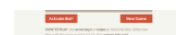


[DevOps](#)

Automate EC2 provisioning in AWS using Jenkins and Ansible Playbook

[mrcloudbook.com](#) · 8 January 2024

EC2 PROVISIONING IN AWS USING ANSIBLE PLAYBOOK AND JENKINS CI-CD

[Mr Cloud Book](#)[Home](#) [Blog](#) [DevSecOps](#) [Contact](#) [About Me](#) [Testimonials](#)[JENKINS](#)[FILE SCAN](#)

We will learn how to create new EC2 instances using the Ansible playbook and automate using Jenkins Pipeline. in the end, we will play the game of 2048.

Contents [[hide](#)]

[Launch an Ubuntu\(22.04\) T2 medium Instance](#)

[Install Jenkins and Trivy](#)

[To Install Jenkins](#)

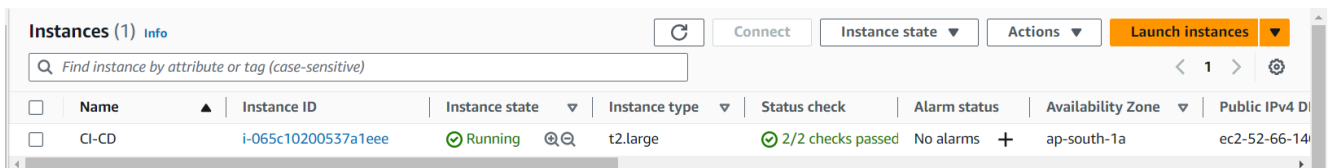
[Install Trivy](#)

[Install Ansible](#)

[Ansible Playbook](#)

Launch an Ubuntu(22.04) T2 medium Instance

Launch an AWS T2 medium Instance. Use the image as Ubuntu. You can create a new key pair or use an existing one. Enable HTTP and HTTPS settings in the Security Group.



Instances (1) Info								
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
<input type="checkbox"/>	CI-CD	i-065c10200537a1eee	Running	t2.large	2/2 checks passed	No alarms	ap-south-1a	ec2-52-66-14

Install Jenkins and Trivy

To Install Jenkins

Connect to your console, and enter these commands to Install Jenkins

```
vi jenkins.sh
```

```
#!/bin/bash
sudo apt update -y
wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public
echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.a
sudo apt update -y
```

```
sudo apt install temurin-17-jdk -y
/usr/bin/java --version
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee
    /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
    https://pkg.jenkins.io/debian-stable binary/ | sudo tee
    /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
sudo systemctl start jenkins
sudo systemctl status jenkins
```



```
sudo chmod 777 jenkins.sh
./jenkins.sh    # this will install jenkins
```



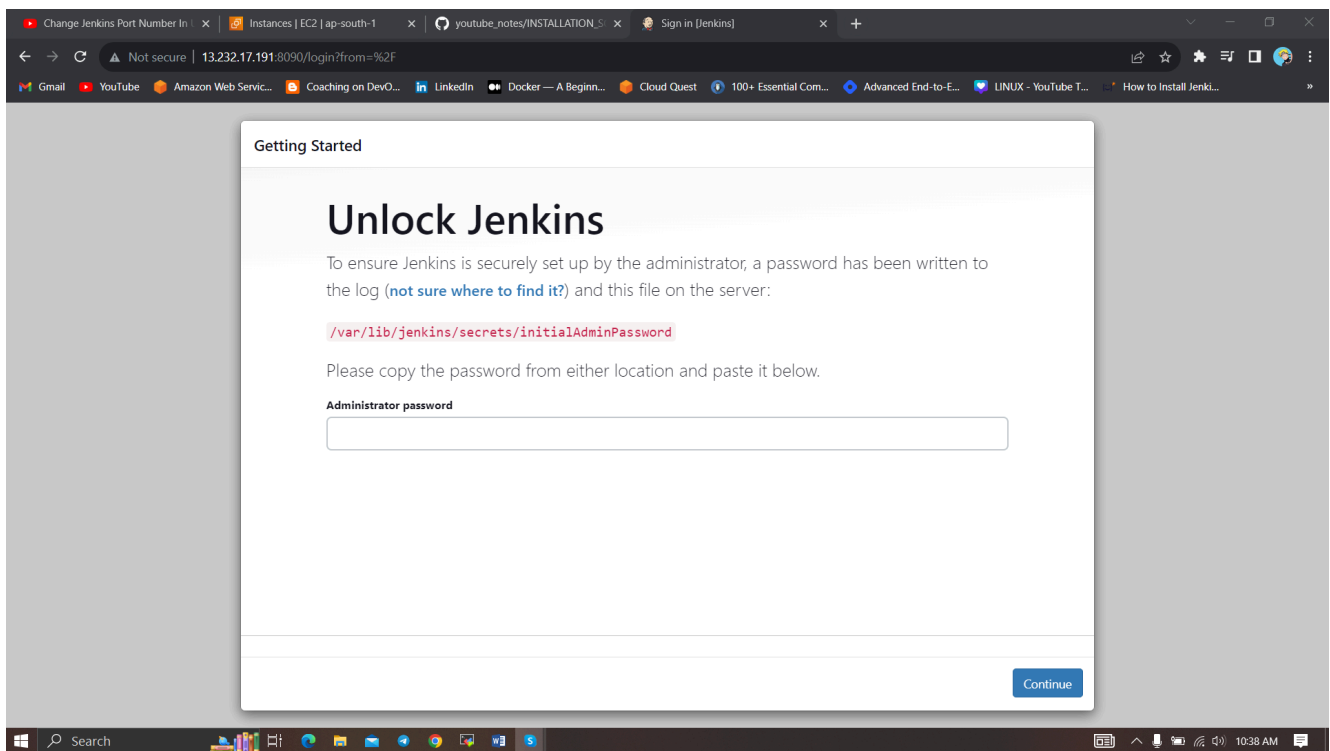
Once Jenkins is installed, you will need to go to your AWS EC2 Security Group and open Inbound Port 8080, since Jenkins works on Port 8080.

