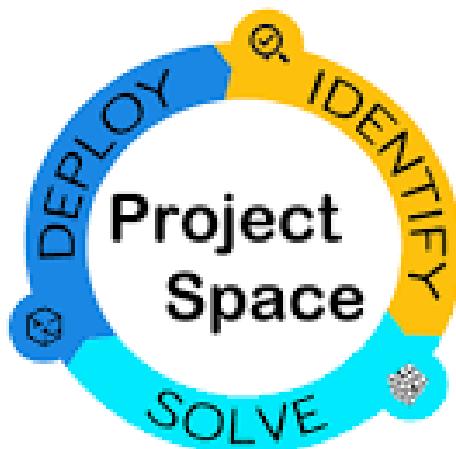


PROJECT CONFIGURATION MANAGEMENT WITH ANSIBLE



Team Name: Ansible Army

Team Number: 06

Team Members:

- B. Yamuna Devi
- Ch. Blessy
- K. Sahasra
- T. G. B. S. Iswarya

INTRODUCTION

Ansible is a modern configuration management tool that provides simple but powerful automation for cross-platform computer support. It is used in IT industry for application deployment, updates on systems and servers, cloud provisioning, orchestration and nearly everything a system administrator does on a daily basis. Configuration management is the process of handling changes to system in a way that assures integrity over time, typically involving tools and processes that facilitate automation and observability.

Ansible helps us to quickly and easily deploy multiple apps. You won't need to handle each and every system for deploying or updating something you can easily create a master server and connect all other client servers to it using ansible than changes made on master server will be reflected on all other servers and this is possible by writing the ansible playbook command on the main server.

Ansible is an open-source software tool enabling infrastructure as code, Ansible was written by Michael DeHaan, the author of the provisioning server application Coder and co-author of the Fedora Unified Network Controller (FUNC) framework for Remote administration.

Ansible is included as part of the Fedora distribution of Linux, owned by Red Hat in October 2015, and is also available for Red Hat Enterprise Linux, CentOS, openSUSE, SUSE Linux Enterprise, Debian, Ubuntu, Scientific Linux, and Oracle Linux via Extra Packages for Enterprise Linux (EPEL), as well as for other operating systems. Ansible is easy to deploy because it does not use any agents or custom security infrastructure. Ansible uses playbook to describe automation jobs, and playbook uses very simple language i.e. YAML

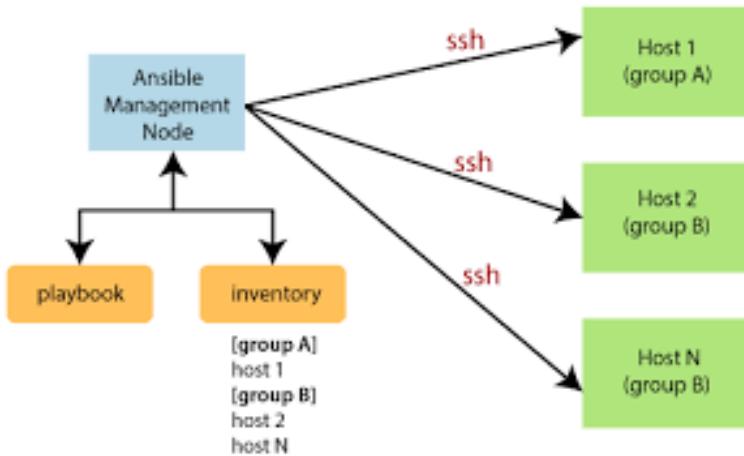
YAML (yet another markup language or YAML ain't markup language)-It is human-readable data serialization language that is often used for writing configuration files. Because of its flexibility and accessibility, YAML is used by the Ansible automation tool to create automation processes, in the form of Ansible Playbooks.

Ansible is designed for multi-tier deployment. Ansible is completely agentless which means Ansible works by connecting your nodes through ssh (by default).

In Ansible, there are two categories of computers: the control node and managed nodes. The control node is a computer that runs Ansible. There must be at least one control node, although a backup control node may also exist. A managed node is any device being managed by the control node.

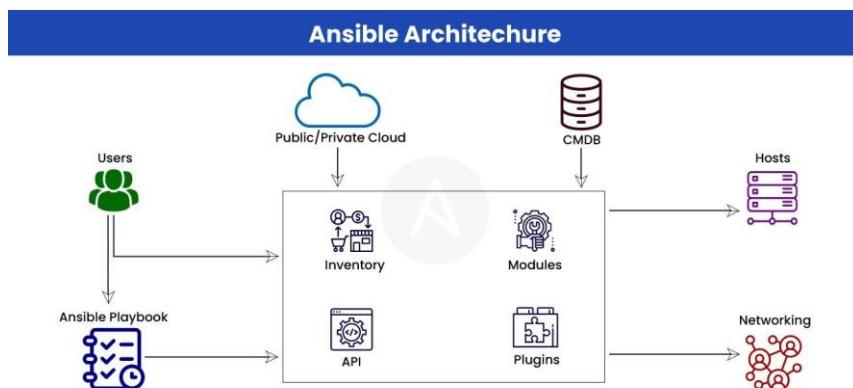
Ansible Workflow:

Ansible works by connecting to your nodes and pushing out a small program called **Ansible modules** to them. Then Ansible executes these modules and removes them after finished. The library of modules can reside on any machine, and there are no daemons, **servers**, or **databases** required.



Ansible Architecture:

The Ansible orchestration engine interacts with a user who is writing the Ansible playbook to execute the Ansible orchestration and interact along with the services of private or public cloud and configuration management database. You can show in the below diagram, such as:



Inventory: Inventory is lists of nodes or hosts having their IP addresses, databases, servers, etc. which are need to be managed.

API's: The Ansible API's works as the transport for the public or private cloud services.

Modules: Ansible connected the nodes and spread out the Ansible modules programs. Ansible executes the modules and removed after finished.

Plugins: Plugins is a piece of code that expends the core functionality of Ansible.

Playbooks: Playbooks consist of your written code, and they are written in YAML format, which describes the tasks and executes through the Ansible.

Hosts: In the Ansible architecture, hosts are the node systems, which are automated by Ansible, and any machine such as RedHat, Linux, Windows, etc.

Networking: Ansible is used to automate different networks, and it uses the simple, secure, and powerful agentless automation framework for IT operations and development.

Cloud: A cloud is a network of remote servers on which you can store, manage, and process the data. These servers are hosted on the internet and storing the data remotely rather than the local server.

CMDB: CMDB is a type of repository which acts as a data warehouse for the IT installations.

Ansible Operations:

1. Configuration Management: Configuration management in terms of Ansible means that it maintains configuration of the product performance by keeping a record and updating detailed information which describes an enterprise's hardware and software.

2. Orchestration: Orchestration is the coordination and management of multiple computer systems, applications and/or services, stringing together multiple tasks in order to execute a larger workflow or process.

3. Infrastructure provisioning: Ansible replaces ad hoc scripting or manual infrastructure management with an automated and repeatable process. Since everything is written in simple script form, it enables organizations to adopt infrastructure as code (IaC).

4. Application deployment: Ansible provides a common framework to configure services; deploy applications, including multi-tier applications; and push application artifacts easily, reliably and consistently. There's no need to write custom code to automate app deployment. Instead, teams only need to write simple task descriptions in YAML files that are easy to read, implement and update as required.

Benefits of Ansible:

- Rich and visual insights to identify, troubleshoot and resolve operational and security issues across the IT ecosystem.
- A collaborative and scalable approach to enterprise automation.
- Infrastructure codification for both on-premises and cloud deployments.
- Easy to transfer automation into multiple domains and across a range of use cases.
- Analytics, policy and governance, and content management tools to streamline and scale automation.

SCOPE OF THE PROJECT

The scope of the project includes assessing the infrastructure, installing and configuring Ansible, developing playbooks to create configuration management policies, managing inventory, implementing continuous configuration monitoring, developing compliance reports, integrating with other tools.

Overall, it involves using Ansible as a configuration management tool to ensure that the infrastructure remains in compliance with defined policies and procedures. The specific tasks involved in the project may vary depending on the organization's requirements and the complexity of the infrastructure being managed. It also depends on the configurations of the hosts like operating systems.

PURPOSE OF THE PROJECT

The purpose of the project configuration management with Ansible is to provide an automated and standardized approach to managing an organization's IT infrastructure. By using Ansible, the project aims to reduce the complexity of managing and deploying configurations across a large number of systems.

The project focuses on simplifying the configuration management process by automating tasks that would otherwise be time-consuming and prone to errors if performed manually. This helps to ensure that configurations remain consistent, up-to-date, and compliant with defined policies and procedures, reducing the risk of security breaches, misconfigurations, and downtime.

MAJOR TOOLS UTILIZED IN THIS PROJECT

This project includes the utilization of various tools like

Ansible: This is the main tool you will use for managing configurations across multiple systems.

Inventory file: The Ansible inventory file defines the hosts and groups of hosts upon which commands, modules, and tasks in a playbook operate. The file can be in one of many formats depending on your Ansible environment and plugins. Common formats include INI and YAML.

Playbook: 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.

Modules: These are pre-built components in Ansible that perform specific tasks, such as managing packages or files, creating users, and executing shell commands. Modules are the main building blocks of Ansible playbooks. Although we do not generally speak of “module plugins”, a module is a type of plugin.

Roles: These are collections of tasks and variables that can be reused across multiple playbooks. They are a way to organize tasks and variables in a structured and reusable manner.

TECHNOLOGIES USED IN THIS PROJECT

Ansible: Ansible is a software tool that provides simple but powerful automation for cross-platform computer support. It is primarily intended for IT professionals, who use it for application deployment, updates on workstations and servers, cloud provisioning, configuration management, intra-service orchestration, and nearly anything a systems administrator does on a weekly or daily basis. Ansible doesn't depend on agent software and has no additional security infrastructure, so it's easy to deploy.

Python: Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically-typed and garbage-collected programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is very high in demand and all the major companies are looking for great Python Programmers to develop websites, software components, and applications or to work with Data Science, AI, and ML technologies. When we are developing this tutorial in 2022, there is a high shortage of Python Programmers whereas market demands a greater number of Python Programmers due to its application in Machine Learning, Artificial Intelligence etc.

AWS: Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud, offering over 200 fully featured services from data centres globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

Docker: Docker is an open-source platform that enables developers to build, deploy, run, update and manage containers—standardized, executable components that combine application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify development and delivery of distributed applications. They have become increasingly popular as organizations shift to cloud-native development and hybrid multicloud environments. It's possible for developers to create containers without Docker, by working directly with capabilities built into Linux and other operating systems. But Docker makes containerization faster, easier and safer. At this writing, Docker reported over 13 million developers using the platform.

JAVA: Java is a widely-used programming language for coding web applications. It has been a popular choice among developers for over two decades, with millions of Java applications in use today. Java is a multi-platform, object-oriented, and network-centric language that can be used as a platform in itself. It is a fast, secure, reliable programming language for coding everything from mobile apps and enterprise software to big data applications and server-side technologies.

Java is a widely used object-oriented programming language and software platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others. The rules and syntax of Java are based on the C and C++ languages.

One major advantage of developing software with Java is its portability. Once you have written code for a Java program on a notebook computer, it is very easy to move the code to a mobile device. When the language was invented in 1991 by James Gosling of Sun Microsystems (later acquired by Oracle), the primary goal was to be able to "write once, run anywhere."

GIT: Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development. Linus Torvalds created Git in 2005 for the development of the Linux kernel. Having a distributed architecture, Git is an example of a DVCS (hence Distributed Version Control System). Rather than have only one single place for the full version history of the software as is common in once-popular version control systems like CVS or Subversion (also known as SVN), in Git, every developer's working copy of the code is also a repository that can contain the full history of all changes.

