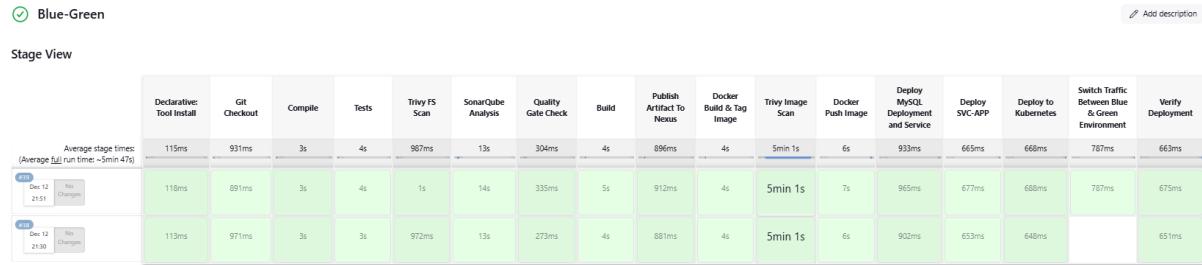


# K8s CI/CD Pipeline for Blue-Green Deployments with Automated Traffic Switching and Security Checks



## Flow Structure:

1. Start → Git Checkout
2. Git Checkout → Compile
3. Compile → Tests
4. Tests → Trivy FS Scan
5. Trivy FS Scan → SonarQube Analysis
6. SonarQube Analysis → Quality Gate Check
7. Quality Gate Check → Build
8. Build → Publish Artifact to Nexus
9. Publish Artifact to Nexus → Docker Build & Tag Image
10. Docker Build & Tag Image → Docker Push Image
11. Docker Push Image → Deploy MySQL Deployment and Service
12. Deploy MySQL Deployment and Service → Deploy SVC-APP
13. Deploy SVC-APP → Deploy to Kubernetes (Blue-Green)
14. Deploy to Kubernetes (Blue-Green) → Switch Traffic Between Blue & Green Environments  
(optional based on SWITCH\_TRAFFIC parameter)
15. Switch Traffic Between Blue & Green Environments → Verify Deployment
16. Verify Deployment → End

## Firstly check & open the Ports in Security Groups

The screenshot shows the AWS Management Console with the URL [ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#SecurityGroup:group-id=sg-0d65da1ffbf540fa6](https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#SecurityGroup:group-id=sg-0d65da1ffbf540fa6). The left sidebar is collapsed. The main content area displays the 'Inbound rules (14)' table. The columns are: Name, Security group rule ID, IP version, Type, Protocol, and Port range. The rules include various ports ranging from 22 to 32767, primarily using TCP and Custom TCP protocols.

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0655ec8a762cab6d8	IPv4	Custom TCP	TCP	30000 - 32767
-	sgr-0e0d13fddb86be721	IPv4	Custom TCP	TCP	6443
-	sgr-0f31871b093e034dc	IPv4	SMTPS	TCP	465
-	sgr-01bbcd1a698d37eca	IPv4	SSH	TCP	22
-	sgr-0ea23ed14d046f7b	IPv4	Custom TCP	TCP	500 - 1000
-	sgr-08425049e7e1837d	IPv4	Custom TCP	TCP	2000 - 10000
-	sgr-04f2be55473190be9	IPv4	Custom TCP	TCP	4500 - 5500
-	sgr-0a67f6f5d5de848a0c	IPv4	HTTPS	TCP	443
-	sgr-07845bc84e5598d7e	IPv4	Custom TCP	TCP	27017
-	sgr-07e20756df67787	IPv4	Custom TCP	TCP	3000 - 10000
-	sgr-08b86f845b0f7143e	IPv4	Custom TCP	TCP	1000 - 1100
-	sgr-03c50928e1dc2f102	IPv4	Custom TCP	TCP	30000 - 40000
-	sgr-069036767bb048799	IPv4	HTTP	TCP	80
-	sgr-0a50ea1f100af9366	IPv4	SMTP	TCP	25

Create a instance with ubuntu with type t2. Medium storage 20gb.

The screenshot shows the AWS Management Console with the URL [ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances](https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances). The left sidebar is collapsed. The main content area shows the 'Launch an instance' wizard. Step 1: Name and tags (Name: SERVER). Step 2: Application and OS Images (Amazon Machine Image) (Selected: Ubuntu Server 24.04 LTS (HVM) SSD Volume Type). Step 3: Summary (Number of instances: 1, Virtual server type: t2.medium, Firewall: maven, Storage: 1 volume(s) - 20 GiB). A callout box highlights the Free tier information: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month.' The 'Launch instance' button is visible at the bottom right.

The screenshot shows the AWS Management Console with the EC2 Instances page. A single instance named "SERVER" (i-00f8b37dd9e375648) is listed as "Running". The instance details page is open, showing the following information:

- Public IPv4 address:** 13.232.187.198
- Private IP4 DNS:** ec2-13-232-187-198.ap-south-1.compute.amazonaws.com
- Instance state:** Running
- Instance type:** t2.medium
- IPV6 address:** -
- Hostname type:** IP name: ip-172-31-13-35.ap-south-1.compute.internal
- Answer private resource DNS name:** IPv4 (A)
- Public IP4 DNS:** ec2-13-232-187-198.ap-south-1.compute.amazonaws.com
- Private IP4 addresses:** 172.31.13.35
- Elastic IP addresses:** -

## Connect to instance using mobaxterm

The screenshot shows a MobaXterm window titled "13.232.187.198 (ubuntu)". The terminal session is authenticated using a public key ("Imported-Openssh-Key"). The terminal displays the following text:

```

MobaXterm Personal Edition v24.2 •
(SSH client, X server and network tools)

SSH session to ubuntu@13.232.187.198
  • Direct SSH : ✓
  • SSH compression : ✓
  • SSH-browser : ✓
  • X11-forwarding : ✓ (remote display is forwarded through SSH)

For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Thu Dec 12 07:18:53 UTC 2024

System load: 0.01 Processes: 123
Usage of /: 9.0% of 18.33GB Users logged in: 0
Memory usage: 6% IPv4 address for enX0: 172.31.13.35
Swap usage: 0% 

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

```

The status bar at the bottom shows network information: ip-172-31-13-35, 0%, 0.44 GB / 3.82 GB, 0.01 Mb/s, 0.00 Mb/s, 181 sec, ubuntu, /: 10%, /boot: 10%, /boot/efi: 6%.

## Use the below commands

sudo apt update

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

sudo apt install unzip

unzip awscliv2.zip

sudo ./aws/install

```

ubuntu@ip-172-31-13-35:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
ubuntu@ip-172-31-13-35:~$ unzip awscliv2.zip
ubuntu@ip-172-31-13-35:~$ sudo apt install unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 53 not upgraded.
Need to get 174 kB of archives.
After this operation, 384 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 unzip amd64 6.0-28ubuntu4.1 [174 kB]
Fetched 174 kB in 0s (10.1 MB/s)
Selecting previously unselected package unzip.
(Reading database ... 70601 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-28ubuntu4.1_amd64.deb ...
Unpacking unzip (6.0-28ubuntu4.1) ...
Setting up unzip (6.0-28ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.

```

**UNREGISTERED VERSION** - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

```

ubuntu@ip-172-31-13-35:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
ubuntu@ip-172-31-13-35:~$ 

```

**UNREGISTERED VERSION** - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

aws configure

now go no aws

security credentials → Access keys → create key → download

Screenshot of the AWS Console Home page for the ap-south-1 region. The top navigation bar shows 'Console Home | Instances | EC2 | ap-south-1' and the URL 'ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1#'. The sidebar on the left lists recently visited services: EC2, Billing and Cost Management, Support, Database Migration Service, RDS, Elastic Container Registry, Elastic Container Service, and IAM. The main content area displays sections for Applications (0), AWS Health (Info), and Cost and usage (Info). A context menu is open over the Applications section, showing options like 'Reset to default layout', 'Create application', and 'Sign out'. The bottom of the screen shows standard browser controls and a footer with links to 'Privacy', 'Terms', and 'Cookie preferences'.