Now, grab your Public IP Address

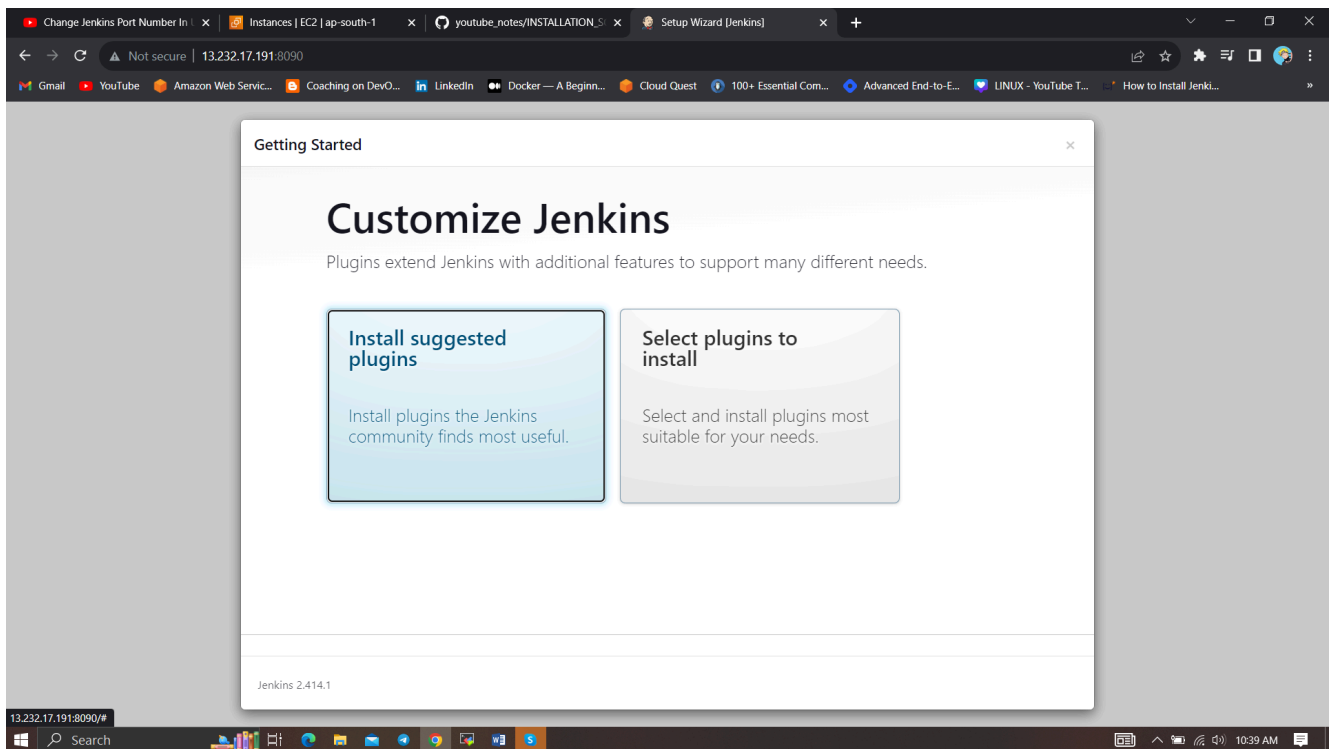


```
<EC2 Public IP Address:8080>
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

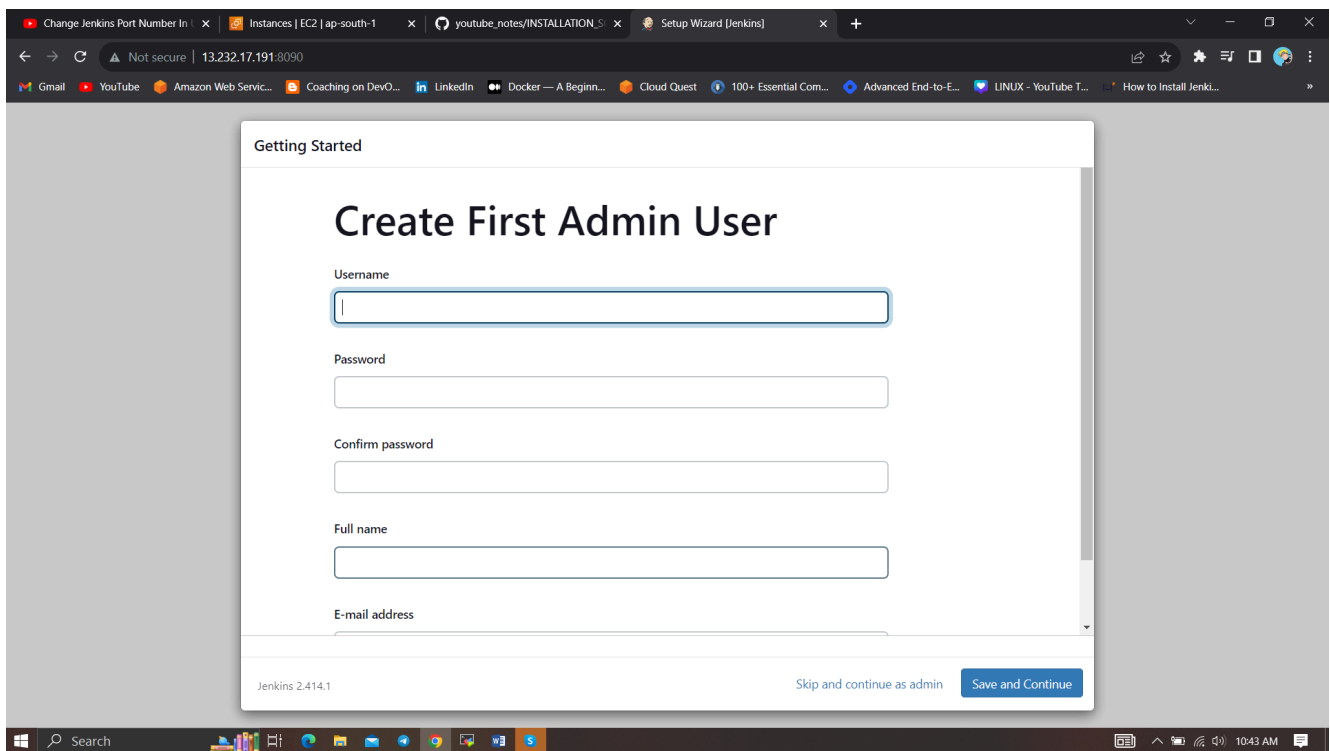




Unlock Jenkins using an administrative