PROOF OF CONCEPT

Step 1: Launching AWS instances

Firstly, Launch an AWS Linux instances for the master node.g

Step 1: Set up an EC2 instance

Step 2: Choosing an Amazon Machine Image as **Amazon Linux2 AMI**

Step 3: Choosing an instance type

Step 4: Configure Instance

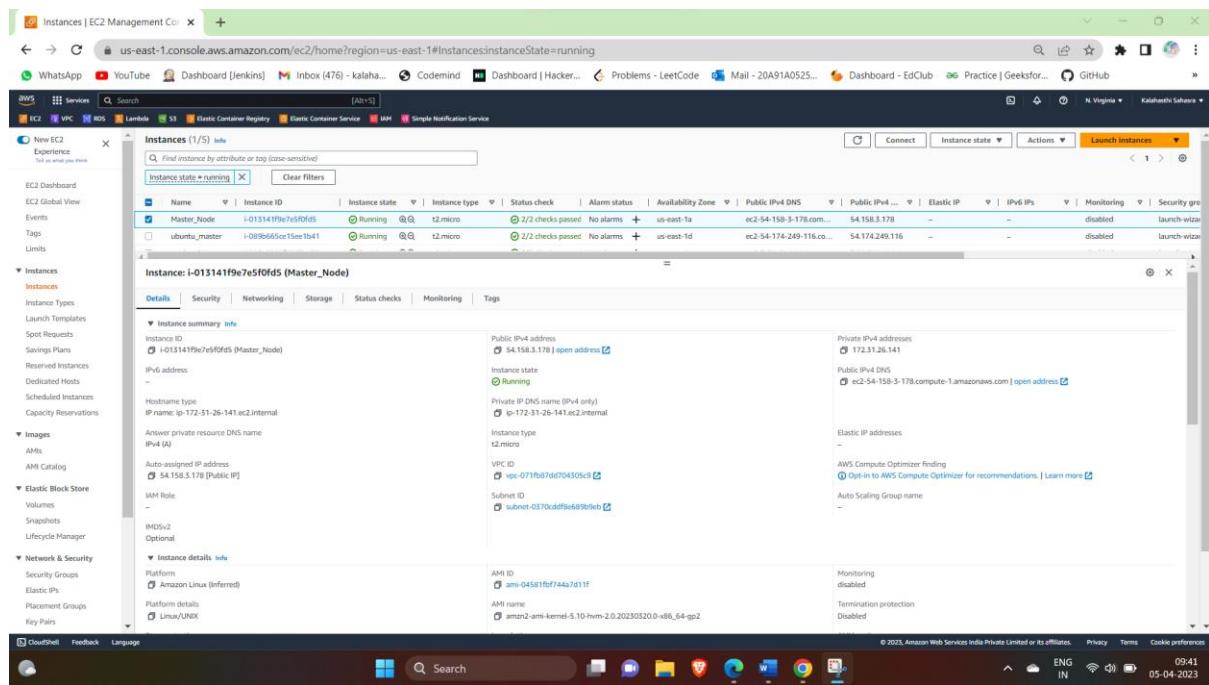
Step 5: Adding Storage

Step 6: Adding tags

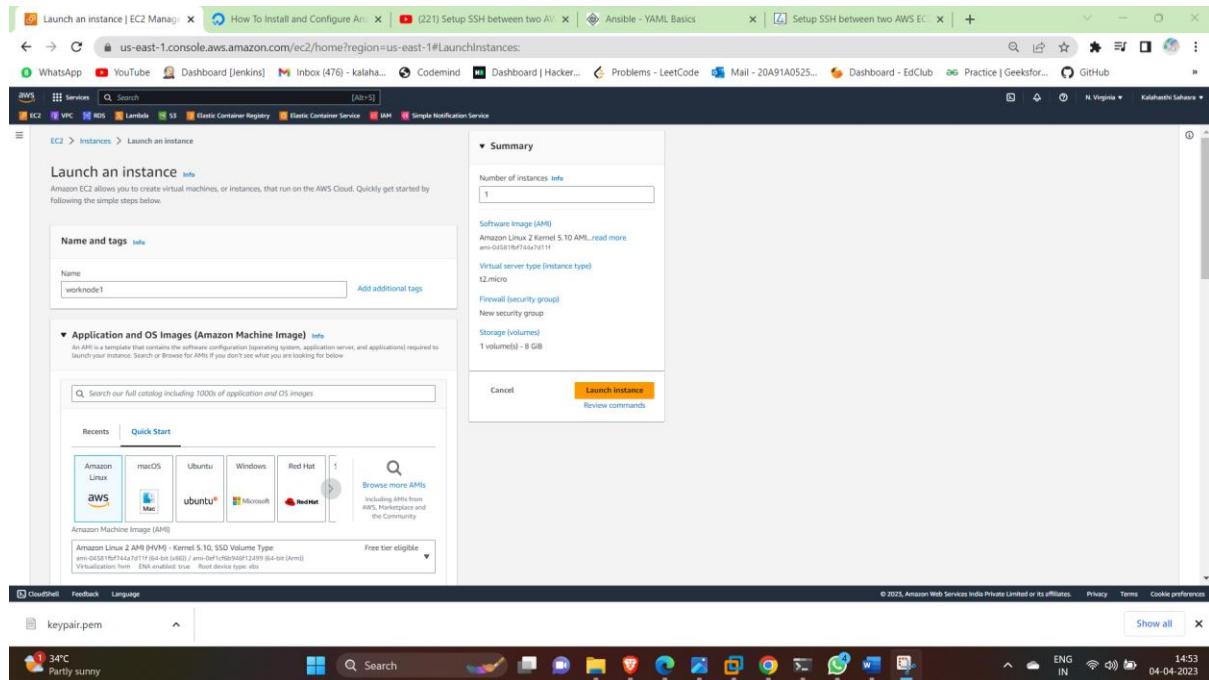
Step 7: Configuring security groups

Step 9: Review

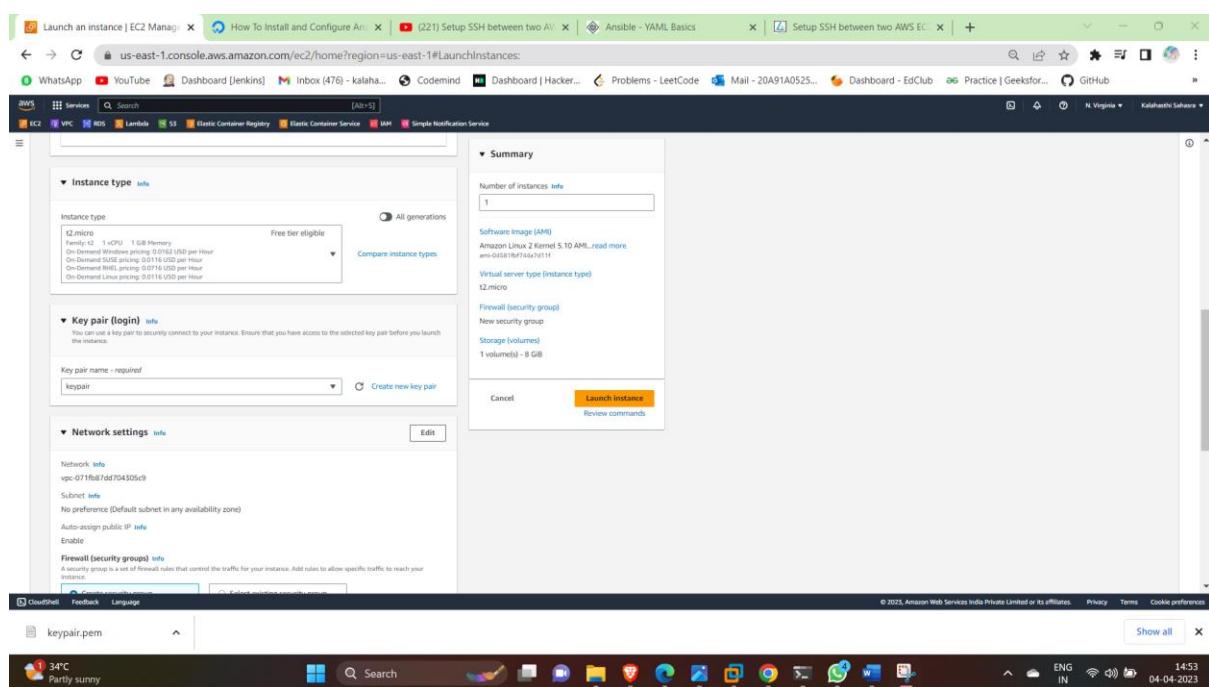
Click on 'Launch' and the instance is created.

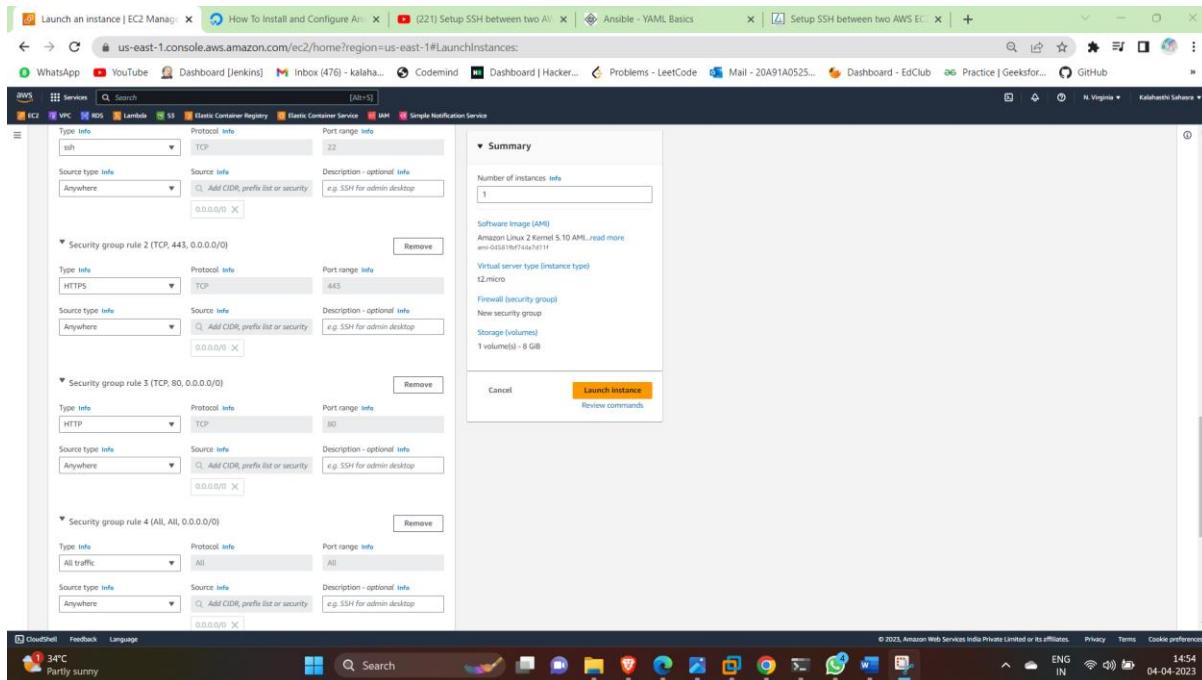


Now, repeat the same process and launch another Linux instance for the work node/slave node.

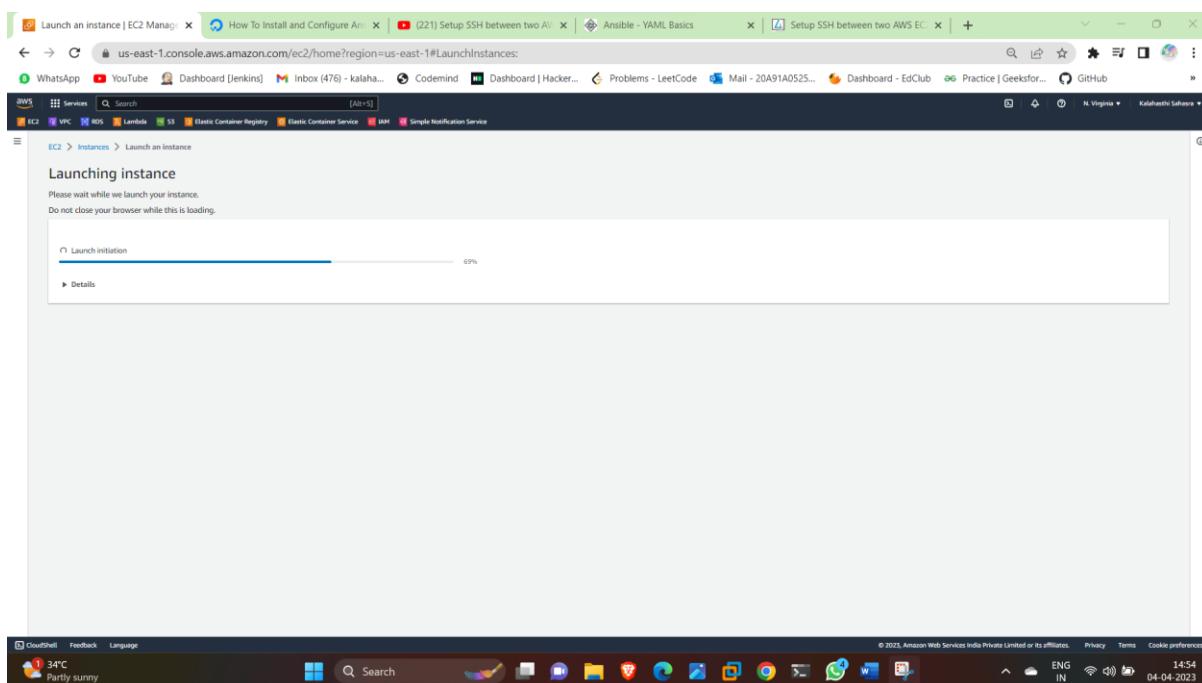


Now, select the keypair and instance type.





Now, launch the instance.



Successfully, launched the two nodes master and work/slay node.

The screenshot shows the AWS Management Console with the EC2 service selected. The main pane displays a table of instances with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Monitoring	Security group
Master_Node	i-013141f9e7e5f0fd5	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-158-3-178.compute-1.amazonaws.com	54.158.3.178	-	-	disabled	launch-wizard-1
worknode1	i-026e8092f9217ec54	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-91-244-114.compute-1.amazonaws.com	54.91.244.114	-	-	disabled	launch-wizard-1

Below the table, there is a monitoring section with several time-series charts showing CPU utilization, network traffic, disk I/O, and network packets over a 1-hour period.

Step 2: Now connect the master node using the ssh on local machine.

The screenshot shows the 'Connect to instance' dialog box for the Master_Node instance. The 'SSH client' tab is selected. The 'Command copied' field contains the command:

```
ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
```

A note at the bottom of the dialog box states: "Notes: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name."

Now update the system using the command: "sudo yum update -y"

```

C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sahasra>cd downloads

C:\Users\Sahasra\Downloads>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Sahasra\Downloads>ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
The authenticity of host 'ec2-54-158-3-178.compute-1.amazonaws.com (54.158.3.178)' can't be established.
ED25519 key fingerprint is SHA256:c/FzER1HMaXjdrRXYBHNZtzBun3s9LdrE0OkkgA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-158-3-178.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

--| _ |_
-| (   /   Amazon Linux 2 AMI
--\_|_||_|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-26-141 ~]$ sudo yum install -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Error: Need to pass a list of pkgs to install
  Mini usage:

install PACKAGE...

Install a package or packages on your system

aliases: install-n, install-na, install-nevra
[ec2-user@ip-172-31-26-141 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update

```

Step 3: Now install ansible using the command: “sudo amazon-linux extras install ansible -y”

```

mini usage.

install PACKAGE...

Install a package or packages on your system

aliases: install-n, install-na, install-nevra
[ec2-user@ip-172-31-26-141 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-26-141 ~]$ sudo amazon-linux-extras install ansible2 -y
Installing ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-ansible2 amzn2extra-docker amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed

```

```

C:\WINDOWS\system32\cmd. x + v
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-ansible2
amzn2extra-docker
amzn2extra-kernel-5.10
(1/9): amzn2-core/2/x86_64/group_gz | 3.7 kB 00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo | 3.0 kB 00:00:00
(3/9): amzn2extra-docker/2/x86_64/primary_db | 3.0 kB 00:00:00
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 3.0 kB 00:00:00
(5/9): amzn2extra-ansible2/2/x86_64/primary_db | 2.5 kB 00:00:00
(6/9): amzn2extra-ansible2/2/x86_64/primary_db | 583 kB 00:00:00
(7/9): amzn2extra-docker/2/x86_64/updateinfo | 103 kB 00:00:00
(8/9): amzn2extra-kernel-5.10/2/x86_64/primary_db | 26 kB 00:00:00
(9/9): amzn2-core/2/x86_64/primary_db | 76 B 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package ansible.noarch 0:2.9.23-1.amzn2 will be installed
--> Processing Dependency: sshpass for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-paramiko for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-keyczar for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-httplib2 for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-crypto for package: ansible-2.9.23-1.amzn2.noarch
--> Running transaction check
--> Package python-keyczar.noarch 0:0.71c-2.amzn2 will be installed
--> Package python2-crypto.x86_64 0:2.6.1-13.amzn2.0.3 will be installed
--> Processing Dependency: libtomcrypt.so.1()(64bit) for package: python2-crypto-2.6.1-13.amzn2.0.3.x86_64
--> Package python2-httplib2.noarch 0:0.18.1-3.amzn2 will be installed
--> Package python2-paramiko.noarch 0:1.16.1-3.amzn2.0.3 will be installed
--> Processing Dependency: python2-ecdsa for package: python2-paramiko-1.16.1-3.amzn2.0.3.noarch
--> Package sshpass.x86_64 0:1.06-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package libtomcrypt.x86_64 0:1.18.2-1.amzn2.0.1 will be installed
--> Processing Dependency: libtommath >= 1.0 for package: libtomcrypt-1.18.2-1.amzn2.0.1.x86_64
--> Processing Dependency: libtommath.so.1()(64bit) for package: libtomcrypt-1.18.2-1.amzn2.0.1.x86_64
--> Package python2-ecdsa.noarch 0:0.13.3-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package libtommath.x86_64 0:1.0.1-4.amzn2.0.1 will be installed