Screenshot of the AWS Security credentials page for the ap-south-1 region. The top navigation bar shows 'Console Home | Security credentials | IAM | Global' and the URL 'us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/security\_credentials'. The sidebar on the left includes sections for Identity and Access Management (IAM), Access management, Access reports, and CloudShell. The main content area shows three sections: 'Access keys (0)', 'CloudFront key pairs (0)', and 'X.509 Signing certificates (0)'. Each section has a 'Create access key' button. The bottom of the screen shows standard browser controls and a footer with links to 'Privacy', 'Terms', and 'Cookie preferences'.

Screenshot of the AWS Create access key wizard for the ap-south-1 region. The top navigation bar shows 'Console Home | Create access key | IAM | Global' and the URL 'us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/security\_credentials/access-key-wizard'. The sidebar on the left shows 'IAM > Security credentials > Create access key'. The main content area displays a green success message: 'Access key created. This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below this, a 'Retrieve access key' step is shown with a sub-step 'Alternatives to root user access keys' and 'Step 2 Retrieve access key'. A table shows the 'Access key' (AKIA4ZL74RSRDVQGLYWD) and 'Secret access key' (XXXXXXXXXXXXXX). A 'Show' link is available for the secret key. A section titled 'Access key best practices' lists several guidelines. At the bottom, there are 'Download .csv file' and 'Done' buttons. The bottom of the screen shows standard browser controls and a footer with links to 'Privacy', 'Terms', and 'Cookie preferences'.

Now go to mobaxterm and paste the credentials

```
ubuntu@ip-172-31-13-35:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
ubuntu@ip-172-31-13-35:~$ aws configure
AWS Access Key ID [None]: AKIA4ZL74R5RDVQGLYWD
AWS Secret Access Key [None]: u0HMs05A7RUob9x2ijTSSHJ60P8DlvIaEHiuT+
Default region name [None]: ap-south-1
Default output format [None]:
ubuntu@ip-172-31-13-35:~$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Now I am connected to aws

Now I need to install terraform

sudo snap install terraform --classic

```
ubuntu@ip-172-31-13-35:~$ sudo snap install terraform --classic
Download snap "core24" (609) from channel "stable"
15% 273kB/s 3m37s
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Now clone the git hub repo

<https://github.com/gadhasusaikrishna/blue-green-e.k.s.git>

```

ubuntu@ip-172-31-13-35:~$ sudo snap install terraform --classic
Download snap "core24" (609) from channel "stable"
Download snap "core24" (609) from channel "stable"
terraform 1.10.0 from Snapcrafters installed
ubuntu@ip-172-31-13-35:~$ git clone https://github.com/gadhasusaikrishna/blue-green-e.k.s.git
Cloning into 'blue-green-e.k.s'...
remote: Enumerating objects: 56, done.
remote: Counting objects: 100% (56/56), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 56 (delta 51), reused 56 (delta 5), pack-reused 0 (from 0)
Resolving deltas: 100% (56/56), 23.02 KiB | 5.75 MiB/s, done.
Resolving deltas: 100% (5/5), done.
ubuntu@ip-172-31-13-35:~$ 

```

cd blue-green-e.k.s

cd Cluster

```

ubuntu@ip-172-31-13-35:~$ cd Blue-Green-Deployment/
-bash: cd: Blue-Green-Deployment/: No such file or directory
ubuntu@ip-172-31-13-35:~$ ls
awscli2.zip  blue-green-e.k.s
ubuntu@ip-172-31-13-35:~$ cd blue-green-e.k.s
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s$ ls
Cluster Jenkinsfile app-deployment-blue.yml bankapp-service.yml mvnw.cmd pom.xml
Dockefile Setup-RBAC.md app-deployment-green.yml mvnw mysql-ds.yml src
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s$ 

```

terraform init

```

ubuntu@ip-172-31-13-35:~$ ls
aws awscli2.zip blue-green-e.k.s
ubuntu@ip-172-31-13-35:~$ cd blue-green-e.k.s
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s$ ls
Cluster Jenkinsfile app-deployment-blue.yml bankapp-service.yml mvnw.cmd pom.xml
Dockefile Setup-RBAC.md app-deployment-green.yml mvnw mysql-ds.yml src
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s$ cd Cluster
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ ls
eks-rbac.md main.tf output.tf variables.tf
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.80.0...
- Installed hashicorp/aws v5.80.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

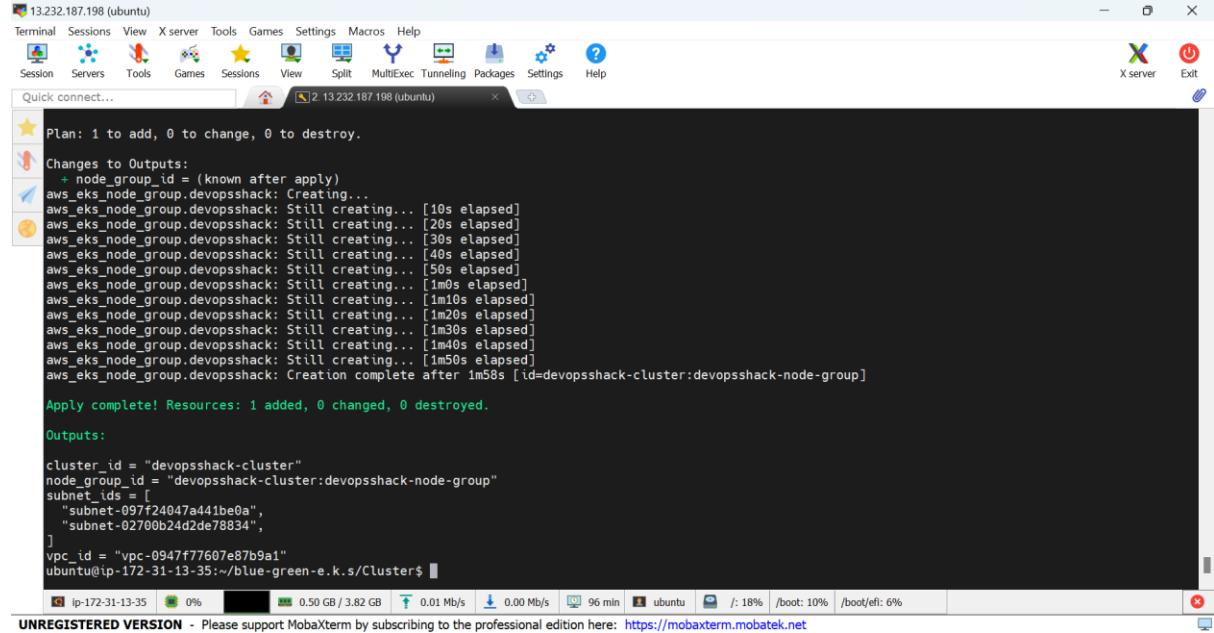
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ 