password and install the suggested plugins.

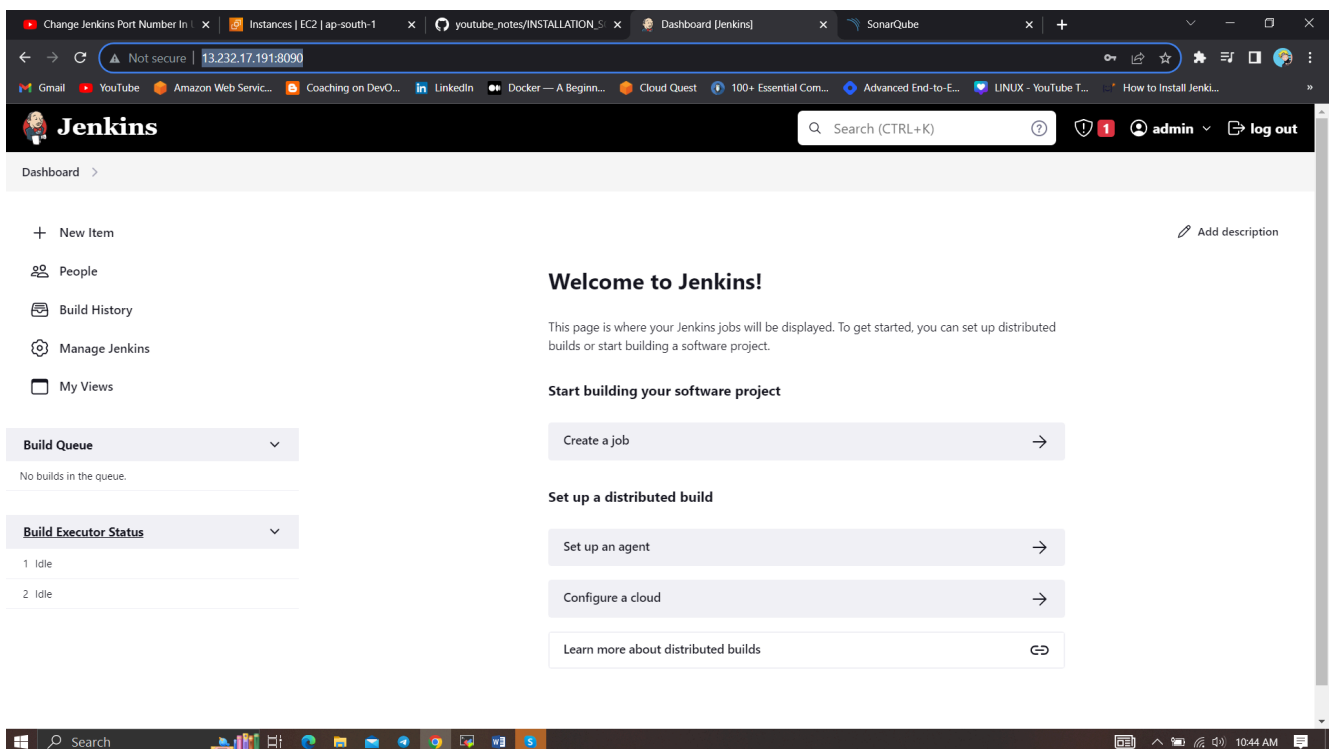


Jenkins will now get installed and install all the libraries.



Create a user click on save and continue.

Jenkins Getting Started Screen.



Install Trivy