```

34°C Partly sunny Search Start File Help ENG IN 15:28 04-04-2023

```

C:\WINDOWS\system32\cmd. x + v
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
amzn2extra-ansible2 | 3.0 kB 00:00:00
amzn2extra-docker | 3.0 kB 00:00:00
amzn2extra-kernel-5.10 | 3.0 kB 00:00:00
(1/9): amzn2-core/2/x86_64/group_gz | 2.5 kB 00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo | 583 kB 00:00:00
(3/9): amzn2extra-docker/2/x86_64/primary_db | 103 kB 00:00:00
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 26 kB 00:00:00
(5/9): amzn2extra-ansible2/2/x86_64/updateinfo | 78 B 00:00:00
(6/9): amzn2extra-ansible2/2/x86_64/primary_db | 40 kB 00:00:00
(7/9): amzn2extra-docker/2/x86_64/updateinfo | 8.0 kB 00:00:00
(8/9): amzn2extra-kernel-5.10/2/x86_64/primary_db | 16 MB 00:00:00
(9/9): amzn2-core/2/x86_64/primary_db | 70 MB 00:00:01

Resolving Dependencies
--> Running transaction check
--> Package ansible.noarch 0:2.9.23-1.amzn2 will be installed
--> Processing Dependency: sshpass for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-paramiko for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-keyczar for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-httplib2 for package: ansible-2.9.23-1.amzn2.noarch
--> Processing Dependency: python-crypto for package: ansible-2.9.23-1.amzn2.noarch
--> Running transaction check
--> Package python-keyczar.noarch 0:0.7lc-2.amzn2 will be installed
--> Package python2-crypto.x86_64 0:2.6.1-13.amzn2.0.3 will be installed
--> Processing Dependency: libtomcrypt.so.1()(64bit) for package: python2-crypto-2.6.1-13.amzn2.0.3.x86_64
--> Package python2-httplib2.noarch 0:0.18.1-3.amzn2 will be installed
--> Package python2-paramiko.noarch 0:1.16.1-3.amzn2.0.3 will be installed
--> Processing Dependency: python2-ecdsa for package: python2-paramiko-1.16.1-3.amzn2.0.3.noarch
--> Package sshpass.x86_64 0:1.06-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package libtomcrypt.x86_64 0:1.18.2-1.amzn2.0.1 will be installed
--> Processing Dependency: libtommath.so.1()(64bit) for package: libtomcrypt-1.18.2-1.amzn2.0.1.x86_64
--> Processing Dependency: libtommath.so.1()(64bit) for package: libtomcrypt-1.18.2-1.amzn2.0.1.x86_64
--> Package python2-ecdsa.noarch 0:0.13.3-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package libtommath.x86_64 0:1.0.1-4.amzn2.0.1 will be installed

34°C Partly sunny 15:28 ENG IN 04-04-2023

```

```

C:\WINDOWS\system32\cmd. x + v
Transaction test succeeded
Running transaction
Installing : sshpass-1.06-1.amzn2.0.1.x86_64 1/9
Installing : python2-httplib2-0.18.1-3.amzn2.noarch 2/9
Installing : libtommath-1.0.1-4.amzn2.0.1.x86_64 3/9
Installing : libtomcrypt-1.18.2-1.amzn2.0.1.x86_64 4/9
Installing : python2-crypto-2.6.1-13.amzn2.0.3.x86_64 5/9
Installing : python-keyczar-0.7lc-2.amzn2.noarch 6/9
Installing : python2-ecdsa-0.13.3-1.amzn2.0.1.noarch 7/9
Installing : python2-paramiko-1.16.1-3.amzn2.0.3.noarch 8/9
Installing : ansible-2.9.23-1.amzn2.noarch 9/9
Verifying : python2-ecdsa-0.13.3-1.amzn2.0.1.noarch 1/9
Verifying : libtommath-1.0.1-4.amzn2.0.1.x86_64 2/9
Verifying : libtomcrypt-2.6.1-13.amzn2.0.3.x86_64 3/9
Verifying : python-keyczar-0.7lc-2.amzn2.noarch 4/9
Verifying : ansible-2.9.23-1.amzn2.noarch 5/9
Verifying : python2-paramiko-1.16.1-3.amzn2.0.3.noarch 6/9
Verifying : libtomcrypt-1.18.2-1.amzn2.0.1.x86_64 7/9
Verifying : python2-httplib2-0.18.1-3.amzn2.noarch 8/9
Verifying : sshpass-1.06-1.amzn2.0.1.x86_64 9/9

Installed:
ansible.noarch 0:2.9.23-1.amzn2

Dependency Installed:
libtomcrypt.x86_64 0:1.18.2-1.amzn2.0.1
python-keyczar.noarch 0:0.7lc-2.amzn2
python2-ecdsa.noarch 0:0.13.3-1.amzn2.0.1
python2-paramiko.noarch 0:1.16.1-3.amzn2.0.3

Complete!
0 ansible2-latest enabled \
[ =2.4.2 =2.4.6 =2.8 =stable ] \
2 httpd_modules available [ =1.0 =stable ] \
3 memcached1.5 available \
[ =1.5.1 =1.5.16 =1.5.17 ] \
6 postgresql10 available [ =10 =stable ] \
9 R3.4 available [ =3.4.3 =stable ] \
10 rust1 available \
[ =1.22.1 =1.26.0 =1.26.1 =1.27.2 =1.31.0 =1.38.0 ] \

```

```
C:\WINDOWS\system32\cmd. x + v
10 rust1           available \
   [ =1.22.1 =1.26.0 =1.26.1 =1.27.2 =1.31.0 =1.38.0
    =stable ]
18 libreoffice     available \
   [ =5.0.6.2_15 =5.3.6.1 =stable ]
19 gimp            available [ =2.8.22 ]
20 docker-latest   enabled \
   [ =17.12.1 =18.03.1 =18.06.1 =18.09.9 =stable ]
21 mate-desktop1.x available \
   [ =1.19.0 =1.20.0 =stable ]
22 GraphicsMagick1.3 available \
   [ =1.3.29 =1.3.32 =1.3.34 =stable ]
23 tomcat8.5       available \
   [ =8.5.31 =8.5.32 =8.5.38 =8.5.40 =8.5.42 =8.5.50
    =stable ]
24 epel             available [ =7.11 =stable ]
25 testing          available [ =1.0 =stable ]
26 ecs              available [ =stable ]
27 corretto8        available \
   [ =1.8.0.192 =1.8.0.202 =1.8.0.212 =1.8.0.222 =1.8.0.232
    =1.8.0.242 =stable ]
29 golang1.11      available \
   [ =1.11.3 =1.11.11 =1.11.13 =stable ]
30 squid4          available [ =4 =stable ]
32 lustre2.10      available \
   [ =2.10.5 =2.10.8 =stable ]
33 java-openjdk11   available [ =11 =stable ]
34 lyncs            available [ =stable ]
36 BCC              available [ =0.x =stable ]
37 mono             available [ =5.x =stable ]
38 nginx1          available [ =stable ]
40 mock              available [ =stable ]
41 postgresql11    available [ =11 =stable ]
43 livepatch        available [ =stable ]
44 python3.8        available [ =stable ]
45 haproxy2         available [ =stable ]
46 collected        available [ =stable ]
47 aws-nitro-enclaves-cli available [ =stable ]
48 R4               available [ =stable ]
_ kernel-5.4       available [ =stable ]
```

Now, check whether the ansible is installed or not by the command: "ansible --version"

```
C:\WINDOWS\system32\cmd. x + v
66 php8.1          available [ =stable ]
67 awscli           available [ =stable ]
68 php8.2          available [ =stable ]
69 dnsmasq          available [ =stable ]
[ec2-user@ip-172-31-26-141 ~]$ ansible --version
ansible 2.9.23
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['~/home/ec2-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.18 (default, Feb 28 2023, 02:51:06) [GCC 7.3.1 20180712 (Red Hat 7.3.1-15)]
[ec2-user@ip-172-31-26-141 ~]$ ansible localhost -m ping
usage: ansible [-h] [--version] [-v] [-b] [--become-method BECOME_METHOD]
                [--become-user BECOME_USER] [-K] [-i INVENTORY] [--list-hosts]
                [-l SUBSET] [-P POLL_INTERVAL] [-B SECONDS] [-o] [-t TREE] [-k]
                [-private-key PRIVATE_KEY_FILE] [-u REMOTE_USER]
                [-C CONNECTION] [-T TIMEOUT]
                [-ssh-common-args SSH_COMMON_ARGS]
                [-sftp-extra-args SFTP_EXTRA_ARGS]
                [-scp-extra-args SCP_EXTRA_ARGS]
                [-ssh-extra-args SSH_EXTRA_ARGS] [-C] [--syntax-check] [-D]
                [-e EXTRA_VARS] [--vault-id VAULT_IDS]
                [-ask-vault-pass | --vault-password-file VAULT_PASSWORD_FILES]
                [-f FORKS] [-M MODULE_PATH] [--playbook-dir BASEDIR]
                [-a MODULE_ARGS] [-m MODULE_NAME]
                pattern
ansible: error: unrecognized arguments: localhost
[ec2-user@ip-172-31-26-141 ~]$ ansible localhost -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[ec2-user@ip-172-31-26-141 ~]$ # rpm -qa | grep ansible
[ec2-user@ip-172-31-26-141 ~]$ su
Password:
su: Authentication failure
[ec2-user@ip-172-31-26-141 ~]$ su
Password:
su: Authentication failure
[ec2-user@ip-172-31-26-141 ~]$ su
```

Step 4: Now change the path to the root using the command "sudo su". Now before installing the ansible, we need to epel file cause ansible was not provided in the yum library. Install the epel using the command "sudo amazon-linux-extras install epel -y"

```

C:\WINDOWS\system32\cmd.x + 
Password:
su: Authentication failure
[ec2-user@ip-172-31-26-141 ~]$ sudo su
[root@ip-172-31-26-141 ec2-user]# rpm -qa | grep ansible
ansible-2.9.23-1.amzn2.noarch
[root@ip-172-31-26-141 ec2-user]# sudo amazon-linux-extras install epel -y
Installing epel-release
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-ansible2 amzn2extra-docker amzn2extra-epel amzn2extra-kernel-5.10
22 metadata files removed
8 square files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-ansible2
amzn2extra-docker
amzn2extra-epel
amzn2extra-kernel-5.10
(1/11): amzn2-core/2/x86_64/group.gz
(2/11): amzn2-core/2/x86_64/updateinfo
(3/11): amzn2extra-docker/2/x86_64/primary_db
(4/11): amzn2extra-epel/2/x86_64/updateinfo
(5/11): amzn2extra-epel/2/x86_64/primary_db
(6/11): amzn2extra-kernel-5.10/2/x86_64/updateinfo
(7/11): amzn2extra-ansible2/2/x86_64/updateinfo
(8/11): amzn2extra-ansible2/2/x86_64/primary_db
(9/11): amzn2extra-docker/2/x86_64/updateinfo
(10/11): amzn2extra-kernel-5.10/2/x86_64/primary_db
(11/11): amzn2-core/2/x86_64/primary_db
| 3.7 kB 00:00:00
| 3.0 kB 00:00:00
| 3.0 kB 00:00:00
| 3.0 kB 00:00:00
| 3.0 kB 00:00:00
| 2.5 kB 00:00:00
| 583 kB 00:00:00
| 103 kB 00:00:00
| 76 B 00:00:00
| 1.8 kB 00:00:00
| 26 kB 00:00:00
| 76 B 00:00:00
| 48 kB 00:00:00
| 8.0 kB 00:00:00
| 16 MB 00:00:01
| 70 MB 00:00:02
Resolving Dependencies
--> Running transaction check
--> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Version          Repository      Size
=====           ======          =====
epel-release     noarch        7-11
                                         amzn2extra-epel      15 k
=====
                                                ======Installing:
=====
                                                Arch
=====
Partly sunny. 15:32 04-04-2023

```

```

C:\WINDOWS\system32\cmd.x + 
53 unbound1.13      available [ =stable ]
54 mariadb10.5      available [ =stable ]
55 kernel-5.10=latest enabled [ =stable ]
56 redis6            available [ =stable ]
57 ruby3.0           available [ =stable ]
58 postgresql12      available [ =stable ]
59 postgresql13      available [ =stable ]
60 mock2             available [ =stable ]
61 dnsmasq2.85       available [ =stable ]
62 kernel-5.15       available [ =stable ]
63 postgresql14      available [ =stable ]
64 firefox            available [ =stable ]
65 lustre             available [ =stable ]
66 php8.1             available [ =stable ]
67 awscli1            available [ =stable ]
68 php8.2             available [ =stable ]
69 dnsmasq            available [ =stable ]
[root@ip-172-31-26-141 ec2-user]# sudo yum repolist
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
240 packages excluded due to repository priority protections
repo id          repo name          status
amzn2-core/2/x86_64    Amazon Linux 2 core repository   30,342amzn2extra-ansible2/2/x86_64    Amazon Extras repo for ansible2      65amzn2ext
ra-docker/2/x86_64    Amazon Extras repo for docker   81amzn2extra-epel/2/x86_64    Amazon Extras repo for epel      1amzn2extra-kerne
l-5.10/2/x86_64    Amazon Extras repo for kernel-5   272epel/x86_64    Extra Packages for Enterprise L 13,530+240repolist: 44,291
[root@ip-172-31-26-141 ec2-user]# sudo yum-config-manager --enable epel
=====
===== repo: epel ======[epel]
async = True
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/2
baseurl =
cache = 0
cachedir = /var/cache/yum/x86_64/2/epel
check_config_file_age = True
compare_providers_priority = 80
cost = 1000
deltarpm_metadata_percentage = 100
deltarpm_percentage =
enabled = True
enablegroups = True
15:32 04-04-2023

```

Now using the command “sudo yum repolist ”.It will list all installed repositories and this are disabled and enabled repositories that are currently present.

```

[root@ip-172-31-26-141 ec2-user]# sudo yum repolist
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
240 packages excluded due to repository priority protections
repo id          repo name          status
amzn2-core/2/x86_64    Amazon Linux 2 core repository   30,342amzn2extra-ansible2/2/x86_64    Amazon Extras repo for ansible2      65amzn2ext
ra-docker/2/x86_64    Amazon Extras repo for docker   81amzn2extra-epel/2/x86_64    Amazon Extras repo for epel      1amzn2extra-kerne
l-5.10/2/x86_64    Amazon Extras repo for kernel-5   272epel/x86_64    Extra Packages for Enterprise L 13,530+240repolist: 44,291

```

Now,enable the repolist

Sudo yum-config-managet –enable epel

```
repo id          repo name           status
amzn2-core/x86_64      Amazon Linux 2 core repository   30,342amzn2extra-ansible2/x86_64    Amazon Extras repo for ansible2      65amzn2ext
ra-docker/x86_64      Amazon Extras repo for docker     81amzn2extra-epel/x86_64        Amazon Extras repo for epel          lamzn2extra-kerne
l-5.10/x86_64        Amazon Extras repo for kernel-5   272epel/x86_64          Extra Packages for Enterprise L 13,530+240repolist: 44,291
[root@ip-172-31-26-141 ec2-user]# sudo yum-config-manager --enable epel
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
=====
repo: epel =====[epel]
async = True
bandwidth = 0
basepersistdir = /var/lib/yum/repos/x86_64/
baseurl =
cache = 0
cachedir = /var/cache/yum/x86_64/2/epel
check_config_file_age = True
compare_providers_priority = 80
cost = 1000
deltarpm_metadata_percentage = 100
deltarpm_percentage =
enabled = True
enablegroups = True
34°C
Partly sunny
ENG IN 15:32 04-04-2023
```

Now, we have to disable the ansible2

Sudo amazon-linux-extras disable ansible2

```
C:\WINDOWS\system32\cmd. + 
mirrorlist_expire = 86400
name = Extra Packages for Enterprise Linux 7 - x86_64
old_base_cache_dir =
password =
persistdir = /var/lib/yum/repos/x86_64/2/epel
pkmdir = /var/cache/yum/x86_64/2/epel/packages
priority = 99
proxy = False
proxy_dict =
proxy_password =
proxy_username =
repo_gpgcheck = False
report_instanceid = False
retries = 7
skip_if_unavailable = False
ssl_check_cert_permissions = True
sslcacert =
sslcclientcert =
sslcclientkey =
sslverify = True
throttle = 0
timeout = 5.0
ui_id = epe1/x86_64
ui_repo_id_vars = releasever,
basearch
username =
[root@ip-172-31-26-141 ec2-user]# sudo amazon-linux-extras disable ansible2
Beware that disabling topics is not supported after they are installed.
 0 ansible2
           available \
  [ =2.4.2  =2.4.6  =2.8  =stable ]
 2 httpd_modules
           available   [ =1.0  =stable ]
 3 memcached1.5
           available \
  [ =1.5.1  =1.5.16  =1.5.17 ]
 6 postgresql10
           available   [ =10  =stable ]
 9 R3.4
           available   [ =3.4.3  =stable ]
10 rust1
           available \
  [ =1.22.1  =1.26.0  =1.26.1  =1.27.2  =1.31.0  =1.38.0
  =stable ]
18 libreoffice
           available \
15:33
04-04-2023
```