```

## terraform plan

terraform apply --auto-approve



```
13.232.187.198 (ubuntu)
Plan: 1 to add, 0 to change, 0 to destroy.

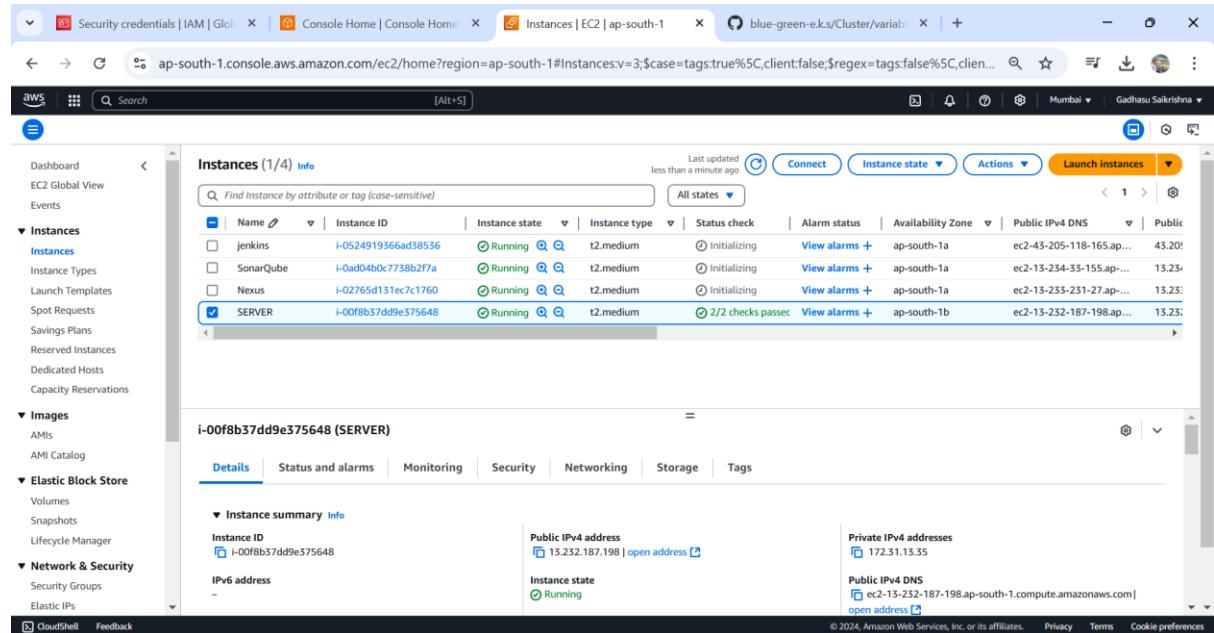
Changes to Outputs:
+ node_group_id = (known after apply)
aws_eks_node_group.devopsshack: Creating...
aws_eks_node_group.devopsshack: Still creating... [10s elapsed]
aws_eks_node_group.devopsshack: Still creating... [20s elapsed]
aws_eks_node_group.devopsshack: Still creating... [30s elapsed]
aws_eks_node_group.devopsshack: Still creating... [40s elapsed]
aws_eks_node_group.devopsshack: Still creating... [50s elapsed]
aws_eks_node_group.devopsshack: Still creating... [1m0s elapsed]
aws_eks_node_group.devopsshack: Still creating... [1m10s elapsed]
aws_eks_node_group.devopsshack: Still creating... [1m20s elapsed]
aws_eks_node_group.devopsshack: Still creating... [1m30s elapsed]
aws_eks_node_group.devopsshack: Still creating... [1m40s elapsed]
aws_eks_node_group.devopsshack: Still creating... [1m50s elapsed]
aws_eks_node_group.devopsshack: Creation complete after 1m50s [id=devopsshack-cluster:devopsshack-node-group]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
cluster_id = "devopsshack-cluster"
node_group_id = "devopsshack-cluster:devopsshack-node-group"
subnet_ids = [
  "subnet-097f24047a441be0a",
  "subnet-02700b24d2de78834",
]
vpc_id = "vpc-0947f77607e87b9a1"
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Now go to aws and launch 3 instances with ubuntu and type t2.medium with 25gb storage



Instances (1/4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
jenkins	i-0524919366ad38536	Running	t2.medium	Initializing	View alarms +	ap-south-1a	ec2-43-205-118-165.ap...	43.20%
SonarQube	i-0ad04b0c7738b2f7a	Running	t2.medium	Initializing	View alarms +	ap-south-1a	ec2-13-234-33-155.ap...	13.23%
Nexus	i-02765d0131ec7c1760	Running	t2.medium	Initializing	View alarms +	ap-south-1a	ec2-13-233-231-27.ap...	13.23%
<b>SERVER</b>	<b>i-00f8b37dd9e375648</b>	<b>Running</b>	<b>t2.medium</b>	<b>2/2 checks passed</b>	<b>View alarm +</b>	<b>ap-south-1b</b>	<b>ec2-13-232-187-198.ap...</b>	<b>13.23%</b>

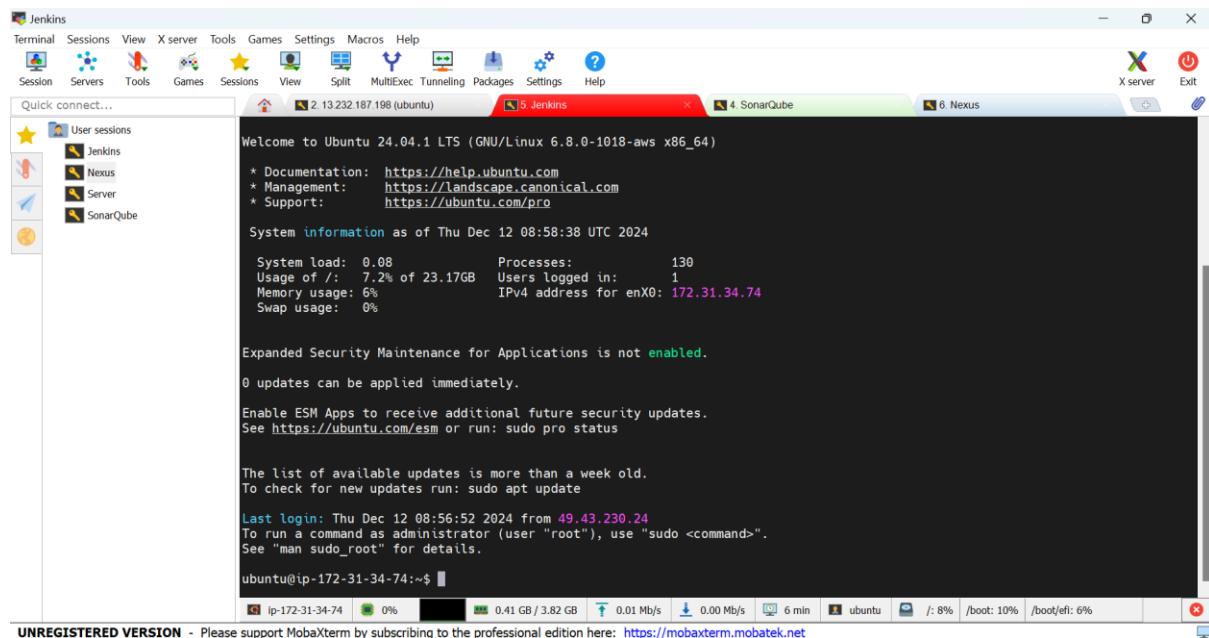
i-00f8b37dd9e375648 (SERVER)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary

Instance ID: i-00f8b37dd9e375648	Public IPv4 address: 13.232.187.198   open address	Private IPv4 addresses: 172.31.13.55
IPv6 address: -	Instance state: Running	Public IPv4 DNS: ec2-13-232-187-198.ap-south-1.compute.amazonaws.com   open address

## Connect to Jenkins,SonarQube&Nexus using mobaxterm



Use the below commands in Jenkins

```
sudo apt update
```

```
sudo apt install openjdk-17-jre-headless -y
```

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
```

```
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

```
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
```

```
sudo apt-get install Jenkins
```

now go to Nexus & Sonarqube and use below commands

```
sudo apt update
```

```
sudo apt install docker . io -y
```

```
sudo usermod -aG docker ubuntu
```

```
newgrp docker
```

```
docker run -d -p 8081:8081 sonatype/nexus3 (use command for only NEXUS)
```

```
docker run -d -p 9000:9000 sonarqube:its-community ( use command for only SonarQube)
```

## SonarQube

The screenshot shows a MobaXterm window titled "SonarQube". The terminal session is connected to an Ubuntu host at IP 172.31.43.141. The user is running a Docker container for SonarQube on port 9000. The terminal output shows the Docker command being run and the resulting logs, including the download of the SonarQube image and the creation of a new Docker container named "sonarqube:lts-community". The container is up and listening on port 9000. The status bar at the bottom shows the connection details and system metrics.