```
vi trivy.sh
```

```
sudo apt-get install wget apt-transport-https gnupg lsb-release -y  
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | g  
echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity  
sudo apt-get update  
sudo apt-get install trivy -y
```



Install Ansible

connect to your Jenkins machine using Putty or Mobaxtreme

Now we are going to run the below commands on the Jenkins machine

Step1:Update your system packages:



```
sudo apt-get update
```

Step 2: First Install Required packages to install Ansible.



```
sudo apt install software-properties-common
```

```

Quick connect...
1. ubuntu@Ansible-master: /
2. ubuntu@Ansible-slave: ~

ubuntu@Ansible-master:/$ sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties
The following packages will be upgraded:
  python3-software-properties software-properties-common
2 upgraded, 0 newly installed, 0 to remove and 111 not upgraded.
Need to get 42.9 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 software-properties-common all 0.99.22.7 [14.1 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-software-properties all 0.99.22.7 [28.8 kB]
Fetched 42.9 kB in 0s (2106 kB/s)
(Reading database ... 102753 files and directories currently installed.)
Preparing to unpack .../software-properties-common_0.99.22.7_all.deb ...
Unpacking software-properties-common (0.99.22.7) over (0.99.22.6) ...
Preparing to unpack .../python3-software-properties_0.99.22.7_all.deb ...
Unpacking python3-software-properties (0.99.22.7) over (0.99.22.6) ...
Setting up python3-software-properties (0.99.22.7) ...
Setting up software-properties-common (0.99.22.7) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@Ansible-master:/$

```

Step3: Add the ansible repository via PPA



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



```

ubuntu@Ansible-master:/$ sudo add-apt-repository --yes --update ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
  Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications- automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/

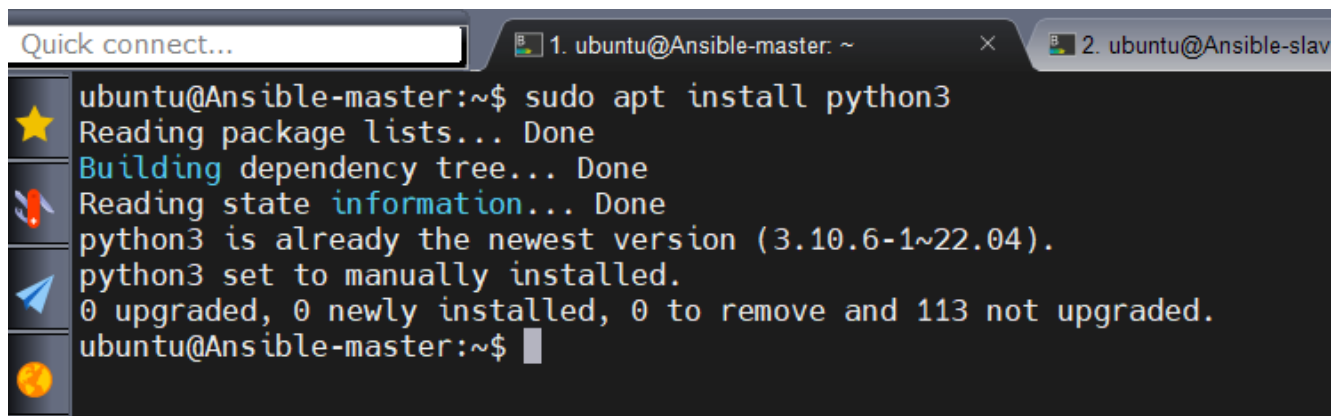
If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Adding deb entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ansible-ubuntu-ansible.gpg with fingerprint 6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease [18.0 kB]
Get:6 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 Packages [1144 B]
Get:7 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main Translation-en [752 B]
Fetched 139 kB in 2s (72.8 kB/s)
Reading package lists... Done
ubuntu@Ansible-master:/$

```

Install Python3 on Jenkins for Ansible



```
sudo apt install python3
```

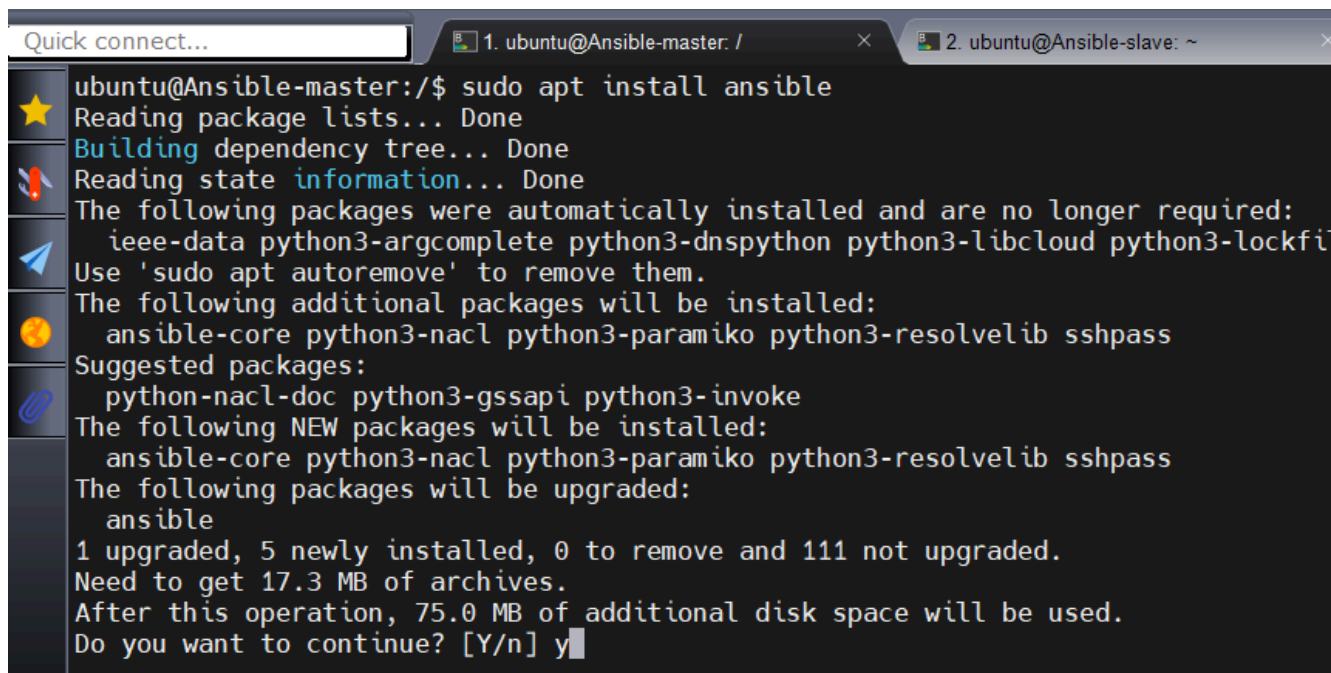


A terminal window titled 'Quick connect...' with two tabs: '1. ubuntu@Ansible-master: ~' and '2. ubuntu@Ansible-slave: ~'. The active tab shows the command 'sudo apt install python3' being executed. The output indicates that python3 is already installed as the newest version (3.10.6-1~22.04) and is set to manually installed. No packages were upgraded, installed, or removed, and 113 packages were not upgraded.