Now install ansible

sudo yum –enable repo epel install ansible

```

61 dnsMasq-2.85      available  [ =stable ]
62 kernel-5.15        available  [ =stable ]
63 postgresql14        available  [ =stable ]
64 firefox             available  [ =stable ]
65 lustre              available  [ =stable ]
66 php8.1              available  [ =stable ]
67 awscli              available  [ =stable ]
68 php8.2              available  [ =stable ]
69 dnsmasq             available  [ =stable ]
[root@ip-172-31-26-141 ec2-user]# sudo yum --enablerepo epel install ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core           | 3.7 KB  00:00
amzn2extra-docker   | 3.0 KB  00:00
amzn2extra-epel     | 3.0 KB  00:00
amzn2extra-kernel-5.10 | 3.0 KB  00:00
epel/x86_64/metalink | 6.7 kB  00:00
224 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
--> Package ansible.noarch 0:2.9.23-1.amzn2 will be updated
--> Package ansible.noarch 0:2.9.27-1.el7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package          Arch    Version       Repository   Size
=====
ansible         noarch  2.9.27-1.el7   epel        17 M
Upgrading:
=====
Transaction Summary
=====
Upgrades       1 Package

Total download size: 17 M
Partly sunny  15:33
[4:40]

```

```

Package : epel-release-7-11.noarch (@amzn2extra-epel)
From   : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7
Is this ok [y/N]: Y
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating : ansible-2.9.27-1.el7.noarch                1/2
  Cleanup  : ansible-2.9.23-1.amzn2.noarch            2/2
  Verifying : ansible-2.9.27-1.el7.noarch                1/2
  Verifying : ansible-2.9.23-1.amzn2.noarch            2/2

Updated:
  ansible.noarch 0:2.9.27-1.el7

Complete!
[root@ip-172-31-26-141 ec2-user]# ansible --version
ansible 2.9.27
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /bin/ansible
  python version = 2.7.18 (default, Feb 28 2023, 02:51:06) [GCC 7.3.1 20180712 (Red Hat 7.3.1-15)]
[root@ip-172-31-26-141 ec2-user]# client_loop: send disconnect: Connection reset
C:\Users\Sahsra\Downloads>ansible --version
Partly sunny  15:34
[4:40]

```

Now check the ansible version “ansible –version” and to deploy any application we required ansible and python.so install python and check the version of python “python --version”

```

C:\WINDOWS\system32\cmd. + -
2 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-141 ~]$ ansible --version
ansible 2.9.27
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ec2-user/.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 = 2.7.18 (default, Feb 28 2023, 02:51:06) [GCC 7.3.1 20180712 (Red Hat 7.3.1-15)]
[ec2-user@ip-172-31-26-141 ~]$ python --version
Python 2.7.18

```

Step 5: Now create and add a user using the commands:

```

sudo -i
useradd -m -s /bin/bash/user1
passwd user1

```

```
[ec2-user@ip-172-31-26-141 ~]$ sudo -i
[root@ip-172-31-26-141 ~]# useradd -m -s /bin/bash user1
[root@ip-172-31-26-141 ~]# passwd user1
Changing password for user user1.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-26-141 ~]# echo -e 'user1:tALL=(ALL):tNOPASSWD:tALL' > /etc/sudoers.d/user1
[root@ip-172-31-26-141 ~]# cat /etc/sudoers.d/user1
user1:tALL=(ALL):tNOPASSWD:tALL
[root@ip-172-31-26-141 ~]# cat /etc/sudoers.d/user1
user1  ALL=(ALL) NOPASSWD:    ALL
[root@ip-172-31-26-141 ~]# sudo yum whatprovides */mkpasswd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
224 packages excluded due to repository priority protections
amzn2-core/2/x86_64/filelists_db | 54 MB  00:00
amzn2extra-docker/2/x86_64/filelists_db | 28 kB  00:00
amzn2extra-epel/2/x86_64/filelists_db | 882 B   00:00
amzn2extra-kernel-5.10/2/x86_64/filelists_db | 7.1 MB  00:00
epel/x86_64/filelists_db | 12 MB  00:00
cjdns-21.1-3.el7.x86_64 : The privacy-friendly network without borders
Repo
Matched from:
Filename   : /usr/libexec/cjdns/mkpasswd

31°C Partly sunny  Search  Apps  File  Home  Mail  Photos  Videos  Chrome  File Explorer  Task View  ENG IN  WiFi  17:15  04-04-2023
```

C:\WINDOWS\system32\cmd. x + v

```
expect-5.45-14.amzn2.x86_64 : A program-script interaction and testing
                               : utility
Repo      : amzn2-core
Matched from:
Filename  : /usr/bin/mkpasswd

expect-5.45-14.amzn2.0.2.x86_64 : A program-script interaction and testing
                               : utility
Repo      : amzn2-core
Matched from:
Filename  : /usr/bin/mkpasswd

[root@ip-172-31-26-141 ~]# sudo yum install expect
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core           | 3.7 kB   00:00
224 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
--> Package expect.x86_64 0:5.45-14.amzn2.0.2 will be installed
--> Processing Dependency: libtcl8.5.so()(64bit) for package: expect-5.45-14.amzn2.0.2.x86_64
--> Running transaction check
--> Package tcl.x86_64 1:8.5.13-8.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
====Package          Arch      Version       Repository      Size
====Installing:
expect     x86_64    5.45-14.amzn2.0.2      amzn2-core    263 k
Installing for dependencies:
tcl       x86_64    1:8.5.13-8.amzn2.0.2      amzn2-core    1.9 M

Transaction Summary
=====
====Install 1 Package (+1 Dependent package)

Total download size: 2.2 M
31°C Partly sunny  Search  Apps  File  Home  Mail  Photos  Videos  Chrome  File Explorer  Task View  ENG IN  WiFi  17:15  04-04-2023
```

```
Total download size: 2.2 M
Installed size: 4.9 M
Is this ok [y/d/N]: y
Downloading packages:
(1/2): expect-5.45-14.amzn2.0.2.x86_64.rpm | 263 kB  00:00
(2/2): tcl-8.5.13-8.amzn2.0.2.x86_64.rpm | 1.9 MB  00:00
                                                Total
                                                13 MB/s | 2.2 MB  00:00

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:tcl-8.5.13-8.amzn2.0.2.x86_64        1/2
  Installing : expect-5.45-14.amzn2.0.2.x86_64        2/2
  Verifying  : 1:tcl-8.5.13-8.amzn2.0.2.x86_64        1/2
  Verifying  : expect-5.45-14.amzn2.0.2.x86_64        2/2

Installed:
  expect.x86_64 0:5.45-14.amzn2.0.2

Dependency Installed:
  tcl.x86_64 1:8.5.13-8.amzn2.0.2

Complete!
[root@ip-172-31-26-141 ~]# mkpasswd user1
-V6rp3tuJ
```

Step 6: Now generate the ssh key using command: “ssh-keygen -t rsa”

```
[root@ip-172-31-26-141 ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:+KoqyXhb4zJhtji4pBNFVX5tTLLKN7Gib9mXITqEw root@ip-172-31-26-141.ec2.internal
The key's randomart image is:
+---[RSA 2048]---+
|   . . + . . |
| . . + . o |
| o . . o o |
+---[SHA256]---+
31C Partly sunny 17:22 04-04-2023
```

```
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:+KoqyXhb4zJhtji4pBNFVX5tTLLKN7Gib9mXITqEw root@ip-172-31-26-141.ec2.internal
The key's randomart image is:
+---[RSA 2048]---+
|   . . + . . |
| . . + . o |
| o . . o o |
+---[SHA256]---+
[root@ip-172-31-26-141 ~]#
[root@ip-172-31-26-141 ~]# sudo -su user1
[user1@ip-172-31-26-141 root]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user1/.ssh/id_rsa):
Created directory '/home/user1/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user1/.ssh/id_rsa.
Your public key has been saved in /home/user1/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:kB6kmyqsYRwpaDn5NorvLzg1Sp2ZlEy/iuaYf3Dm2/k user1@ip-172-31-26-141.ec2.internal
The key's randomart image is:
+---[RSA 2048]---+
|   . . |
|   . o |
|   o.o |
| . + o.. |
|oR oot .S |
|= * .X . |
|o= & + |
|+=O B . |
|=B+=.o.E |
+---[SHA256]---+
[user1@ip-172-31-26-141 root]$ cd ..
31C Partly sunny 17:22 04-04-2023
```

```
= * .X .
|o= & +
|+=O B .
|=B+=.o.E
+---[SHA256]---+
[user1@ip-172-31-26-141 root]$ cd ..
[user1@ip-172-31-26-141 ~]$ cd .ssh
bash: cd: No such file or directory
[user1@ip-172-31-26-141 ~]$ ls
bin dev home lib64 media opt root sbin sys usr
boot etc lib local nt proc run srv tmp var
[user1@ip-172-31-26-141 ~]$ sudo -su user1
[user1@ip-172-31-26-141 ~]$ cd ~
[user1@ip-172-31-26-141 ~]$ ls
[user1@ip-172-31-26-141 ~]$ cd root
bash: cd: root: No such file or directory
[user1@ip-172-31-26-141 ~]$ cd ..
[user1@ip-172-31-26-141 home]$ cd ..
[user1@ip-172-31-26-141 ~]$ cd ..
[user1@ip-172-31-26-141 ~]$ cd ~/.ssh
[user1@ip-172-31-26-141 .ssh]$ ls
id_rsa id_rsa.pub
[user1@ip-172-31-26-141 .ssh]$ ll
total 8
-rw----- 1 user1 user1 1675 Apr  4 09:34 id_rsa
-rw-r--r-- 1 user1 user1 417 Apr  4 09:34 id_rsa.pub
[user1@ip-172-31-26-141 .ssh]$ ssh -i ~/.ssh/id_rsa 54.158.3.178
The authenticity of host '54.158.3.178' ('54.158.3.178') can't be established.
ECDSA key fingerprint is SHA256:zuPiZP509ERRNKviuzk8p4pU2T+MSLSzIxtcromT0g.
ECDSA key fingerprint is MD5:e4:ac:a6:be:e1:b6:02:a1:79:e0:55:ed:82:b0:69:68.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.158.3.178' (ECDSA) to the list of known hosts.
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[user1@ip-172-31-26-141 .ssh]$ sudo yum install git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                         | 3.7 kB  00:00:00
224 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
31C Partly sunny 17:23 04-04-2023
```

Now, install the git using command “sudo yum install git”

```
[user1@ip-172-31-26-141 .ssh]$ sudo yum install git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
224 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
31°C Partly sunny 17:23 ENG IN 04-04-2023
```

C:\WINDOWS\system32\cmd

```
224 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.39.2-1.amzn2.0.1 will be installed
--> Processing Dependency: perl-Git = 2.39.2-1.amzn2.0.1 for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.39.2-1.amzn2.0.1 for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core = 2.39.2-1.amzn2.0.1 for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git::I18N) for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git) for package: git-2.39.2-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.39.2-1.amzn2.0.1 will be installed
--> Package git-core-doc.noarch 0:2.39.2-1.amzn2.0.1 will be installed
--> Package perl-Git.noarch 0:2.39.2-1.amzn2.0.1 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.39.2-1.amzn2.0.1.noarch
--> Package perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Size                               Package          Arch      Version           Repository
=====                         ======          ======     ======             =====
git            x86_64    2.39.2-1.amzn2.0.1   amzn2-core    65 k
Installing for dependencies:
git-core        x86_64    2.39.2-1.amzn2.0.1   amzn2-core    8.8 M
git-core-doc    noarch    2.39.2-1.amzn2.0.1   amzn2-core    3.0 M
perl-Error      noarch    1:0.17020-2.amzn2   amzn2-core    32 k
perl-Git        noarch    2.39.2-1.amzn2.0.1   amzn2-core    41 k
perl-TermReadKey x86_64    2.30-20.amzn2.0.2   amzn2-core    31 k

Transaction Summary
=====
Install 1 Package (+5 Dependent packages)

Total download size: 12 M
Installed size: 43 M
Is this ok [y/d/N]: y
Downloading packages:
31°C Partly sunny 17:23 ENG IN 04-04-2023
```

C:\WINDOWS\system32\cmd

```
Total download size: 12 M
Installed size: 43 M
Is this ok [y/d/N]: y
Downloading packages:
(1/6): git-2.39.2-1.amzn2.0.1.x86_64.rpm | 65 kB 00:00:00
(2/6): git-core-doc-2.39.2-1.amzn2.0.1.noarch.rpm | 3.0 MB 00:00:00
(3/6): perl-Error-0.17020-2.amzn2.noarch.rpm | 32 kB 00:00:00
(4/6): perl-Git-2.39.2-1.amzn2.0.1.noarch.rpm | 41 kB 00:00:00
(5/6): perl-TermReadKey-2.30-20.amzn2.0.2.x86_64.rpm | 31 kB 00:00:00
(6/6): git-core-2.39.2-1.amzn2.0.1.x86_64.rpm | 8.8 MB 00:00:00
Total                                         41 MB/s | 12 MB 00
:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : git-core-2.39.2-1.amzn2.0.1.x86_64          1/6
  Installing : git-core-doc-2.39.2-1.amzn2.0.1.noarch       2/6
  Installing : 1:perl-Error-0.17020-2.amzn2.noarch         3/6
  Installing : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64  4/6
  Installing : perl-Git-2.39.2-1.amzn2.0.1.noarch         5/6
  Installing : git-2.39.2-1.amzn2.0.1.x86_64             6/6
  Verifying  : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64  1/6
  Verifying  : perl-Git-2.39.2-1.amzn2.0.1.noarch         2/6
  Verifying  : git-core-2.39.2-1.amzn2.0.1.x86_64          3/6
  Verifying  : git-core-doc-2.39.2-1.amzn2.0.1.noarch       4/6
  Verifying  : 1:perl-Error-0.17020-2.amzn2.noarch         5/6
  Verifying  : git-2.39.2-1.amzn2.0.1.x86_64             6/6

Installed:
  git.x86_64 0:2.39.2-1.amzn2.0.1

Dependency Installed:
  git-core.x86_64 0:2.39.2-1.amzn2.0.1
  git-core-doc.noarch 0:2.39.2-1.amzn2.0.1
  perl-Error.noarch 1:0.17020-2.amzn2
  perl-Git.noarch 0:2.39.2-1.amzn2.0.1
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
31°C Partly sunny 17:24 ENG IN 04-04-2023
```