```
ubuntu@ip-172-31-43-141:~$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-43-141:~$ newgrp docker
ubuntu@ip-172-31-43-141:~$ docker run -d -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
6414378bb6477: Pull complete
17da8ec43a12: Pull complete
d12988e90d61: Pull complete
f4d133ca2b7f: Pull complete
143733ae87a4: Pull complete
8438621478bb: Pull complete
3d0284140b24: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:c337c07849de45a727f09db875779ad7b5784e0b02b096c1f8cd72e27a9fdc
Status: Downloaded newer image for sonarqube:lts-community
74b6df71b12f7bc96607e9d4b071e5f4713e32e2e7acd9d7e0cd28f7b400e4b
ubuntu@ip-172-31-43-141:~$ docker ps -a
CONTAINER ID IMAGE NAMES COMMAND CREATED STATUS PORTS
74b6df71b12f sonarqube:lts-community "/opt/sonarqube/dock..." 10 seconds ago Up 4 seconds 0.0.0.0:9000->9000/tcp, :::9
ubuntu@ip-172-31-43-141:~$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

## Nexus

The screenshot shows a MobaXterm window titled "Nexus". The terminal session is connected to an Ubuntu host at IP 172.31.44.47. The user is running a Docker container for Sonatype Nexus on port 8081. The terminal output shows the Docker command being run and the resulting logs, including the download of the Sonatype Nexus image and the creation of a new Docker container named "sonatype/nexus3". The container is up and listening on port 8081. The status bar at the bottom shows the connection details and system metrics.

```
ubuntu@ip-172-31-44-47:~$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-44-47:~$ newgrp docker
ubuntu@ip-172-31-44-47:~$ docker run -d -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
a09bb1026942: Pull complete
f773e5399e27: Pull complete
3cbdef5ie75d: Pull complete
c2a1dd00ce83: Pull complete
04adf4cf78d5: Pull complete
d878677f59c6: Pull complete
77476ce54871: Pull complete
Digest: sha256:aac80ef14735e30cc473c115787903fc0bfa44b738b6e6f8d921f35ff968606f
Status: Downloaded newer image for sonatype/nexus3:latest
5a4bee65ccfc6a5a125a80470a9f5034647e579f7db3cffc6634d18bab9c87a06
ubuntu@ip-172-31-44-47:~$ docker ps -a
CONTAINER ID IMAGE NAMES COMMAND CREATED STATUS PORTS
5a4bee65ccfc sonatype/nexus3 "/opt/sonatype/nexus..." 10 seconds ago Up 4 seconds 0.0.0.0:8081->8081/tcp, :::8081->808
ubuntu@ip-172-31-44-47:~$
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Both SonarQube & Nexus are Working.

The screenshot shows a web browser window for the Sonatype Nexus Repository. The URL is 13.127.231.177:9000/projects/main. The page displays the "Welcome" screen with usage statistics for components, unique logins, and requests per day. It also shows options for importing projects from various platforms like Azure DevOps, Bitbucket, GitHub, and GitLab. A message at the bottom indicates that anonymous usage metrics will be collected.

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform. First, you need to set up a DevOps platform configuration.

From Azure DevOps From Bitbucket Server From Bitbucket Cloud From GitHub From GitLab

Are you just testing or have an advanced use-case? Create a project manually.

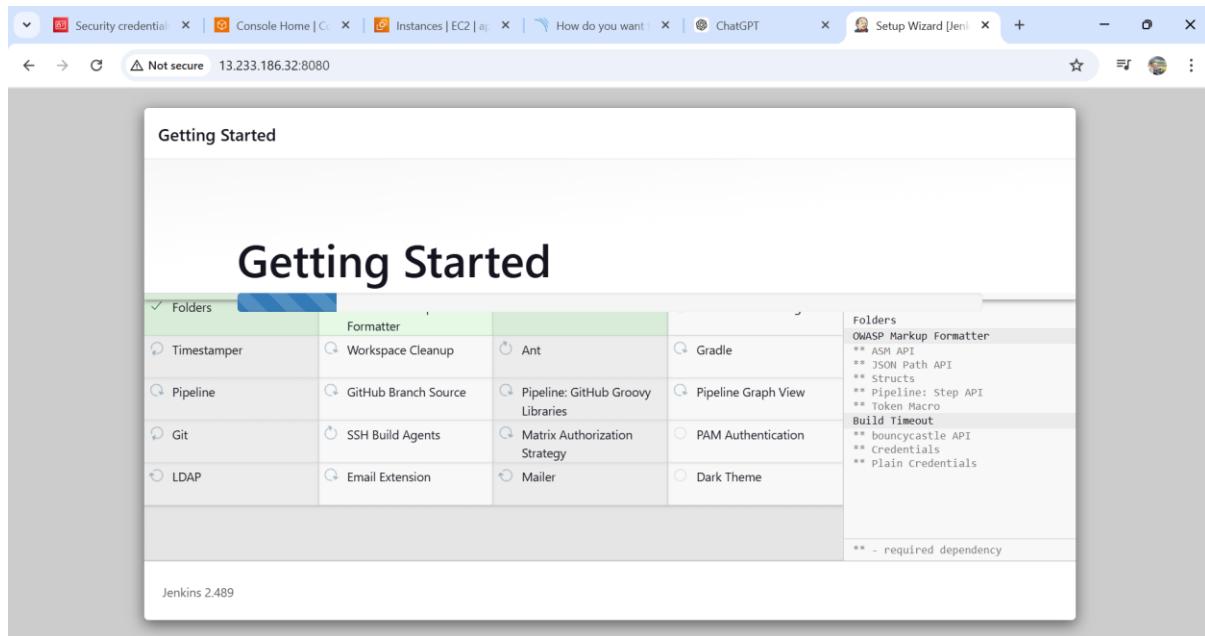
Sonatype will start to collect anonymous, non-sensitive usage metrics and performance information to shape the future of Nexus Repository. Learn more about the information we collect or decline.

Now go to Jenkins and follow below commands

```
sudo systemctl enable Jenkins
```

```
sudo systemctl start Jenkins
```

```
sudo systemctl status Jenkins
```



Use this on Jenkins

```
# Add Docker's official GPG key:
```

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
# Add the repository to Apt sources:
```

```
echo \
```

```
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
```

```
$(./etc/os-release && echo "$VERSION_CODENAME") stable" | \
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

```
sudo usermod -ag docker Jenkins
```

now restart the Jenkins

The screenshot shows the Jenkins dashboard at [13.233.186.32:8080](http://13.233.186.32:8080). The top navigation bar includes tabs for Security credentials, Console Home, Instances | EC2, How do you want..., ChatGPT, and Dashboard Jenkins. The dashboard features a sidebar with links for New Item, Build History, Manage Jenkins, and My Views. The main area has sections for the Build Queue (No builds in the queue), Create a job, Set up a distributed build (with sub-links for Set up an agent and Configure a cloud), and Start building your software project. A "Welcome to Jenkins!" message is centered.

Now install Trivy on Jenkins

```
sudo apt-get install wget apt-transport-https gnupg lsb-release
```

```
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null
```

```
echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list
```

```
sudo apt-get update
```

```
sudo apt-get install trivy
```

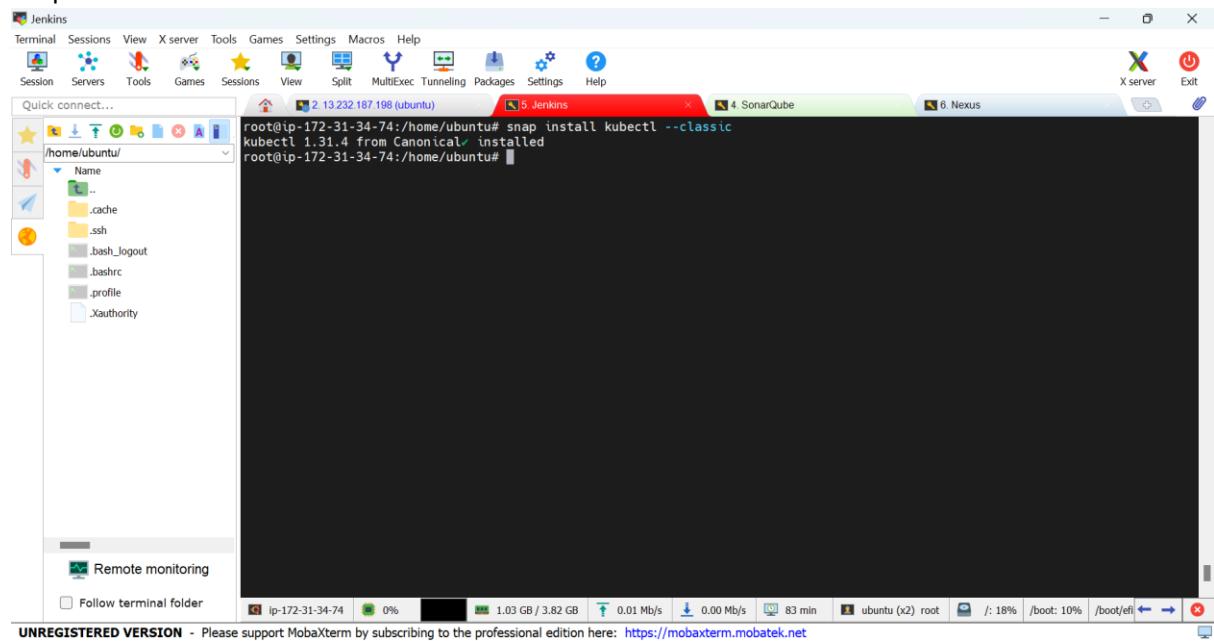
The screenshot shows a MobaXterm session titled 'Jenkins' connected to a root shell on an Ubuntu host. The terminal window displays the command history for installing Trivy:

```
root@ip-172-31-34-74:/home/ubuntu# sudo usermod -ag docker jenkins
root@ip-172-31-34-74:/home/ubuntu# sudo apt-get install wget apt-transport-https gnupg lsb-release
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list
sudo apt-get update
sudo apt-get install trivy
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
gnupg is already the newest version (2.4.4-2ubuntu17).
gnupg set to manually installed.
lsb-release is already the newest version (22.0-2).
lsb-release set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 0 newly installed, 0 to remove and 50 not upgraded.
Need to get 3974 B in 0s.
After this operation, 35.8 kB 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 noble/universe amd64 apt-transport-https all 2.7.14build2 [3974 B]
Fetched 3974 B in 0s (228 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 85625 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
Setting up apt-transport-https (2.7.14build2) ...
Scanning processes...
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

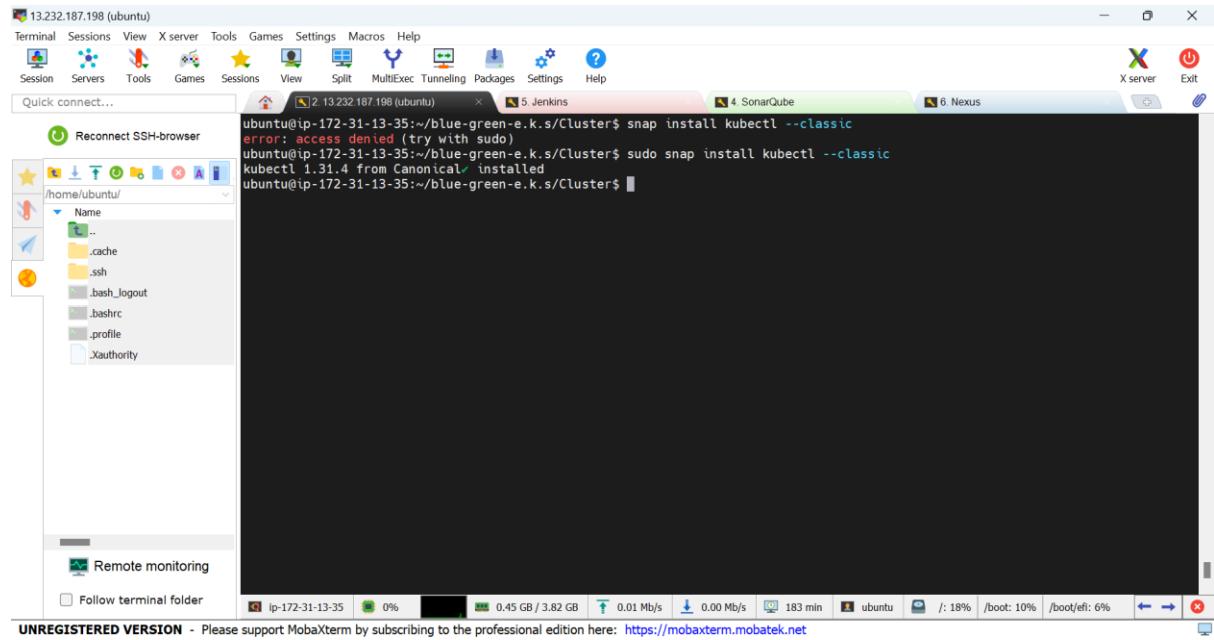
## Install kubectl on Jenkins

`snap install kubectl --classic`

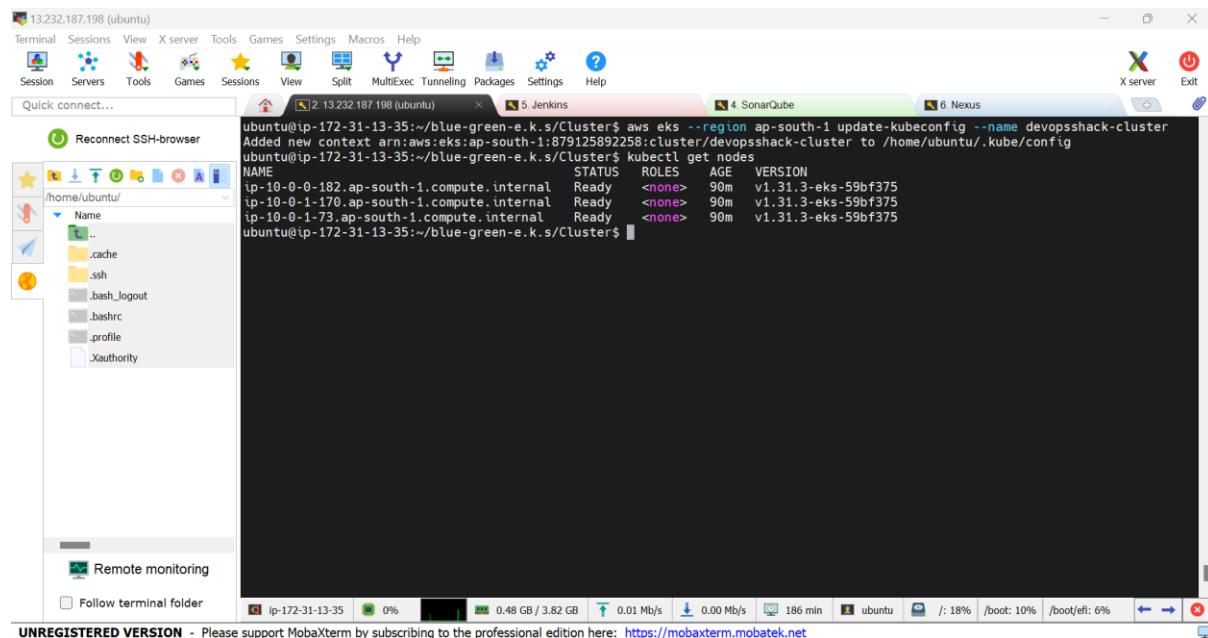


now go to server instance and perform below commands

`snap install kubectl --classic`



```
aws eks --region ap-south-1 update-kubeconfig --name devopsshack-cluster
```



A screenshot of the MobaXterm interface. The terminal window shows the command `aws eks --region ap-south-1 update-kubeconfig --name devopsshack-cluster` being run on an Ubuntu host (IP 13.232.187.198). The output indicates that a new context was added for the specified cluster. The terminal also displays the result of the `kubectl get nodes` command, listing three nodes: ip-10-0-0-182.ap-south-1.compute.internal, ip-10-0-1-170.ap-south-1.compute.internal, and ip-10-0-1-73.ap-south-1.compute.internal, all in a Ready state. The file browser sidebar shows the current directory is /home/ubuntu.