```
ubuntu@Ansible-master:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 113 not upgraded.
ubuntu@Ansible-master:~$
```

Install Ansible on Ubuntu 22.04 LTS

```
sudo apt install ansible -y
```



A terminal window titled 'Quick connect...' with two tabs: '1. ubuntu@Ansible-master: /' and '2. ubuntu@Ansible-slave: ~'. The active tab shows the command 'sudo apt install ansible' being executed. The output lists packages to be removed, additional packages to be installed (ansible-core, python3-nacl, python3-paramiko, python3-resolverlib, sshpass), suggested packages, and new packages to be installed. It also shows the disk space requirements and asks for confirmation to continue, which is answered with 'y'.

```
ubuntu@Ansible-master:/$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ieee-data python3-argcomplete python3-dnspython python3-libcloud python3-lockfile
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ansible-core python3-nacl python3-paramiko python3-resolverlib sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  ansible-core python3-nacl python3-paramiko python3-resolverlib sshpass
The following packages will be upgraded:
  ansible
1 upgraded, 5 newly installed, 0 to remove and 111 not upgraded.
Need to get 17.3 MB of archives.
After this operation, 75.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```



```
sudo apt install ansible-core
```

```
ubuntu@Ansible-master:/$ sudo apt install ansible-core
Reading package lists... done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ieee-data python3-argcomplete python3-dnspython python3-libcloud python3-lockfile
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  ansible-core
0 upgraded, 1 newly installed, 0 to remove and 111 not upgraded.
5 not fully installed or removed.
Need to get 0 B/1020 kB of archives.
After this operation, 6288 kB of additional disk space will be used.
(Reading database ... 87805 files and directories currently installed.)
Preparing to unpack .../ansible-core_2.15.2-1ppa~jammy_all.deb ...
Unpacking ansible-core (2.15.2-1ppa~jammy) ...
Setting up python3-resolvelib (0.8.1-1) ...
Setting up ansible-core (2.15.2-1ppa~jammy) ...
Setting up sshpass (1.09-1) ...
Setting up ansible (8.3.0-1ppa~jammy) ...
```

Step2: To check version :

```
ansible --version
```

Install Python-pip3:

```
sudo apt install python3-pip -y
( this is just comment - Package manager for python)Install Boto Framework
```

Install Boto Framework – AWS SDK

```
sudo pip3 install boto boto3
```

Ansible will access AWS resources using Boto SDK.

```
sudo apt-get install python3-boto -y  
pip list boto | grep boto
```

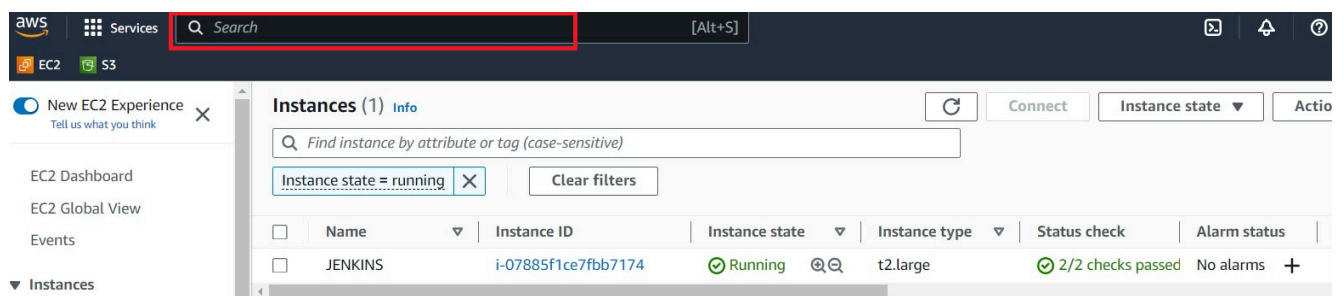
(the above command should display the below output)

```
ubuntu@ip-172-31-25-57:~$ pip list boto | grep boto  
DEPRECATION: The default format will switch to columns in the future. You  
mns) in your pip.conf under the [list] section) to disable this warning.  
boto (2.49.0)  
boto3 (1.16.7)  
botocore (1.19.7)  
ubuntu@ip-172-31-25-57:~$
```

let's create and attach an IAM role to Jenkins machines for the provision of a new ec2 instance

