet uses its own declarative language to manage system configurations and requires an agent on each node. It’s well-suited for large, complex environments and offers strong reporting and automation features.

Chef   
Chef uses Ruby-based scripts (cookbooks) for configuration management and requires an agent on managed nodes. It’s highly customizable and is ideal for complex setups, with a focus on flexibility and detailed configuration.

**2. Explain agent less and agent base with examples.**   
Agentless   
Definition:

(A Constituent College of Somaiya Vidyavihar University)



An agentless tool doesn’t require any special software or agents to be installed on the managed nodes. It typically uses standard protocols like SSH or WinRM to communicate directly with the systems.

Example:

Ansible is an agentless tool. It uses SSH to connect to Linux servers and WinRM for Windows servers to execute commands and apply configurations. Since there’s no need for an agent on the managed servers, setup is simpler, and there’s less overhead.

Agent-Based   
Definition:   
An agent-based tool requires a specific agent software to be installed on each managed node.

This agent communicates with a central server to receive and apply configuration changes.

Example:

Puppet and Chef are agent-based tools. Puppet’s agent runs on each node, periodically checking in with the central Puppet server to receive updates and apply configurations. Similarly, Chef’s agent (called the Chef Client) performs regular checks with the Chef server to manage system states and updates.



**Outcomes:**   
CO3: Comprehend the effective code building and testing process.



**Conclusion: (Conclusion to be based on the Results and outcomes achieved)**

Use ansible(IAC) to run tasks on the existing nodes present with us.

**Grade: AA / AB / BB / BC / CC / CD /DD**   
**Signature of faculty in-charge with date**

**References:**   
**Websites:**

1. Ansible Full Course | Ansible Tutorial For Beginners | Learn Ansible Step By Step |

Simplilearn

<https://www.youtube.com/watch?v=EcnqJbxBcM0&t=1342s> **till time stamp 9:10**

2. How to Install and Configure Ansible on Windows

<https://phoenixnap.com/kb/install-ansible-on-windows>

3. Ansible Hello World

<https://www.youtube.com/watch?v=vyiJrTHXE0M>

|  |
| --- |
| 4. Using Ansible playbooks |

<https://docs.ansible.com/ansible/latest/playbook_guide/index.html>

(A Constituent College of Somaiya Vidyavihar University)