Vi sa.yml

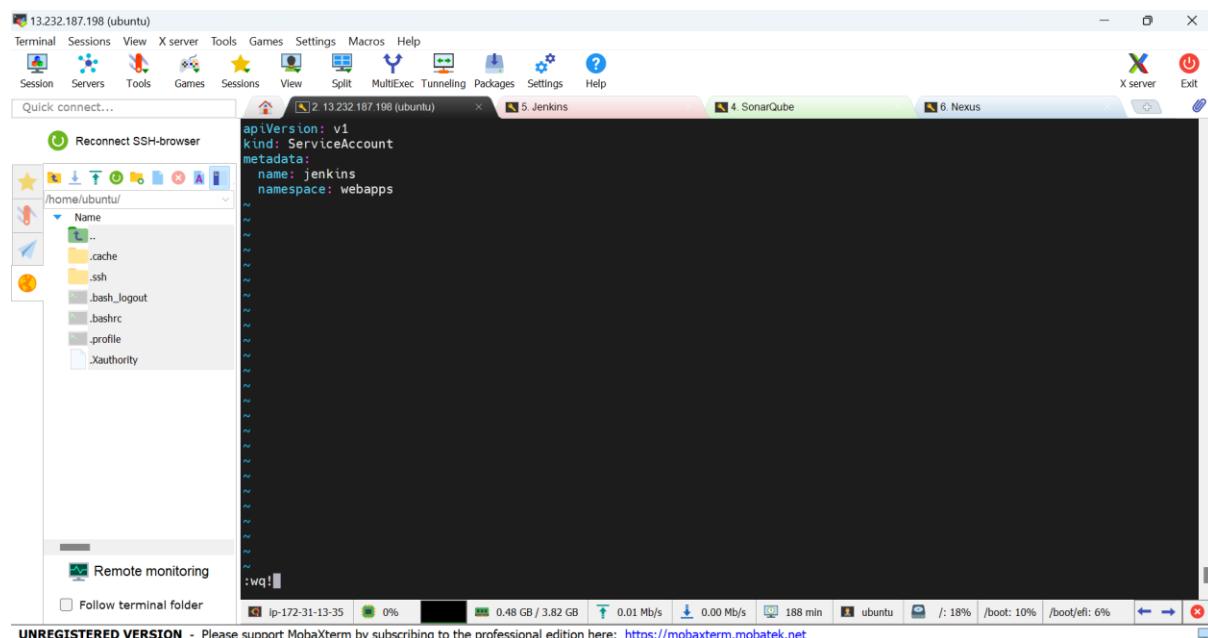
```
apiVersion: v1
```

```
kind: ServiceAccount
```

```
metadata:
```

```
  name: jenkins
```

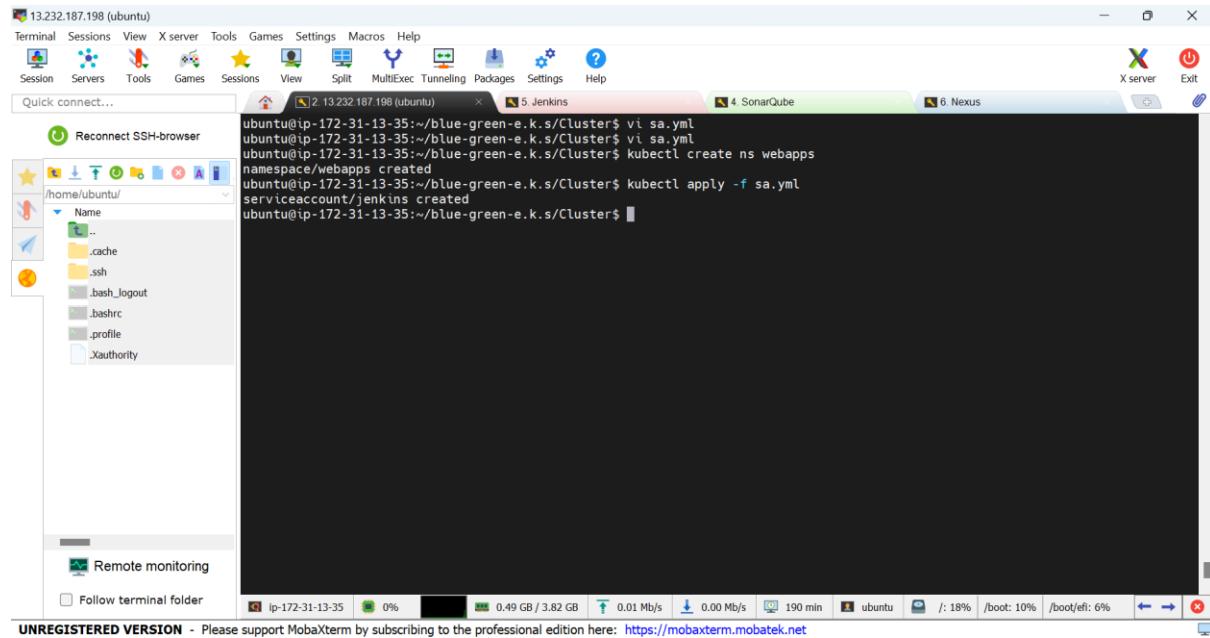
```
  namespace: webapps
```



A screenshot of the MobaXterm interface. The terminal window shows the creation of a ServiceAccount named "jenkins" in the "webapps" namespace. The command used was `apiVersion: v1 kind: ServiceAccount metadata: name: jenkins namespace: webapps`. The file browser sidebar shows the current directory is /home/ubuntu.

```
kubectl create ns webapps
```

```
kubectl apply -f sa.yaml
```

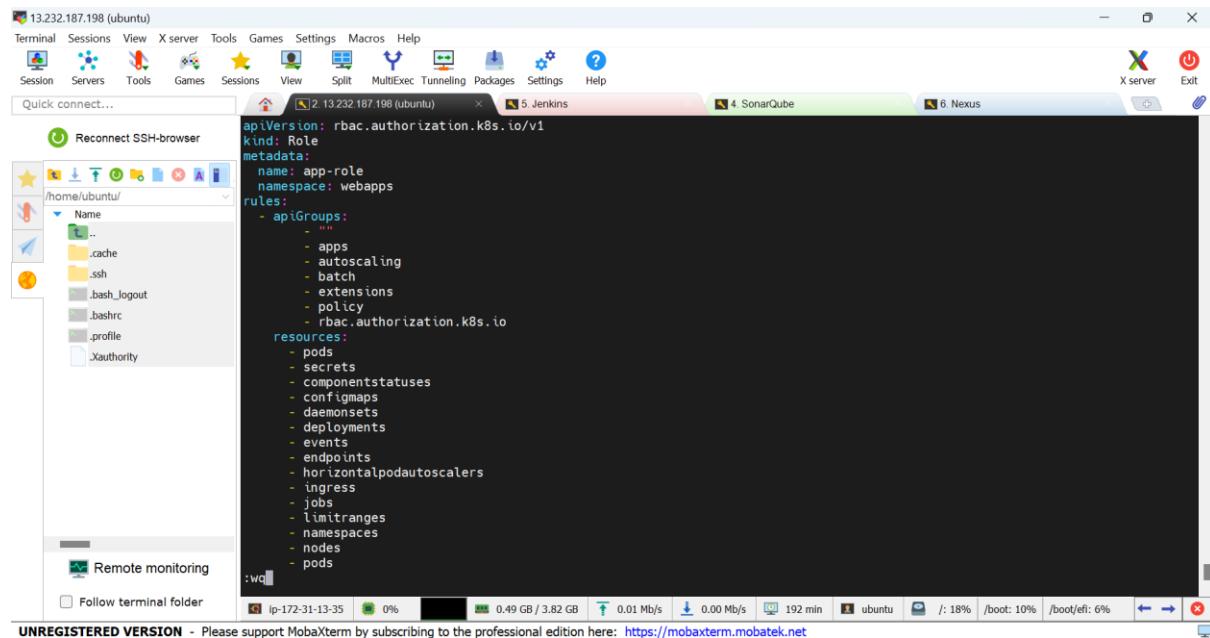


A screenshot of the MobaXterm interface. The title bar shows "13.232.187.198 (ubuntu)". The terminal window (session 2) displays the following command sequence:

```
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ vi sa.yaml
namespace/webapps created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl create ns webapps
serviceaccount/jenkins created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$
```

The left sidebar shows a file tree under "/home/ubuntu/" with files like ".cache", ".ssh", ".bash\_logout", ".bashrc", ".profile", and ".xauthority". The bottom status bar shows network information: IP: ip-172-31-13-35, CPU: 0%, RAM: 0.49 GB / 3.82 GB, Network: 0.01 Mb/s, Disk: 190 min, and battery: 18%.

```
Vi role.yaml
```



A screenshot of the MobaXterm interface. The title bar shows "13.232.187.198 (ubuntu)". The terminal window (session 2) displays the following command:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: app-role
  namespace: webapps
rules:
  - apiGroups:
      - ""
    resources:
      - pods
      - secrets
      - componentstatuses
      - configmaps
      - daemonsets
      - deployments
      - events
      - endpoints
      - horizontalpodautoscalers
      - ingress
      - jobs
      - limitranges
      - namespaces
      - nodes
      - pods
:wq
```

The left sidebar shows a file tree under "/home/ubuntu/" with files like ".cache", ".ssh", ".bash\_logout", ".bashrc", ".profile", and ".xauthority". The bottom status bar shows network information: IP: ip-172-31-13-35, CPU: 0%, RAM: 0.49 GB / 3.82 GB, Network: 0.01 Mb/s, Disk: 192 min, and battery: 18%.

kubectl apply -f role.yml