Navigate to **AWS CONSOLE**

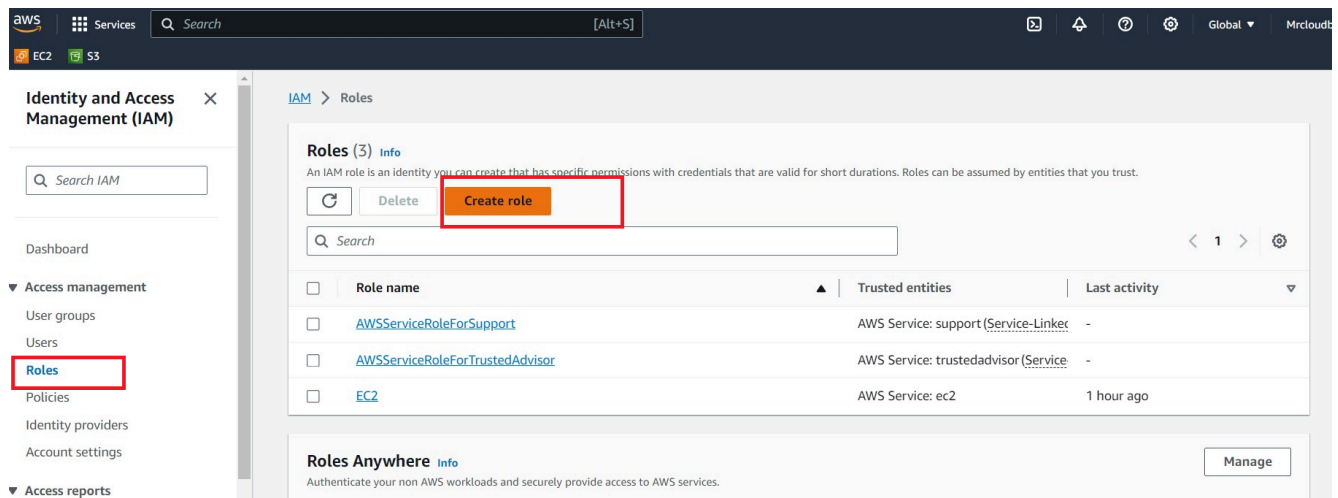
Click the "Search" field.



Type "IAM enter"

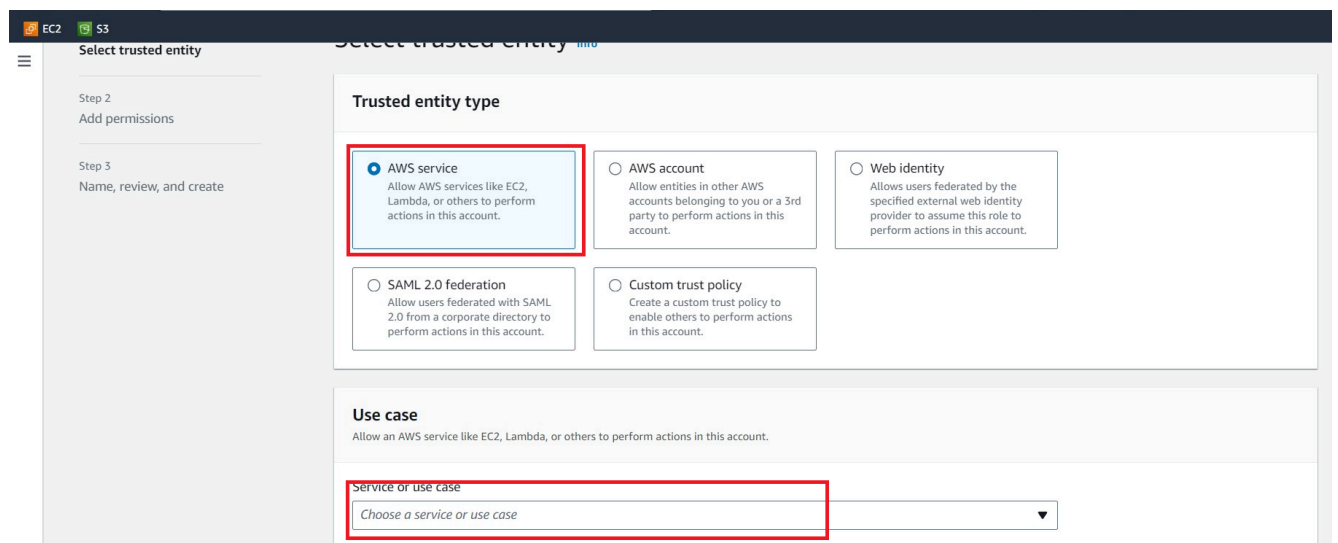
Click “Roles”

Click “Create role”



Click “AWS service”

Click “Choose a service or use case”



Click “EC2”

Click “Next”

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

EC2

Choose a use case for the specified service.

Use case

☒ EC2
Allows EC2 instances to call AWS services on your behalf.

☐ EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

☐ EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

☐ EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

☐ EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

☐ EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

☐ EC2 - Spot Fleet
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.


☐ EC2 - Scheduled Instances
Allows EC2 Scheduled Instances to manage instances on your behalf.

Cancel **Next**

Click the “Search” field.

Add permissions policies

AmazonEC2FullAccess

<input checked="" type="checkbox"/>	 AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via th...
-------------------------------------	---	-------------	--

click Next

Click the “Role name” field.

Type “Jenkins-cicd”

Click “Create role”

Click “EC2”

go to the Jenkins instance and add this role to the Ec2 instance.

select Jenkins instance -> Actions -> Security -> Modify IAM role


Add a newly created Role and click on Update IAM role.