Step 7: Now clone the repository using the command “git clone repository_name”

```
Complete!
[user1@ip-172-31-26-141 .ssh]$ git clone https://github.com/devops4solutions/Ansible-Sample-Application-Deployment.git
Cloning into 'Ansible-Sample-Application-Deployment'...
remote: Enumerating objects: 296, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 296 (delta 26), reused 21 (delta 21), pack-reused 267
Receiving objects: 100% (296/296), 49.85 KiB | 6.23 MiB/s, done.
Resolving deltas: 100% (94/94), done.
[user1@ip-172-31-26-141 .ssh]$ ls
Ansible-Sample-Application-Deployment  id_rsa  id_rsa.pub  known_hosts
```

```
[user1@ip-172-31-26-141 .ssh]$ ls
Ansible-Sample-Application-Deployment id_rsa id_rsa.pub known_hosts
[user1@ip-172-31-26-141 .ssh]$ cd Ansible-Sample-Application-Deployment
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible.cfg group_vars Jenkinsfile README.md test
configs inventories main.yml roles
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ vi ansible_aut.pem [user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem configs inventories main.yml roles
ansible.cfg group_vars Jenkinsfile README.md test
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd inventories
[user1@ip-172-31-26-141 inventories]$ ls
dev prod
[user1@ip-172-31-26-141 inventories]$ cd dev
[user1@ip-172-31-26-141 dev]$ ls
hosts
[user1@ip-172-31-26-141 dev]$ vi hosts
[user1@ip-172-31-26-141 dev]$ cd ..
[user1@ip-172-31-26-141 inventories]$ cd ..
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd roles/
[user1@ip-172-31-26-141 roles]$ ls
add_devops_user tomcat
[user1@ip-172-31-26-141 roles]$ cd add_devops_user
[user1@ip-172-31-26-141 add_devops_user]$ ls
handlers tasks
[user1@ip-172-31-26-141 add_devops_user]$ cd tasks/
[user1@ip-172-31-26-141 tasks]$ ls
add_user.yml
[user1@ip-172-31-26-141 tasks]$ vi add_user.yml
```



31°C Partly sunny 1725 ENG IN 04-04-2023

Add_user.yml:

- name: Add a new user named devops


```
user:
  name={{username}}
  password= {{devops_password}}
```
- name: Add devops user to the sudoers


```
copy:
  dest: "/etc/sudoers.d/{{username}}"
  content: "{{username}} ALL=(ALL) NOPASSWD: ALL"
```
- name: Deploy SSH Key

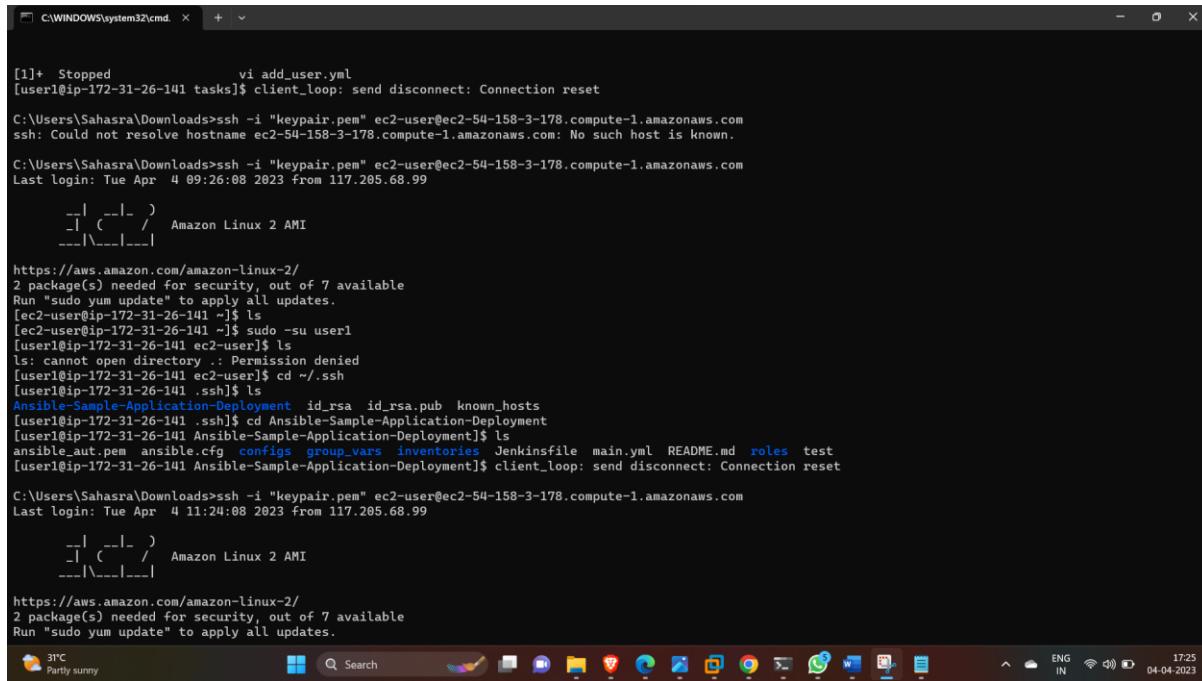

```
authorized_key: user={{username}}
  key="{{lookup ('file', '~/.ssh/id_rsa.pub') }}"
  state=present
```
- name: Disable Password Authentication


```
lineinfile:
  dest=/etc/ssh/sshd_config
  regexp='^PasswordAuthentication'
  line="PasswordAuthentication no"
  state=present
  backup=yes
```
- name: Disable Root Login


```
lineinfile:
  dest=/etc/ssh/sshd_config
  regexp='^PermitRootLogin'
  line="PermitRootLogin no"
  state=present
  backup=yes
```
- notify:
 - restart ssh

dev.yml:

Prod.yml:



```
[1]+ Stopped                  vi add_user.yml
[user1@ip-172-31-26-141 tasks]$ client_loop: send disconnect: Connection reset
C:\Users\Sahasra\Downloads>ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
ssh: Could not resolve hostname ec2-54-158-3-178.compute-1.amazonaws.com: No such host is known.

C:\Users\Sahasra\Downloads>ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
Last login: Tue Apr  4 09:26:08 2023 from 117.205.68.99
--| --|_
-| ( /  Amazon Linux 2 AMI
---\|---|---|_<

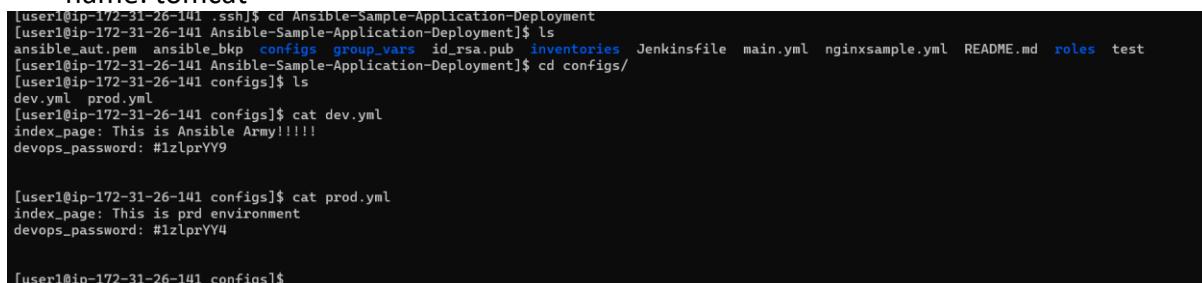
https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-141 ~]$ ls
[ec2-user@ip-172-31-26-141 ~]$ sudo -su user1
[user1@ip-172-31-26-141 ec2-user]$ ls
ls: cannot open directory .: Permission denied
[user1@ip-172-31-26-141 ec2-user]$ cd ~/.ssh
[user1@ip-172-31-26-141 .ssh]$ ls
Ansible-Sample-Application-Deployment  id_rsa  id_rsa.pub  known_hosts
[user1@ip-172-31-26-141 .ssh]$ cd Ansible-Sample-Application-Deployment
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem  ansible.cfg  configs  group_vars  inventories  Jenkinsfile  main.yml  README.md  roles  test
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ client_loop: send disconnect: Connection reset

C:\Users\Sahasra\Downloads>ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
Last login: Tue Apr  4 11:24:08 2023 from 117.205.68.99
--| --|_
-| ( /  Amazon Linux 2 AMI
---\|---|---|_<

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
```

main.yml:

```
---
- hosts: all
  become: true
  become_user: root
  gather_facts: false
  tasks:
    - include_role:
        name: add_devops_user
        tasks_from: add_user.yml
- hosts: all
  become: true
  become_user: root
  gather_facts: false
  tasks:
    - include_role:
        name: tomcat
        tasks_from: prod.yml
```



```
[user1@ip-172-31-26-141 .ssh]$ cd Ansible-Sample-Application-Deployment
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem  ansible_bkp  configs  group_vars  id_rsa.pub  inventories  Jenkinsfile  main.yml  nginxsample.yml  README.md  roles  test
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd configs/
[user1@ip-172-31-26-141 configs]$ ls
dev.yml  prod.yml
[user1@ip-172-31-26-141 configs]$ cat dev.yml
index_page: This is Ansible Army!!!!
devops_password: #1zlprYY9

[user1@ip-172-31-26-141 configs]$ cat prod.yml
index_page: This is prd environment
devops_password: #1zlprYY4

[user1@ip-172-31-26-141 configs]$
```

```
C:\WINDOWS\system32\cmd.x Command Prompt + 
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sahasra>cd downloads

C:\Users\Sahasra\Downloads>ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
ssh: Could not resolve hostname ec2-54-158-3-178.compute-1.amazonaws.com: No such host is known.

C:\Users\Sahasra\Downloads>ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
ssh: Could not resolve hostname ec2-54-158-3-178.compute-1.amazonaws.com: No such host is known.

C:\Users\Sahasra\Downloads>ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
Last login: Tue Apr  4 11:34:25 2023 from 157.48.195.203
   _|_ _|_
  _| (   /  Amazon Linux 2 AMI
 ---\_\_|\_--|_

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-141 ~]$ cd ~/.ssh
[ec2-user@ip-172-31-26-141 .ssh]$ ls
authorized_keys
[ec2-user@ip-172-31-26-141 .ssh]$ sudo su user1
[user1@ip-172-31-26-141 ~]$ cd ~/
[user1@ip-172-31-26-141 ~]$ ls
Ansible-Sample-Application-Deployment id_rsa id_rsa.pub known_hosts
[user1@ip-172-31-26-141 ~]$ cd Ansible-Sample-Application-Deployment
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem ansible.cfg configs group_vars inventories Jenkinsfile main.yml README.md roles test
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cat ansible.cfg
[defaults]
host_key_checking = False

[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ mv ansible.cfg ansible_bkp
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem ansible_bkp configs group_vars inventories Jenkinsfile main.yml README.md roles test
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/dev.yml'

25°C Humid  Search ENG IN 06:29 05-04-2023
```

```
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/dev.yml'

PLAY [all] *****
TASK [include_role : add_devops_user] *****
TASK [add_devops_user : Add a new user named devops] *****
[172.31.26.141]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: Warning: Permanently added '172.31.26.141' (ECDSA) to the list of known hosts.\r\nWarning: UNPROTECTED PRIVATE KEY FILE! @\r\n\\nPermissions 0664 for '/home/user1/.ssh/Ansible-Sample-Application-Deployment/ansible_aut.pem' are too open.\r\nIt is required that your private key files are NOT accessible by others.\r\nThis private key will be ignored.\r\nLoad key \"/home/user1/.ssh/Ansible-Sample-Application-Deployment/ansible_aut.pem\": bad permissions\r\nPermission denied (publickey,gssapi-keyex,gssapi-with-mic).", "unreachable": true}

PLAY RECAP *****172.31.26.141 : ok=0
changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0

[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ sudo chmod 600 ansible_aut.pem
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/dev.yml'

PLAY [all] *****
TASK [include_role : add_devops_user] *****
TASK [add_devops_user : Add a new user named devops] *****
[WARNING]: Platform linux on host 172.31.26.141 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
changed: [172.31.26.141]

TASK [add_devops_user : Add devops user to the sudoers] *****
changed: [172.31.26.141]

TASK [add_devops_user : Deploy SSH Key] *****
[WARNING]: Unable to find 'id_rsa.pub' in expected paths (use -vvvv to see paths)

25°C Humid  Search ENG IN 06:29 05-04-2023
```

```
C:\WINDOWS\system32\cmd. x Command Prompt x + - m -e '@configs/dev.yml'

PLAY [all] *****
TASK [include_role : add_devops_user] *****
TASK [add_devops_user : Add a new user named devops] *****
[WARNING]: Platform linux on host 172.31.26.141 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
changed: [172.31.26.141]

TASK [add_devops_user : Add devops user to the sudoers] *****
changed: [172.31.26.141]

TASK [add_devops_user : Deploy SSH Key] *****
[WARNING]: Unable to find 'id_rsa.pub' in expected paths (use -vvvvv to see paths)
fatal: [172.31.26.141]: FAILED! => {"msg": "An unhandled exception occurred while running the lookup plugin 'file'. Error was a <class 'ansible.errors.AnsibleError>; original message: could not locate file in lookup: id_rsa.pub"}

PLAY RECAP *****
    changed=2    unreachable=0      failed=1    skipped=0    rescued=0    ignored=0

[User@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_authorized_keys -e '@configs/dev.yml'

PLAY [all] *****
TASK [include_role : add_devops_user] *****
TASK [add_devops_user : Add a new user named devops] *****
[WARNING]: Platform linux on host 172.31.26.141 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.26.141]

TASK [add_devops_user : Add devops user to the sudoers] *****
ok: [172.31.26.141]

TASK [add_devops_user : Deploy SSH Key] *****
[WARNING]: Unable to find 'id_rsa.pub' in expected paths (use -vvvvv to see paths)
fatal: [172.31.26.141]: FAILED! => {"msg": "An unhandled exception occurred while running the lookup plugin 'file'. Error was a <class 'ansible.errors.AnsibleError>; original message: could not locate file in lookup: id_rsa.pub"}

PLAY RECAP *****
    changed=0    unreachable=0      failed=1    skipped=0    rescued=0    ignored=0
```

```
ansibile_all_pem ansible_bkp configs group_vars id_rsa.pub inventories Jenkinsfile main.yml README.md roles test
[user@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ vi id_rsa.pub
[user@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_all.pem -e '@configs/dev.yml'

PLAY [all] ****
TASK [include_role : add_devops_user] ****

TASK [add_devops_user : Add a new user named devops] ****
[WARNING]: Platform linux on host 54.158.3.178 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [54.158.3.178]

TASK [add_devops_user : Add devops user to the sudoers] ****
ok: [54.158.3.178]

TASK [add_devops_user : Deploy SSH Key] ****
changed: [54.158.3.178]

TASK [add_devops_user : Disable Password Authentication] ****
ok: [54.158.3.178]

TASK [add_devops_user : Disable Root Login] ****
```

```
C:\WINDOWS\system32\cmd. x Command Prompt x + -  
changed: [54.158.3.178]  
  
PLAY [all] ****  
  
TASK [include_role : tomcat] ****  
  
TASK [tomcat : Install Java 1.7] ****  
changed: [54.158.3.178]  
  
TASK [add group "tomcat"] ****  
changed: [54.158.3.178]  
  
TASK [add user "tomcat"] ****  
changed: [54.158.3.178]  
  
TASK [tomcat : Download Tomcat] ****  
changed: [54.158.3.178]  
  
TASK [tomcat : Extract archive] ****  
[WARNING]: Consider using the unarchive module rather than running 'tar'. If you need to use command because unarchive is insufficient you can add 'warn: false' to this command task or set 'command_warnings=False' in ansible.cfg to get rid of this message.  
changed: [54.158.3.178]  
  
TASK [tomcat : Symlink install directory] ****  
changed: [54.158.3.178]  
  
TASK [tomcat : Change ownership of Tomcat installation] ****  
changed: [54.158.3.178]  
  
TASK [tomcat : Configure Tomcat server] ****  
changed: [54.158.3.178]  
  
TASK [tomcat : Configure Tomcat users] ****  
changed: [54.158.3.178]  
  
TASK [tomcat : Create sample directory] ****  
changed: [54.158.3.178]  
  
TASK [tomcat : copy sample index.html file] ****  
changed: [54.158.3.178]
```

```

C:\WINDOWS\system32\cmd x Command Prompt x + 
changed: [54.158.3.178]

PLAY [all] ****
TASK [include_role : tomcat] ****
TASK [tomcat : Install Java 1.7] ****
changed: [54.158.3.178]

TASK [add group "tomcat"] ****
changed: [54.158.3.178]

TASK [add user "tomcat"] ****
changed: [54.158.3.178]

TASK [tomcat : Download Tomcat] ****
changed: [54.158.3.178]

TASK [tomcat : Extract archive] ****
[WARNING]: Consider using the unarchive module rather than running 'tar'. If you need to use command because unarchive is insufficient you can add 'warn: false' to this command task or set 'command_warnings=False' in ansible.cfg to get rid of this message.
changed: [54.158.3.178]

TASK [tomcat : Symlink install directory] ****
changed: [54.158.3.178]