```
13.232.187.198 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunnelling Packages Settings Help
Quick connect...
Reconnect SSH-browser
/home/ubuntu/
Name
.. .cache .ssh .bash_logout .bashrc .profile .xauthority
Remote monitoring
Follow terminal folder
ip-172-31-13-35 0% 0.50 GB / 3.82 GB 0.01 Mb/s 0.00 Mb/s 195 min ubuntu /: 18% /boot: 10% /boot/efi: 6%
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

```
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ vi sa.yaml
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ vi sa.yaml
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl create ns webapps
namespace/webapps created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl apply -f sa.yaml
serviceaccount/jenkins created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl apply -d role.yaml
error: unknown shorthand flag: 'd' in '-d'
See 'kubectl apply --help' for usage.
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl apply -d role.yaml
error: unknown shorthand flag: 'd' in '-d'
See 'kubectl apply --help' for usage.
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl apply -f role.yaml
role.rbac.authorization.k8s.io/app-role created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$
```

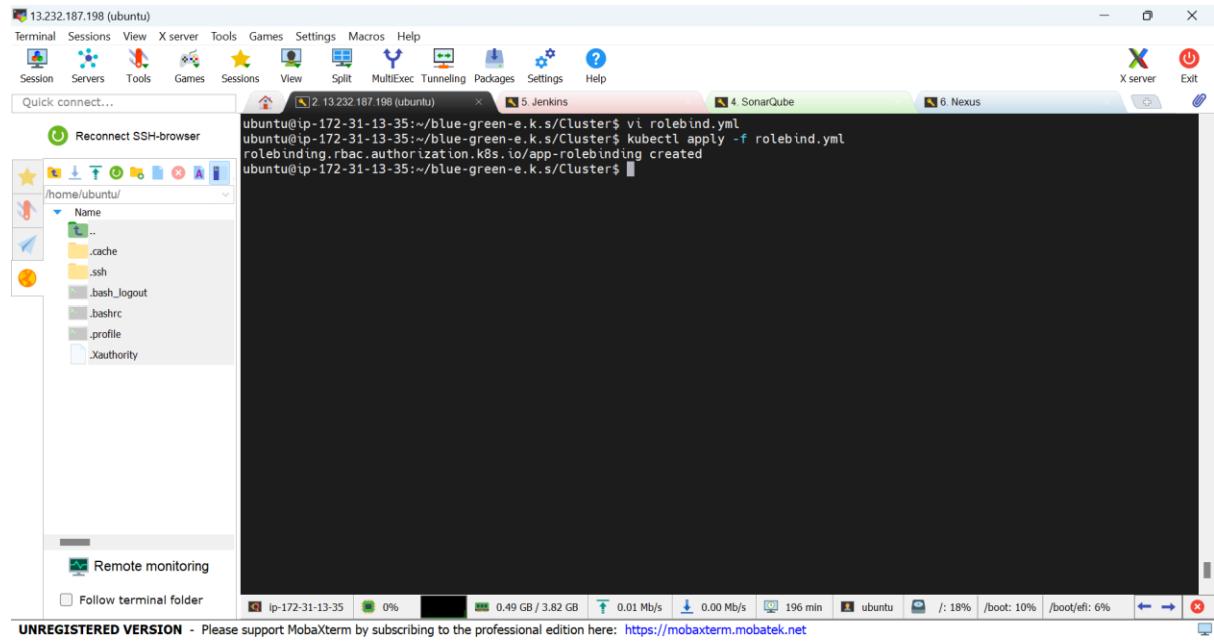
vi rolebind.yml

```
13.232.187.198 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunnelling Packages Settings Help
Quick connect...
Reconnect SSH-browser
/home/ubuntu/
Name
.. .cache .ssh .bash_logout .bashrc .profile .xauthority
Remote monitoring
Follow terminal folder
ip-172-31-13-35 0% 0.49 GB / 3.82 GB 0.01 Mb/s 0.00 Mb/s 196 min ubuntu /: 18% /boot: 10% /boot/efi: 6%
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: app-rolebinding
  namespace: webapps
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: app-role
subjects:
- namespace: webapps
  kind: ServiceAccount
  name: jenkins
```

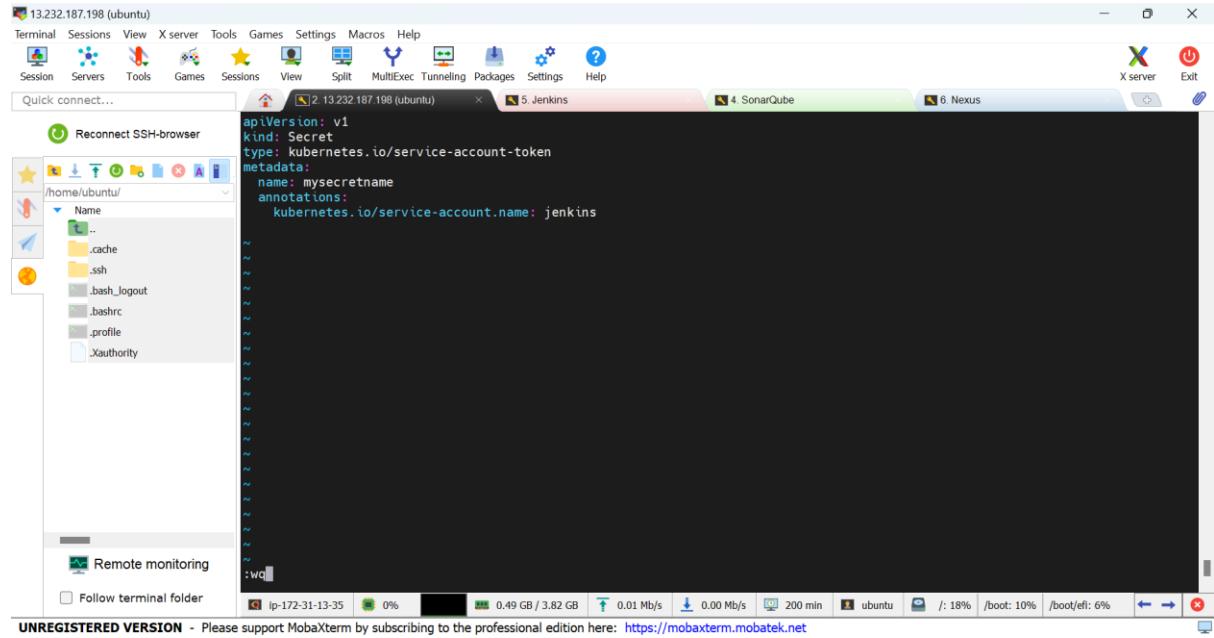
```
:wq
```

kubectl apply -f role.yml



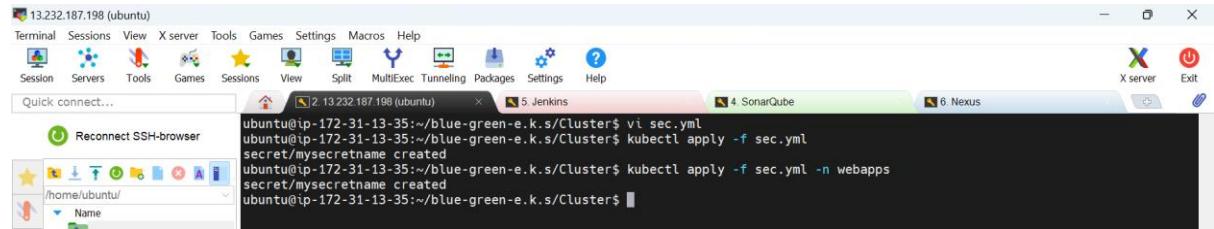
```
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ vi rolebind.yml
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl apply -f rolebind.yml
rolebinding.rbac.authorization.k8s.io/app-rolebinding created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$
```

Vi sec.yml



```
apiVersion: v1
kind: Secret
type: kubernetes.io/service-account-token
metadata:
  name: mysecretname
  annotations:
    kubernetes.io/service-account.name: jenkins
```

kubectl apply -f sec.yml -n webapps



```
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ vi sec.yml
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl apply -f sec.yml
secret/mysecretname created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl apply -f sec.yml -n webapps
secret/mysecretname created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$
```

kubectl describe secret mysecretname -n webapps

```
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ vi sec.yml
secret/mysecretname created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl apply -f sec.yml
secret/mysecretname created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl apply -f sec.yml -n webapps
secret/mysecretname created
ubuntu@ip-172-31-13-35:~/blue-green-e.k.s/Cluster$ kubectl describe secret mysecretname -n webapps
Name:           mysecretname
Namespace:      webapps
Labels:         <none>
Annotations:   kubernetes.io/service-account.name: jenkins
               kubernetes.io/service-account.uid: ff399838-4eb5-4bee-8f7f-527715523c1c
Type:          kubernetes.io/service-account-token
Data
====
ca.crt:        1107 bytes
token:          7 bytes
Annotations:   kubernetes.io/service-account.name: jenkins
               kubernetes.io/service-account.uid: ff399838-4eb5-4bee-8f7f-527715523c1c
Events:        <none>
```

Now go to Jenkins

Manage Jenkins → credentials → system → global →

Secret text → <paste TOKEN>

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Kind: Secret text

Scope: Global (Jenkins, nodes, items, all child items, etc.)