[EC2](#) > [Instances](#) > [i-07885f1ce7fbb7174](#) > [Modify IAM role](#)

Modify IAM role [Info](#)


Attach an IAM role to your instance.



Instance ID

 [i-07885f1ce7fbb7174](#) (JENKINS)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.



 [Create new IAM role](#) 


[Cancel](#) [Update IAM role](#)


Let's go to the Jenkins machine and add the Ansible Plugin


Manage Jenkins -> Plugins -> Available Plugins


search for Ansible and install

[Dashboard](#) > [Manage Jenkins](#) > [Plugins](#)


 Updates

 Available plugins

 Installed plugins

 Advanced settings

Plugins



Name ↓

[Ansible plugin](#) 253.v4fe719ffdd8a_
Invoke [Ansible](#) Ad-Hoc commands and playbooks.
[Report an issue with this plugin](#)

Give this command in your Jenkins machine to find the path of your ansible which is used in the tool section of Jenkins.

which ansible

```
Quick connect... 1. ubuntu@ip-172-31-42-165: ~  
ubuntu@ip-172-31-42-165:~$ which ansible  
/usr/bin/ansible  
ubuntu@ip-172-31-42-165:~$
```

Copy that path and add it to the tools section of Jenkins at ansible installations.

Dashboard > Manage Jenkins > Tools

Dependency-Check installations ▾ Edited

Ansible installations

Ansible installations ^ Edited

Add Ansible

Ansible

Name

ansible

Path to ansible executables directory

/usr/bin

☐ Install automatically ?

Ansible Playbook



```
--
name: Provisioning a new EC2 instance and security group
hosts: localhost
connection: local
gather_facts: False
tags: provisioning
pre_tasks:
  - name: Gather facts
    setup:
  - name: Print python version
    debug:
      msg: "Using Python {{ ansible_python_version }}"
  - name: Install dependencies
    shell: "/usr/bin/python3.10 -m pip install {{ item }}"
    loop:
      - boto3
      - botocore
vars:
  ansible_python_interpreter: /usr/bin/python3.10
  keypair: Mumbai
  instance_type: t2.micro
  image_id: ami-0f5ee92e2d63afc18
  wait: yes
```

```
group: webserver
count: 1
region: ap-south-1
security_group: ec2-security-group
tag_name:
  Name: Aj-ec2
tasks:
- name: Create a security group
  amazon.aws.ec2_group:
    name: "{{ security_group }}"
    description: Security Group for webserver Servers
    region: "{{ region }}"
    rules:
      - proto: tcp
        from_port: 22
        to_port: 22
        cidr_ip: 0.0.0.0/0
      - proto: tcp
        from_port: 8080
        to_port: 8080
        cidr_ip: 0.0.0.0/0
      - proto: tcp
        from_port: 3000
        to_port: 3000
        cidr_ip: 0.0.0.0/0
      - proto: tcp
        from_port: 80
        to_port: 80
        cidr_ip: 0.0.0.0/0
      - proto: tcp
        from_port: 443
        to_port: 443
        cidr_ip: 0.0.0.0/0
    rules_egress:
      - proto: all
        cidr_ip: 0.0.0.0/0
  register: basic_firewall
- name: Launch the new EC2 Instance
  amazon.aws.ec2_instance:
    security_group: "{{ security_group }}"
    instance_type: "{{ instance_type }}"
    image_id: "{{ image_id }}"
    wait: "{{ wait }}"
    region: "{{ region }}"
    key_name: "{{ keypair }}"
```

```

count: "{{ count }}"
tags: "{{ tag_name }}"
user_data: |
    #!/bin/bash
    sudo apt update -y
    sudo apt install docker.io -y
    sudo systemctl start docker
    sudo systemctl enable docker
    sudo docker run -d --name 2048 -p 3000:3000 sevenajay/2048:lat
register: ec2

```

Write a sample pipeline for Provision



```

pipeline {
    agent any
    tools{
        ansible 'ansible'
    }
    stages {
        stage('cleanws') {
            steps {
                cleanWs()
            }
        }
        stage('checkout'){
            steps{
                git branch: 'main', url: 'https://github.com/Aj7Ay/ANSIBLE'
            }
        }
        stage('TRIVY FS SCAN') {
            steps {
                sh "trivy fs . > trivyfs.txt"
            }
        }
        stage('ansible provision') {
            steps {
                // To suppress warnings when you execute the playbook
                sh "pip install --upgrade requests==2.20.1"
                ansiblePlaybook playbook: 'ec2.yaml'
            }
        }
    }
}

```



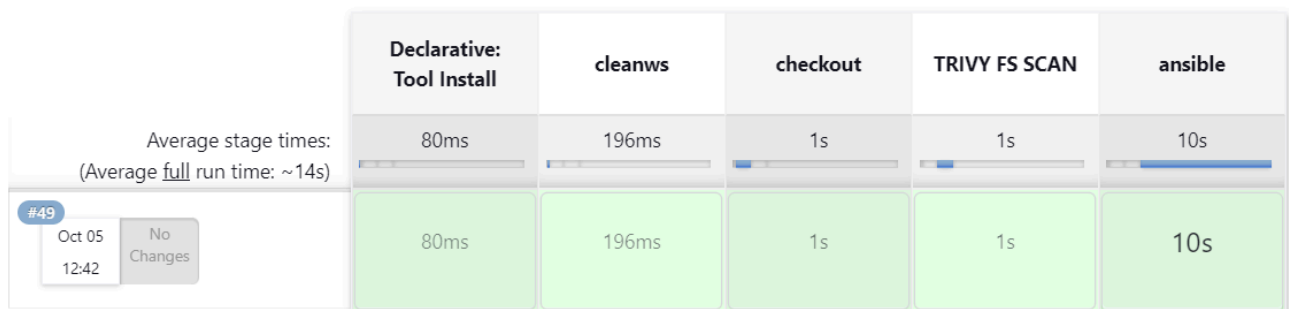
```

    }
  }
}

```

stage view

Stage View



Provision Ec2-instance

Instances (1/2) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
JENKINS	i-07885f1ce7fbb7174	Running	t2.large	2/2 checks passed	No alarms	ap-south-1a	ec2-13-232-2
Aj-ec2	i-029a4dd88f03bcb17	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-2-140