```

Now add all the private IP address of remote servers to the hosts file.

```

C:\WINDOWS\system32\cmd x Command Prompt x + 
[user1@ip-172-31-26-141 ec2-user]$ cd ~/.ssh
[user1@ip-172-31-26-141 .ssh]$ ls
Ansible-Sample-Application-Deployment id_rsa id_rsa.pub known_hosts
[user1@ip-172-31-26-141 .ssh]$ cd Ansible-Sample-Application-Deployment
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem configs id_rsa.pub Jenkinsfile README.md test
ansible_bkp group_vars inventories main.yml roles
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd inventories
[user1@ip-172-31-26-141 inventories]$ ls
dev prod
[user1@ip-172-31-26-141 inventories]$ cd dev
[user1@ip-172-31-26-141 dev]$ ls
hosts
[user1@ip-172-31-26-141 dev]$ vi hosts
[user1@ip-172-31-26-141 dev]$ cat hosts
[all]
172.31.25.148

```

```

[user1@ip-172-31-26-141 dev]$ cat hosts
[all]
172.31.25.148

[user1@ip-172-31-26-141 dev]$ cd ..
[user1@ip-172-31-26-141 inventories]$ cd ..
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd roles/
[user1@ip-172-31-26-141 roles]$ ls
add_devops_user tomcat
[user1@ip-172-31-26-141 roles]$ cd add_devops_user/
[user1@ip-172-31-26-141 add_devops_user]$ ls
handlers tasks
[user1@ip-172-31-26-141 add_devops_user]$ cd tasks
[user1@ip-172-31-26-141 tasks]$ ls
add_user.yml
[user1@ip-172-31-26-141 tasks]$ vi add_user.yml
[user1@ip-172-31-26-141 tasks]$ cd ..
[user1@ip-172-31-26-141 add_devops_user]$ cd ..
[user1@ip-172-31-26-141 roles]$ cd ..
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cat ansible.cfg
cat: ansible.cfg: No such file or directory
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem configs id_rsa.pub Jenkinsfile README.md test
ansible_bkp group_vars inventories main.yml roles
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cat ansible_bkp
[defaults]

```

```

C:\WINDOWS\system32\cmd x Command Prompt x + 
ED25519 key fingerprint is SHA256:c/FGzERihMaXjdrXYBHNZtzBunn3s9Ldrt00kkgA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-158-3-178.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Tue Apr  4 15:43:05 2023 from 106.217.217.84
  _|_ _|_
  _|(_ /  Amazon Linux 2 AMI
  ___\___|___|_

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-141 ~]$ sudo su user1
[user1@ip-172-31-26-141 ec2-user]$ cd ~/.ssh
[user1@ip-172-31-26-141 .ssh]$ ping client_loop: send disconnect: Connection reset

C:\Users\Sahasra\Downloads>ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
Last login: Tue Apr  4 17:38:19 2023 from 106.76.212.27
  _|_ _|_
  _|(_ /  Amazon Linux 2 AMI
  ___\___|___|_

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-141 ~]$ sudo su user1
[user1@ip-172-31-26-141 ec2-user]$ cd ~/.ssh
[user1@ip-172-31-26-141 .ssh]$ ls
Ansible-Sample-Application-Deployment id_rsa id_rsa.pub known_hosts
[user1@ip-172-31-26-141 .ssh]$ cd Ansible-Sample-Application-Deployment

```

Now ping the work nodes to the master node using command: "ping private_IP_address"

```
ansible-Sample-Application-Deployment id_rsa id_rsa.pub known_hosts
[user1@ip-172-31-26-141 .ssh]$ cd Ansible-Sample-Application-Deployment
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ping 172.31.25.148
PING 172.31.25.148 (172.31.25.148) 56(84) bytes of data.
64 bytes from 172.31.25.148: icmp_seq=1 ttl=255 time=0.731 ms
64 bytes from 172.31.25.148: icmp_seq=2 ttl=255 time=0.487 ms
64 bytes from 172.31.25.148: icmp_seq=3 ttl=255 time=0.478 ms
64 bytes from 172.31.25.148: icmp_seq=4 ttl=255 time=0.462 ms
64 bytes from 172.31.25.148: icmp_seq=5 ttl=255 time=0.570 ms
64 bytes from 172.31.25.148: icmp_seq=6 ttl=255 time=0.498 ms
```

25°C Partly sunny 06:32 05-04-2023

```
C:\WINDOWS\system32\cmd. x Command Prompt x + v
64 bytes from 172.31.25.148: icmp_seq=832 ttl=255 time=0.457 ms
64 bytes from 172.31.25.148: icmp_seq=833 ttl=255 time=0.535 ms
64 bytes from 172.31.25.148: icmp_seq=834 ttl=255 time=0.480 ms
64 bytes from 172.31.25.148: icmp_seq=835 ttl=255 time=0.470 ms
64 bytes from 172.31.25.148: icmp_seq=836 ttl=255 time=0.421 ms
64 bytes from 172.31.25.148: icmp_seq=837 ttl=255 time=0.763 ms
64 bytes from 172.31.25.148: icmp_seq=838 ttl=255 time=0.586 ms
64 bytes from 172.31.25.148: icmp_seq=839 ttl=255 time=0.440 ms
64 bytes from 172.31.25.148: icmp_seq=840 ttl=255 time=0.626 ms
64 bytes from 172.31.25.148: icmp_seq=841 ttl=255 time=0.498 ms
64 bytes from 172.31.25.148: icmp_seq=842 ttl=255 time=0.454 ms
64 bytes from 172.31.25.148: icmp_seq=843 ttl=255 time=0.443 ms
64 bytes from 172.31.25.148: icmp_seq=844 ttl=255 time=0.567 ms
^Z
[1]+ Stopped ping 172.31.25.148
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem ansible_bkp configs group_vars id_rsa.pub inventories Jenkinsfile main.yml README.md roles test
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd inventories/
[user1@ip-172-31-26-141 inventories]$ ls
dev prod
[user1@ip-172-31-26-141 inventories]$ cd prod/
[user1@ip-172-31-26-141 prod]$ ls
hosts
[user1@ip-172-31-26-141 prod]$ vi hosts
[user1@ip-172-31-26-141 prod]$ cd ..
[user1@ip-172-31-26-141 inventories]$ cd ..
```

Now, execute the playbook using ad-hoc command “ansible-playbook yaml_filename.yml -i path – user username –key-file keypair.pem -e path”

```
dev prod
[user1@ip-172-31-26-141 inventories]$ cd prod/
[user1@ip-172-31-26-141 prod]$ ls
hosts
[user1@ip-172-31-26-141 prod]$ vi hosts
[user1@ip-172-31-26-141 prod]$ cd ..
[user1@ip-172-31-26-141 inventories]$ cd ..
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/prod/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/prod.yml'

PLAY [all] ****
TASK [include_role : add_devops_user] ****
TASK [add_devops_user : Add a new user named devops] ****
[WARNING]: Platform linux on host 172.31.25.148 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.25.148]

TASK [add_devops_user : Add devops user to the sudoers] ****
ok: [172.31.25.148]
```

25°C Partly sunny 06:33 05-04-2023

```

C:\WINDOWS\system32\cmd. x Command Prompt + -
[User@ip-172-31-26-141 prod]$ vi hosts
[User@ip-172-31-26-141 prod]$ cd ..
[User@ip-172-31-26-141 inventories]$ cd ..
[User@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/prod/hosts --user ec2-user --key-file ansible_key.pem -e 'configs/prod.yml'

PLAY [all] ****
TASK [include_role : add_devops_user] ****
TASK [add_devops_user : Add a new user named devops] ****
[WARNING]: Platform linux on host 172.31.25.148 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.25.148]

TASK [add_devops_user : Add devops user to the sudoers] ****
ok: [172.31.25.148]

TASK [add_devops_user : Deploy SSH Key] ****
ok: [172.31.25.148]

TASK [add_devops_user : Disable Password Authentication] ****
ok: [172.31.25.148]

TASK [add_devops_user : Disable Root Login] ****
ok: [172.31.25.148]

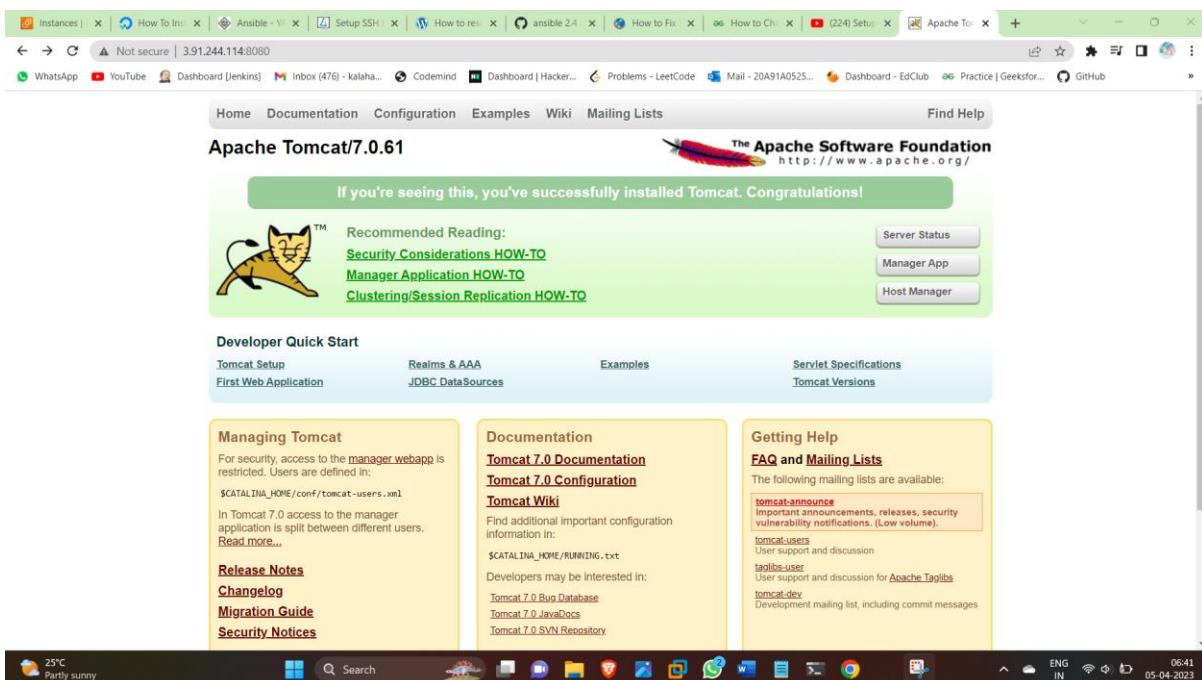
PLAY [all] ****
TASK [include_role : tomcat] ****
TASK [tomcat : Install Java 1.7] ****
ok: [172.31.25.148]

TASK [add group "tomcat"] ****
ok: [172.31.25.148]

TASK [add user "tomcat"] ****
ok: [172.31.25.148]

```

Now, browse the IP address of the slave node with the port number 8080. We can observe that the default Apache application in both the slave nodes.



```
user@ip-172-31-26-141:~/ssl$ Command Prompt + Task [tomcat : copy sample index.html file] **** ok: [172.31.25.148] Task [tomcat : Install Tomcat init script] **** ok: [172.31.25.148] Task [tomcat : Start Tomcat] **** ok: [172.31.25.148] Task [wait for tomcat to start] **** ok: [172.31.25.148] PLAY RECAP **** 172.31.25.148 : ok=19 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0 [user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/dev.yml' PLAY [all] **** TASK [include_role : add_devops_user] **** TASK [add_devops_user : Add a new user named devops] **** [WARNING]: Platform linux on host 172.31.25.148 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. ok: [172.31.25.148] TASK [add_devops_user : Add devops user to the sudoers] **** ok: [172.31.25.148] TASK [add_devops_user : Deploy SSH Key] **** ok: [172.31.25.148] TASK [add_devops_user : Disable Password Authentication] **** ok: [172.31.25.148] TASK [add_devops_user : Disable Root Login] **** ok: [172.31.25.148] PLAY [all] ****
```

This is dev environment

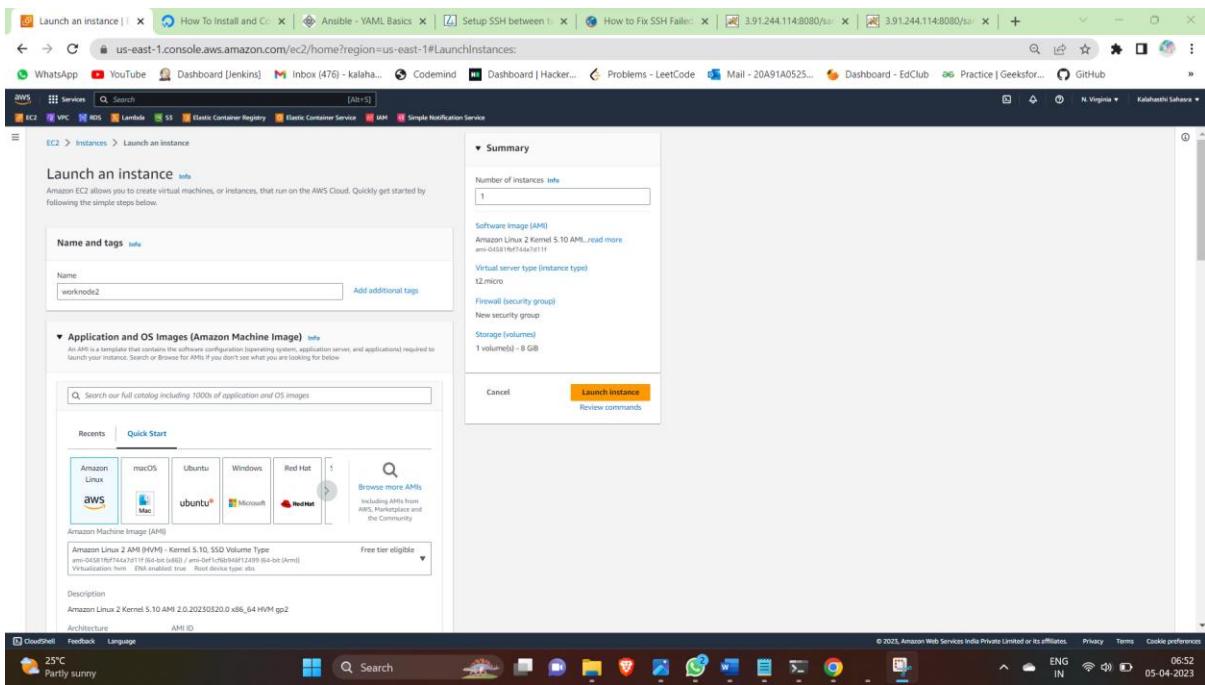


This is prd environment



We have successfully configured and deployed the sample ansible application.

Step 8: Now, we are launching another slave node



The screenshot shows the AWS CloudShell interface with multiple tabs open. The main focus is on the EC2 instance configuration page. Key details include:

- Security group rule 4 (All, All, 0.0.0.0/0):** Type: All traffic, Protocol: All, Port range: All.
- Software Image (AMI):** Amazon Linux 2 Kernel 5.10 AMI.
- Virtual server type (instance type):** t2.micro
- Configure storage:** 1x 8 GB gp2 Root volume (Not encrypted).
- Advanced details:** Advanced settings for the instance.

The 'Launch instance' button is prominently displayed at the bottom right of the configuration panel.

The screenshot shows the AWS CloudShell interface with multiple tabs open. The main focus is on the 'Success' message indicating the successful launch of the instance (ID: i-0525842a530fb2b53). Below this, a 'Next Steps' section provides various actions:

Create billing and free tier usage alerts	Connect to your instance	Connect an RDS database	Create EBS snapshot policy	Manage detailed monitoring	Create Load Balancer
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.	Once your instance is running, log into it from your local computer.	Configure the connection between an EC2 instance and a database to allow traffic flow between them.	Create a policy that automates the creation, retention, and deletion of EBS snapshots.	Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.	Create a application, network gateway or classic Elastic Load Balancer
Create AWS budget	Manage CloudWatch alarms	Get instance screenshot	Get system log	Change shutdown behavior	AWS-ConfigureCloudWatchOnEC2instance Automation
AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.	Create or update Amazon CloudWatch alarms for the instance.	Capture a screenshot from the instance and view it as an image. This is useful for troubleshooting an unresponsive instance.	View the instance's system log to troubleshoot issues.	Change the behavior of the instance for when you initiate a shutdown from the operating system of the instance itself.	The AWS-ConfigureCloudWatchOnEC2instance Automation document enables or disables CloudWatch monitoring on an EC2 instance.

The 'Create AWS budget' button is highlighted in the first column.

Successfully created the work node sample instance.

Step 9: Now we will add private IP address to the hosts file in both dev and prod environment.

```
user@ip-172-31-26-141:~/ssh
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sahasra>cd downloads

C:\Users\Sahasra\Downloads>ssh -i "keypair.pem" ec2-user@ec2-54-158-3-178.compute-1.amazonaws.com
Last login: Wed Apr  5 01:04:17 2023 from 157.48.253.244
  _.-|_--|_
  |  (   /  Amazon Linux 2 AMI
  \_\_\_\_\_\_\_|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-141 ~]$ sudo su user1
[user1@ip-172-31-26-141 ec2-user]$ cd ~/.ssh
[user1@ip-172-31-26-141 .ssh]$ ls
Ansible-Sample-Application-Deployment  id_rsa  id_rsa.pub  known_hosts
[user1@ip-172-31-26-141 .ssh]$ cd Ansible-Sample-Application-Deployment
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem  configs  id_rsa.pub  Jenkinsfile  README.md  test
ansible_bkp  group_vars  inventories  main.yml  roles
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd roles
bash: cd: roles: No such file or directory
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd roles
[user1@ip-172-31-26-141 roles]$ ls
add_devops_user  tomcat
[user1@ip-172-31-26-141 roles]$ cd inventories/
bash: cd: inventories/: No such file or directory
[user1@ip-172-31-26-141 roles]$ cd ..
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem  configs  id_rsa.pub  Jenkinsfile  README.md  test
ansible_bkp  group_vars  inventories  main.yml  roles
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd inventories/
[user1@ip-172-31-26-141 inventories]$ ls
dev  prod
[user1@ip-172-31-26-141 inventories]$ cd dev
[user1@ip-172-31-26-141 dev]$ vi hosts
[user1@ip-172-31-26-141 dev]$ cd ..
[user1@ip-172-31-26-141 dev]$ 

32°C  Sunny  Search  ENG IN  10:52  05-04-2023
```

```
[user1@ip-172-31-26-141 dev]$ vi hosts
[user1@ip-172-31-26-141 dev]$ cd ..
[user1@ip-172-31-26-141 inventories]$ vi hosts
[user1@ip-172-31-26-141 inventories]$ ls
dev  hosts  prod
[user1@ip-172-31-26-141 inventories]$ cd prod
[user1@ip-172-31-26-141 prod]$ ls
hosts
[user1@ip-172-31-26-141 prod]$ vi hosts
[user1@ip-172-31-26-141 prod]$ 4L, 35B written
[user1@ip-172-31-26-141 prod]$ cd ..
[user1@ip-172-31-26-141 inventories]$ cd ..
```

```
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_aut.pem  ansible_bkp  configs  group_vars  id_rsa.pub  inventories  Jenkinsfile  main.yml  README.md  roles  test
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd inventories
[user1@ip-172-31-26-141 inventories]$ ls
dev  hosts  prod
[user1@ip-172-31-26-141 inventories]$ cd dev
[user1@ip-172-31-26-141 dev]$ ls
hosts
[user1@ip-172-31-26-141 dev]$ cat hosts
[all]
172.31.25.148
172.31.86.100

[user1@ip-172-31-26-141 dev]$ cd ..
[user1@ip-172-31-26-141 inventories]$ ls
dev  hosts  prod
[user1@ip-172-31-26-141 inventories]$ cd prod
[user1@ip-172-31-26-141 prod]$ ls
hosts
[user1@ip-172-31-26-141 prod]$ vi hosts
[user1@ip-172-31-26-141 prod]$ cat hosts
[all]
172.31.25.148
172.31.86.100

[user1@ip-172-31-26-141 prod]$
```

Step 11: Now we will execute the playbook code using ad-hoc command and repeat the process.

```
[user@ip-172-31-26-141 prod]$ 4L_35B written
[user@ip-172-31-26-141 prod]$ cd ..
[user@ip-172-31-26-141 inventories]$ cd ..
[user@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/dev.yml'

PLAY [all] ****
TASK [include_role : add_devops_user] ****
TASK [add_devops_user : Add a new user named devops] ****
The authenticity of host '172.31.86.100 (172.31.86.100)' can't be established.
EDSA key fingerprint is SHA256:c9rGKorMyNA/tfsVGz6djh9CP8zZPvrmEY78pcp8d9v4.ECDSA key fingerprint is MDS:7f:1d:91:c2:79:d3:49:4e:5a:c1:17:ce:32:ba:9e:78.
Are you sure you want to continue connecting (yes/no)? [WARNING]: Platform linux on host 172.31.25.148 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.25.148]
yes
[WARNING]: Platform linux on host 172.31.86.100 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
changed: [172.31.86.100]

TASK [add_devops_user : Add devops user to the sudoers] ****
ok: [172.31.25.148]
changed: [172.31.86.100]

TASK [add_devops_user : Deploy SSH Key] ****
ok: [172.31.25.148]
changed: [172.31.25.148]
```



```
user@ip-172-31-26-141:~/.ssh ~ + v
ok: [172.31.25.148]

TASK [add_devops_user : Disable Root Login] ****
ok: [172.31.25.148]
changed: [172.31.86.100]

RUNNING HANDLER [add_devops_user : restart ssh] ****
changed: [172.31.86.100]

PLAY [all] ****
TASK [include_role : tomcat] ****
TASK [tomcat : Install Java 1.7] ****
ok: [172.31.25.148]
changed: [172.31.86.100]

TASK [add group "tomcat"] ****
ok: [172.31.25.148]
changed: [172.31.86.100]

TASK [add user "tomcat"] ****
ok: [172.31.25.148]
changed: [172.31.86.100]

TASK [tomcat : Download Tomcat] ****
ok: [172.31.25.148]
changed: [172.31.86.100]

TASK [tomcat : Extract archive] ****
ok: [172.31.25.148]
[WARNING]: Consider using the unarchive module rather than running 'tar'. If
you need to use command because unarchive is insufficient you can add 'warn:false' to this command task or set 'command_warnings=False' in ansible.cfg to
get rid of this message.
changed: [172.31.86.100]

TASK [tomcat : Symlink install directory] ****
ok: [172.31.25.148]
changed: [172.31.86.100]
```



The Apache Software Foundation
http://www.apache.org/

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:

- Security Considerations HOW-TO
- Manager Application HOW-TO
- Clustering/Session Replication HOW-TO

Developer Quick Start:

- Tomcat Setup
- First Web Application
- Realms & AAA
- JDBC DataSources
- Examples
- Servlet Specifications
- Tomcat Versions

Managing Tomcat:

For security, access to the `manager_webapp` is restricted. Users are defined in: `$CATALINA_HOME/conf/tomcat-users.xml`. In Tomcat 7.0 access to the manager application is split between different users. [Read more...](#)

Release Notes

Changelog

Migration Guide

Security Notices

Documentation:

- Tomcat 7.0 Documentation
- Tomcat 7.0 Configuration
- Tomcat Wiki

Find additional important configuration information in: `$CATALINA_HOME/RUNNING.txt`

Developers may be interested in:

- Tomcat 7.0 Bug Database
- Tomcat 7.0 JavaDocs
- Tomcat 7.0 SVN Repository

Getting Help:

FAQ and Mailing Lists:

The following mailing lists are available:

- `tomcat-announce`: Important announcements, releases, security vulnerability notifications. (Low volume).
- `tomcat-users`: User support and discussion
- `taglibs-user`: User support and discussion for [Apache Taglibs](#)
- `tomcat-dev`: Development mailing list, including commit messages

This is dev environment



```
[user1@ip-172-31-26-141:~/ssl] + - x
[user1@ip-172-31-26-141 inventories]$ ls
dev hosts prod
[user1@ip-172-31-26-141 inventories]$ cd dev
[user1@ip-172-31-26-141 dev]$ ls
hosts
[user1@ip-172-31-26-141 dev]$ cat hosts
[all]
172.31.25.148
172.31.86.100

[user1@ip-172-31-26-141 dev]$ cd ..
[user1@ip-172-31-26-141 inventories]$ ls
dev hosts prod
[user1@ip-172-31-26-141 inventories]$ cd prod
[user1@ip-172-31-26-141 prod]$ ls
hosts
[user1@ip-172-31-26-141 prod]$ vi hosts
[user1@ip-172-31-26-141 prod]$ cat hosts
[all]
172.31.25.148
172.31.86.100

[user1@ip-172-31-26-141 prod]$ ansible-playbook main.yml -i inventories/prod/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/prod.yml'
ERROR! the playbook: main.yml could not be found
[user1@ip-172-31-26-141 prod]$ cd ..
[user1@ip-172-31-26-141 inventories]$ cd ..
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/prod/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/prod.yml'

PLAY [all] ****
TASK [include_role : add_devops_user] ****
TASK [add_devops_user : Add a new user named devops] ****
[WARNING]: Platform linux on host 172.31.86.100 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.86.100]
[WARNING]: Platform linux on host 172.31.25.148 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.25.148]

32°C
Sunny
ENG IN 10:59
05-04-2023
```

```
[user1@ip-172-31-26-141:~/ssl] + - x
changed: [172.31.86.100]
ok: [172.31.25.148]

TASK [tomcat : Configure Tomcat server] ****
ok: [172.31.86.100]
ok: [172.31.25.148]

TASK [tomcat : Configure Tomcat users] ****
ok: [172.31.25.148]
ok: [172.31.86.100]

TASK [tomcat : Create sample directory] ****
ok: [172.31.86.100]
ok: [172.31.25.148]

TASK [tomcat : copy sample index.html file] ****
changed: [172.31.86.100]
changed: [172.31.25.148]

TASK [tomcat : Install Tomcat init script] ****
ok: [172.31.25.148]
ok: [172.31.86.100]

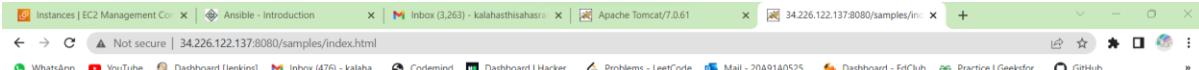
TASK [tomcat : Start Tomcat] ****
ok: [172.31.86.100]
ok: [172.31.25.148]

TASK [wait for tomcat to start] ****
ok: [172.31.86.100]
ok: [172.31.25.148]

RUNNING HANDLER [restart tomcat] ****
changed: [172.31.86.100]
changed: [172.31.25.148]

PLAY RECAP ****
172.31.25.148 : ok=20 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
172.31.86.100 : ok=20 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$
```



This is prd environment



Step 12: Now, we are modifying the dev.yml and prod.yml code and checked whether it was reflected on the slave nodes or not.

```
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ls
ansible_automount_bkp configs group_vars id_rsa.pub inventories Jenkinsfile main.yml README.md roles test
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ cd configs/
[user1@ip-172-31-26-141 configs]$ ls
dev.yml prod.yml
[user1@ip-172-31-26-141 configs]$ vi dev.yml
[user1@ip-172-31-26-141 configs]$ cat dev.yml
index_page: This is Ansible Army!!!!
devops_password: #1zlpYY9

[user1@ip-172-31-26-141 configs]$ cd ..
```

Step 13: Execute the playbook using the ad-hoc command.

```
[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_authorized_keys -e '@configs/dev.yml'

PLAY [all] ****
TASK [include_role : add_devops_user] ****
TASK [add_devops_user : Add a new user named devops] ****
[WARNING]: Platform linux on host 172.31.25.148 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.25.148]
[WARNING]: Platform linux on host 172.31.86.100 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.86.100]

TASK [add_devops_user : Add devops user to the sudoers] ****
ok: [172.31.25.148]
ok: [172.31.86.100]

TASK [add_devops_user : Deploy SSH Key] ****
ok: [172.31.86.100]
ok: [172.31.25.148]

TASK [add_devops_user : Disable Password Authentication] ****
ok: [172.31.86.100]
ok: [172.31.25.148]

TASK [add_devops_user : Disable Root Login] ****
ok: [172.31.25.148]
```

```

user@ip-172-31-26-141:~/ssl
ok: [172.31.86.100]
ok: [172.31.25.148]

TASK [tomcat : Configure Tomcat server] ****
ok: [172.31.25.148]
ok: [172.31.86.100]

TASK [tomcat : Configure Tomcat users] ****
ok: [172.31.86.100]
ok: [172.31.25.148]

TASK [tomcat : Create sample directory] ****
ok: [172.31.86.100]
ok: [172.31.25.148]

TASK [tomcat : copy sample index.html file] ****
changed: [172.31.25.148]
changed: [172.31.86.100]

TASK [tomcat : Install Tomcat init script] ****
ok: [172.31.25.148]
ok: [172.31.86.100]

TASK [tomcat : Start Tomcat] ****
ok: [172.31.86.100]
ok: [172.31.25.148]

TASK [wait for tomcat to start] ****
ok: [172.31.25.148]
ok: [172.31.86.100]

RUNNING HANDLER [restart tomcat] ****
changed: [172.31.25.148]
changed: [172.31.86.100]

PLAY RECAP ****
172.31.25.148 : ok=20 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
172.31.86.100 : ok=20 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[user1@ip-172-31-26-141 Ansible-Sample-Application-Deployment]$
```

Step 14: The modification that we have done on the master node get reflected in the one slave node.



This is Ansible Army!!!!



And also reflected on the second node that we have added recently to the master node



This is Ansible Army!!!!