Instance: i-029a4dd88f03bcb17 (Aj-ec2)

Details Security Networking Storage Status checks Monitoring Tags

Inst: i-029a4dd88f03bcb17 (Aj-ec2)

IPv6 address -

65.2.140.71 [open address](#)

Instance state Running

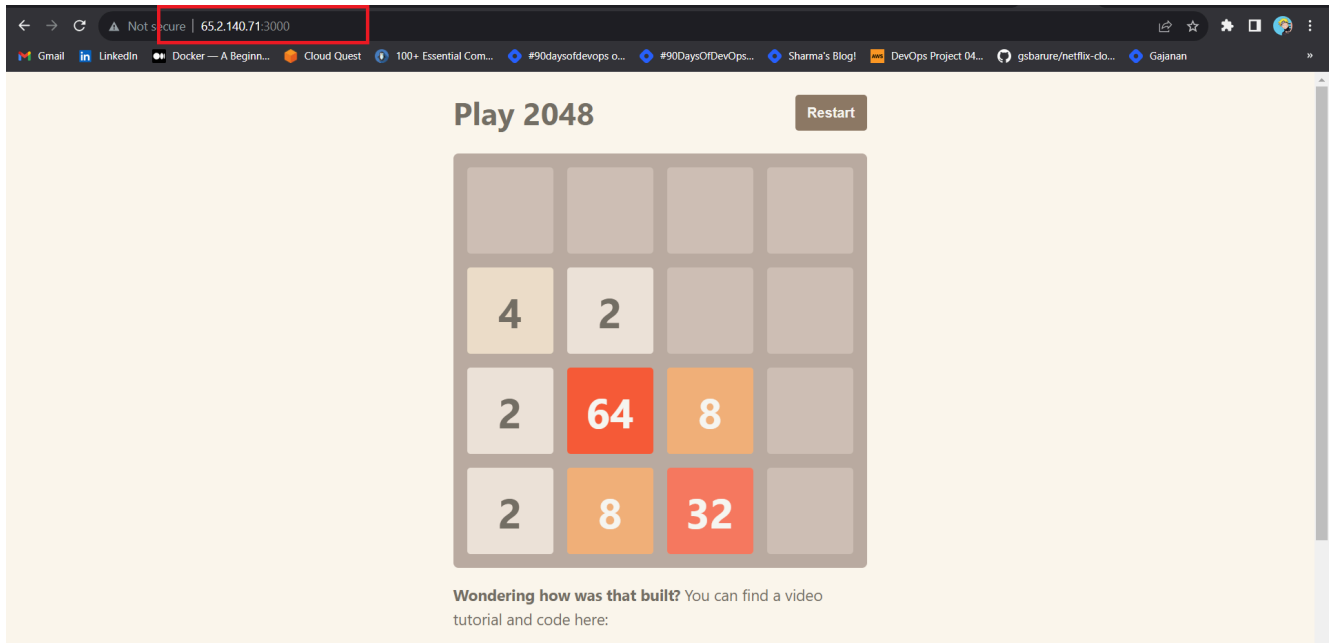
172.31.39.135

Public IPv4 DNS
ec2-65-2-140-71.ap-south-1.compute.amazonaws.com
[open address](#)

copy the Public IP of the provisioned instance

```
<public-ip:3000>
```

Play Game 2048



Delete the instances.

Continuous improvement is not just about doing the same thing better. It's about reimagining what's possible and striving for excellence in every line of code, every pipeline, and every deployment. In the world of Jenkins and Ansible, automation is the key to unlocking innovation. So, keep coding, keep automating, and keep pushing the boundaries of what's achievable. Your journey has just begun.

Thanks for Reading my Blog.



Ajay Kumar Yegireddi is a DevSecOps Engineer and System Administrator, with a passion for sharing real-world DevSecOps projects and tasks. **Mr. Cloud Book**, provides hands-on tutorials and practical insights to help others master DevSecOps tools and workflows. Content is designed to bridge the gap between development, security, and operations, making complex concepts easy to understand for both beginners and professionals.

Comments

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

☐ Save my name, email, and website in this browser for the next time I comment.

I'm not a robot

reCAPTCHA
Privacy - Terms

Post Comment

Uncategorized

Day -1: Kick Off Cloud Security with AWS Registration

18 August 2025

Uncategorized

How to Automate Incident Response : How Q Developer Helped Me Automate a Daily Pain Point

AI

How to Run Docker Model Runner on Ubuntu 24.04

11 July 2025

22 July 2025

Upskill with Ajay: DevSecOps Mastery

Join Mr Cloud book to master DevSecOps through real-world projects. Learn CI/CD, security integration, automation, and more, gaining hands-on skills for industry-level challenges.



Important Links

[Privacy Policy](#)[Terms & Conditions](#)[Contact](#)

Resources

[Blog](#)[YouTube Channel](#)

© 2024 • Powered by [Mr Cloud Book](#)

[Follow Us on YouTube](#)