Index.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Simple HTML HomePage</title>
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">
<style>
  @import url('https://fonts.googleapis.com/css2?family=Sriracha&display=swap');
  body {
    margin: 0;
    box-sizing: border-box;
  }
  /* CSS for header */
  .header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    background-color: #f5f5f5;
  }
  .header .logo {
    font-size: 25px;
    font-family: 'Sriracha', cursive;
    color: #000;
    text-decoration: none;
    margin-left: 30px;
  }
  .nav-items {
    display: flex;
    justify-content: space-around;
    align-items: center;
    background-color: #f5f5f5;
    margin-right: 20px;
  }
  .nav-items a {
    text-decoration: none;
    color: #000;
    padding: 35px 20px;
  }
  /* CSS for main element */
  .intro {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    width: 100%;
    height: 520px;
    background: linear-gradient(to bottom, rgba(0, 0, 0, 0.5) 0%, rgba(0, 0, 0, 0.5) 100%), url("https://images.unsplash.com/photo-1587620962725-abab7fe55159?ixlib=rb-1.2.1&ixid=MnwxA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1031&q=80");
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
  }

```

```
.intro h1 {
    font-family: sans-serif;
    font-size: 60px;
    color: #fff;
    font-weight: bold;
    text-transform: uppercase;
    margin: 0;
}
.intro p {
    font-size: 20px;
    color: #d1d1d1;
    text-transform: uppercase;
    margin: 20px 0;
}
.intro button {
    background-color: #5edaf0;
    color: #000;
    padding: 10px 25px;
    border: none;
    border-radius: 5px;
    font-size: 20px;
    font-weight: bold;
    cursor: pointer;
    box-shadow: 0px 0px 20px rgba(255, 255, 255, 0.4)
}
.achievements {
    display: flex;
    justify-content: space-around;
    align-items: center;
    padding: 40px 80px;
}
.achievements .work {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    padding: 0 40px;
}
.achievements .work i {
    width: fit-content;
    font-size: 50px;
    color: #333333;
    border-radius: 50%;
    border: 2px solid #333333;
    padding: 12px;
}
.achievements .work .work-heading {
    font-size: 20px;
    color: #333333;
    text-transform: uppercase;
```

```
margin: 10px 0;
}

.achievements .work .work-text {
  font-size: 15px;
  color: #585858;
  margin: 10px 0;
}

.about-me {
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 40px 80px;
  border-top: 2px solid #eeeeee;
}

.about-me img {
  width: 500px;
  max-width: 100%;
  height: auto;
  border-radius: 10px;
}

.about-me-text h2 {
  font-size: 30px;
  color: #333333;
  text-transform: uppercase;
  margin: 0;
}

.about-me-text p {
  font-size: 15px;
  color: #585858;
  margin: 10px 0;
}

/* CSS for footer */

.footer {
  display: flex;
  justify-content: space-between;
  align-items: center;
  background-color: #302f49;
  padding: 40px 80px;
}

.footer .copy {
  color: #fff;
}

.bottom-links {
  display: flex;
  justify-content: space-around;
  align-items: center;
  padding: 40px 0;
}

.bottom-links .links {
  display: flex;
```

```

flex-direction: column;
justify-content: center;
align-items: center;
padding: 0 40px;
}
.bottom-links .links span {
  font-size: 20px;
  color: #fff;
  text-transform: uppercase;
  margin: 10px 0;
}
.bottom-links .links a {
  text-decoration: none;
  color: #a1a1a1;
  padding: 10px 20px;
}

```

</style>

</head>

<body>

<header class="header">

 Ansible army

 <nav class="nav-items">

 Home

 About

 Contact

 </nav>

</header>

<main>

 <div class="intro">

 <h1>DevOps Team</h1>

 <p> I am DevOps Engineer</p>

 <button>More Details</button>

 </div>

 <div class="achievements">

 <div class="work">

 <i class="fas fa-atom"></i>

 <p class="work-heading">Projects</p>

 <p class="work-text">I have worked on many projects and I am very proud of them. I am a very good developer and I am always looking for new projects.</p>

 </div>

 <div class="work">

 <i class="fas fa-skiing"></i>

 <p class="work-heading">Skills</p>

 <p class="work-text">I have a lot of skills and I am very good at them. I am very good at programming and I am always looking for new skills.</p>

 </div>

 <div class="work">

 <i class="fas fa-ethernet"></i>

 <p class="work-heading">Network</p>

```

<p class="work-text">I have a lot of network skills and I am very good at them. I am very good at networking and I am always looking for new network skills.</p>
</div>
</div>
<div class="about-me">
<div class="about-me-text">
<h2>About Me</h2>
<p>I am a web developer and I love to create websites. I am a very good developer and I am always looking for new projects. I am a very good developer and I am always looking for new projects.</p>
</div>

</div>
</main>
<footer class="footer">
<div class="copy">&copy; 2022 Developer</div>
<div class="bottom-links">
<div class="links">
<span>More Info</span>
<a href="#">Home</a>
<a href="#">About</a>
<a href="#">Contact</a>
</div>
<div class="links">
<span>Social Links</span>
<a href="#"></a>
<a href="#"></a>
<a href="#"></a>
</div>
</div>
</footer>
</body>
</html>

```

Step 15: The given html code will be get reflected in the one slave node.

```

[User@ip-172-31-26-148 tomcat]$ cd templates/
[User@ip-172-31-26-148 tomcat]$ ls -l
index.html.js  server.xml tomcat-users.xml
[User@ip-172-31-26-148 tomcat]$ cat index.html.js
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Simple HTML Home Page</title>
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" rel="stylesheet">
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Sriracha&display=swap');
    </style>
</head>
<body>
    <div style="text-align: center; margin-top: 50px;>
        <h1>Hello World!</h1>
        <p>This is a simple static website generated by Ansible. It's running on port 8080. You can access it at http://172.31.26.148.</p>
    </div>
</body>
</html>
[User@ip-172-31-26-148 templates]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/dev.yml'
[User@ip-172-31-26-148 templates]$ cd ..
[User@ip-172-31-26-148 roles]$ cd ..
[User@ip-172-31-26-148 Ansible-Sample-Application-Deployment]$ ansible-playbook main.yml -i inventories/dev/hosts --user ec2-user --key-file ansible_aut.pem -e '@configs/dev.yml'

PLAY [all] *****
  TASK [include_role : add_devops_user] *****
  TASK [add_devops_user : Add a new user named devops] *****
[WARNING]: Platform linux on host 172.31.86.100 is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.
[WARNING]: Platform linux on host 172.31.25.148 is using the discovered Python interpreter at /usr/bin/python, but
future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.
ok: [172.31.86.100]
ok: [172.31.25.148]

```

Instances (1/4) [Info](#)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
Master_Node	i-013141f9e7ef5fd5	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-54-158-3-178.com...	54.158.3.17
worknode1	i-026e8092f9217ec54	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-51-244-114.com...	3.91.244.11
worknode2	i-0525842a538fb2b33	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1d	ec2-34-226-122-137.co...	34.226.122.
JenkinsDemol...	i-0dcbe8fe957eda0af0	Stopped	t2.micro	-	No alarms	+ us-east-1c	-	-

Instance: i-026e8092f9217ec54 (worknode1)

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status checks](#) [Monitoring](#) [Tags](#)

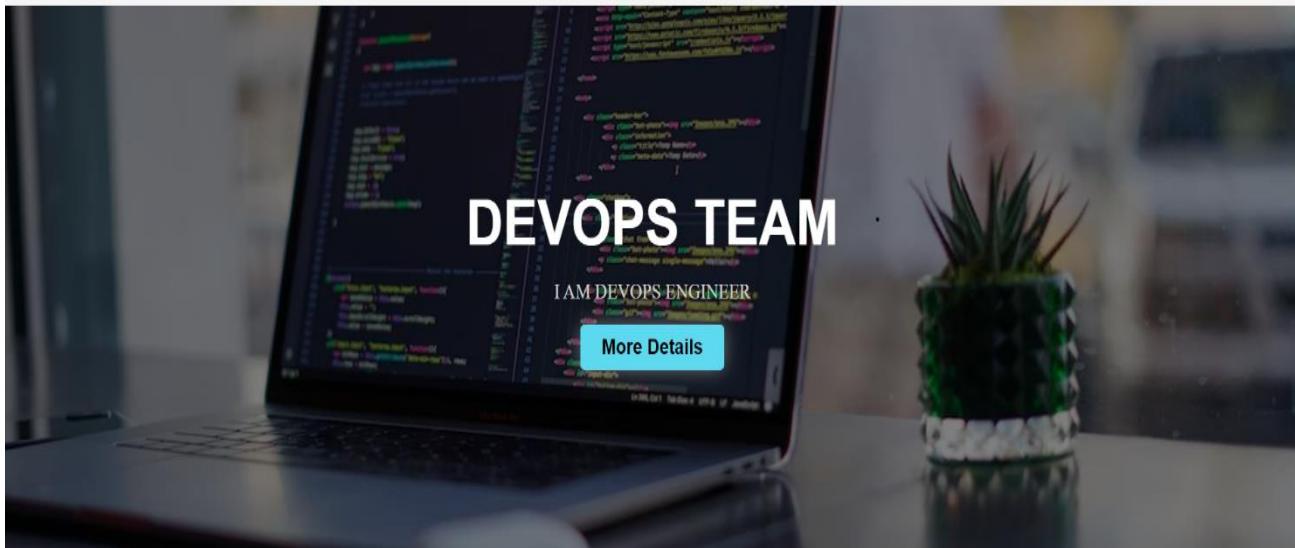
[Instance summary](#) [Info](#)

Instance ID i-026e8092f9217ec54 (worknode1)	Public IPv4 address 3.91.244.114 [open address]	Private IPv4 addresses 172.31.25.148
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-91-244-114.compute-1.amazonaws.com [open address]
Hostname type IP name: ip-172-31-25-148.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-25-148.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations.
Auto-assigned IP address 3.91.244.114 [Public IP]	VPC ID vpc-071fb87dd704305c9	

[WhatsApp](#) [YouTube](#) [Dashboard | Jenkins](#) [Inbox \(476\) - kalaha...](#) [Codemind](#) [Dashboard | Hacker...](#) [Problems - LeetCode](#) [Mail - 20A91A0525...](#) [Dashboard - EdClub](#) [Practice | Geeksfor...](#) [GitHub](#)

An Ansible army

[Home](#) [About](#) [Contact](#)

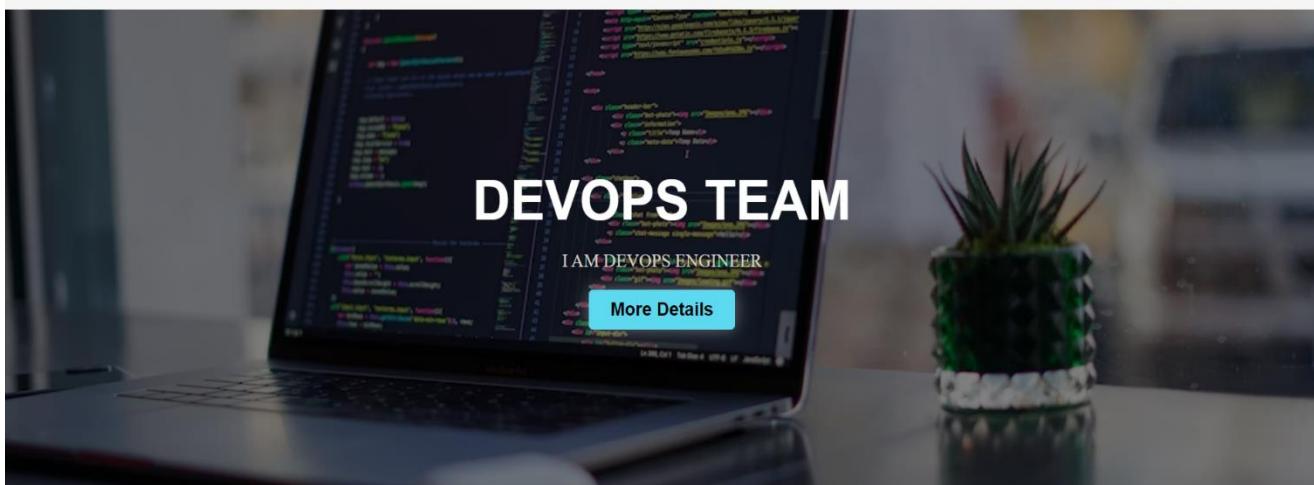


Screenshot of the AWS EC2 Instances page showing a list of running instances. The selected instance is 'worknode2' (i-0525842a538fb2b33). The instance details page is open, showing the instance summary, status checks (2/2 checks passed), monitoring, and tags.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
Master_Node	i-013141f9e7e5f0fd5	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-158-3-178.com...	54.158.3.1
worknode1	i-026e8092f9217ec54	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-3-91-244-114.com...	3.91.244.1
worknode2	i-0525842a538fb2b33	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	ec2-34-226-122-137.co...	34.226.122.137
JenkinsDemol...	i-0dcebfe957eda0afdf	Stopped	t2.micro	-	No alarms	us-east-1c	-	-

[Ansible army](#)

[Home](#) [About](#) [Contact](#)



REPOSITORY DETAILS:

The frontend sample clone web application and YAML Playbooks we used in this project will be available in the following git hub repository

<https://github.com/sahasrakalahasti/Ansible-Army-Team-06>

OUTCOME

At the end of this project, the master node is connected with worker/slave nodes. The web application has been deployed in the master node and it reflected in the slave nodes private Ip which has been added in the hosts. The changes updated in the master node are depicted in the slave nodes. This represents the automation of application deployment.

Enhanced scalability: Ansible is highly scalable and can automate tasks across large numbers of servers and devices. This makes it a valuable tool for organizations that need to manage complex IT environments. Ansible allows you to define groups of hosts and execute tasks across those groups, allowing you to easily scale your automation efforts as your environment grows.

Reduced costs: By automating routine tasks, Ansible can help organizations save time and reduce labour costs associated with manual processes. Ansible can also help organizations to avoid costly downtime by automating tasks such as system patching and updates.

CONCLUSION

Project configuration management with Ansible involves using Ansible, an automation tool, to manage configurations and tasks across multiple systems. The purpose is to automate routine tasks, minimize errors, and increase efficiency in IT operations. To work on project configuration management with Ansible, you need to understand operating systems, networking, scripting, infrastructure as code, and containerization.

Additionally, Ansible can help organizations maintain a strong security posture by automating tasks such as patch management, vulnerability scanning, and compliance auditing. By leveraging Ansible's capabilities, organizations can achieve greater efficiency, cost savings, and security, while freeing up IT teams to focus on more strategic work.

You will also need to have access to the necessary tools, such as Ansible itself, inventory files, playbooks, modules, roles, and version control systems. The benefits include reducing manual intervention, improving consistency and accuracy, and enabling greater agility and scalability in IT operations.

REFERENCES LINK:

Intellipaat: <https://intellipaat.com/blog/tutorial/devops-tutorial/ansible-tutorial/#:~:text=Ansible%20is%20an%20open%2Dsource,efficient%2C%20reliable%2C%20and%20powerful>

Ansible documentation: <https://docs.ansible.com/>

Devops4solution: <https://devops4solutions.com/setup-ssh-between-two-aws-ec2-instances-using-ansible/>

Aws documentation: <https://docs.aws.amazon.com>

CONTACT EMAIL'S:

- **YAMUNA:** 20A91A0512@aec.edu.in
- **BLESSY:** 20A91A0513@aec.edu.in
- **SAHASRA:** 20A91A0525@aec.edu.in
- **ISWARYA:** 20A91A0561@aec.edu.in