Secret:

ID: k8-token

Description: k8-token

Create

## Manage Jenkins → Plugins → install

Install	Name	Released
<input checked="" type="checkbox"/>	SonarQube Scanner 2.17.3	External Site/Tool Integrations Build Reports
	This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	23 days ago
<input checked="" type="checkbox"/>	Config File Provider 880.v08956a_a_5d6e_d	Groovy-related External Site/Tool Integrations Maven
	Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.	1 mo 20 days ago
<input checked="" type="checkbox"/>	Maven Integration 3.24	Build Tools
	This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.	1 mo 19 days ago
<input checked="" type="checkbox"/>	Pipeline Maven Integration 1469.v15ca_a_b_90b_44	pipeline Maven
	This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path.	16 days ago
<input checked="" type="checkbox"/>	Pipeline Stage View 2.34	User Interface
	Pipeline Stage View Plugin.	1 yr 1 mo ago
<input checked="" type="checkbox"/>	Docker Pipeline 580.v0c340686b_54	pipeline DevOps Deployment docker
	Build and use Docker containers from pipelines.	6 mo 24 days ago
<input checked="" type="checkbox"/>	Kubernetes 4302.va_756e4b_67715	Cloud Providers Cluster Management kubernetes Agent Management
	This plugin integrates Jenkins with Kubernetes	7 days 13 hr ago
<input checked="" type="checkbox"/>	Kubernetes Client API 6.10.0-240-57880ce6b_0_2	kubernetes Library plugins (for use by other plugins)
	Kubernetes Client API plugin for use by other Jenkins plugins.	10 mo ago
<input checked="" type="checkbox"/>	Kubernetes Credentials 190.v03c305394de0_	kubernetes credentials
	Common classes for Kubernetes credentials	2 mo 25 days ago

Add the sonarqube,nexus,maven k8s & docker tokens in the credentials secyions and start writing the pipeline

Stage	Declarative Tool Install	Git Checkout	Compile	Tests	Trivy FS Scan	SonarQube Analysis	Quality Gate Check	Build	Published Artifact to Nexus	Docker Build & Tag Image	Trivy Image Scan	Docker Push Image	Deploy MySQL Deployment Service	Deploy SVC-APP	Deploy to Kubernetes	Switch Traffic Between Blue & Green Environment	Verify Deployment
Average stage times (Average 1st run time: ~5min-46s)	112ms	912ms	3s	3s	900ms	13s	295ms	4s	879ms	4s	5min 1s	6s	1s	660ms	659ms	787ms	664ms
Dec 12 21:51	118ms	881ms	3s	4s	7s	14s	330ms	5s	812ms	4s	5min 1s	7s	965ms	677ms	688ms	787ms	675ms
Dec 12 21:53	113ms	971ms	3s	3s	972ms	13s	273ms	4s	881ms	4s	5min 1s	6s	902ms	653ms	640ms		651ms
Dec 12 21:53	107ms	876ms	3s	3s	968ms	13s	277ms	4s	844ms	4s	5min 1s	6s	1s	690ms	643ms		666ms

**SonarQube Quality Gate**

- Last build (#39), 12 min ago
- Last stable build (#38), 21 min ago
- Last successful build (#38), 21 min ago
- Last completed build (#38), 21 min ago

## Pipeline code

Is given

Below

```

pipeline {
    agent any

    tools {
        maven 'maven3'
    }

    parameters {
        choice(name: 'DEPLOY_ENV', choices: ['blue', 'green'], description: 'Choose which environment to deploy: Blue or Green')
        choice(name: 'DOCKER_TAG', choices: ['blue', 'green'], description: 'Choose the Docker image tag for the deployment')
        booleanParam(name: 'SWITCH_TRAFFIC', defaultValue: false, description: 'Switch traffic between Blue and Green')
        booleanParam(name: 'SKIP_TRIVY_SCAN', defaultValue: false, description: 'Skip the Trivy scan stage')
        booleanParam(name: 'SKIP_QUALITY_GATE', defaultValue: false, description: 'Skip the Quality Gate Check stage')
    }
}

environment {
    IMAGE_NAME = "thug24/bankapp"
    TAG = "${params.DOCKER_TAG}"
    SCANNER_HOME = tool 'sonar-scanner'
    KUBE_NAMESPACE = 'webapps'
}
}

stages {
    stage('Git Checkout') {
        steps {
            git branch: 'main', credentialsId: 'git-cred', url: 'https://github.com/gadhasusaikrishna/blue-green-e.k.s.git'
        }
    }
}

stage('Compile') {
    steps {
        sh "mvn clean compile"
    }
}

```

```

stage('Tests') {
    steps {
        sh "mvn test -DskipTests=true"
    }
}

// Trivy FS Scan stage, only run if SKIP_TRIVY_SCAN is false
stage('Trivy FS Scan') {
    when {
        expression { return !params.SKIP_TRIVY_SCAN }
    }
    steps {
        sh "trivy fs --format table -o fs.html ."
    }
}

stage('SonarQube Analysis') {
    steps {
        withSonarQubeEnv('sonar') {
            sh "$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectKey=Multitier -Dsonar.projectName=Multitier -Dsonar.java.binaries=target"
        }
    }
}

// Quality Gate Check only if SKIP_QUALITY_GATE is false
stage('Quality Gate Check') {
    when {
        expression { return !params.SKIP_QUALITY_GATE }
    }
    steps {
        script {
            currentBuild.result = 'SUCCESS'
            echo "Quality Gate Check is bypassed and marked as green."
        }
    }
}

```

```
        }
    }
}

stage('Build') {
    steps {
        sh "mvn clean package -DskipTests"
    }
}

stage('Publish Artifact To Nexus') {
    steps {
        withMaven(globalMavenSettingsConfig: 'maven-settings') {
            // Fix the mvn command to properly skip tests
            sh "mvn deploy -DskipTests"
        }
    }
}

stage('Docker Build & Tag Image') {
    steps {
        script {
            withDockerRegistry(credentialsId: 'docker-credential') {
                sh "docker build -t ${IMAGE_NAME}:${TAG} ."
            }
        }
    }
}

stage('Docker Push Image') {
    steps {
        script {
            withDockerRegistry(credentialsId: 'docker-credential') {
                sh "docker push ${IMAGE_NAME}:${TAG} "
            }
        }
    }
}
```

```

        }
    }
}

stage('Deploy MySQL Deployment and Service') {
    steps {
        script {
            withKubeConfig(caCertificate: "", clusterName: 'devopsshack-cluster', contextName: "", credentialsId: 'k8-token',
namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl: 'https://133C1D104C905CF3677143280A07BF49.gr7.ap-south-1.eks.amazonaws.com') {
                sh "kubectl apply -f mysql-ds.yml -n ${KUBE_NAMESPACE}"
            }
        }
    }
}

stage('Deploy SVC-APP') {
    steps {
        script {
            withKubeConfig(caCertificate: "", clusterName: 'devopsshack-cluster', contextName: "", credentialsId: 'k8-token',
namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl: 'https://133C1D104C905CF3677143280A07BF49.gr7.ap-south-1.eks.amazonaws.com') {
                sh """
                if ! kubectl get svc bankapp-service -n ${KUBE_NAMESPACE}; then
                    kubectl apply -f bankapp-service.yml -n ${KUBE_NAMESPACE}
                fi
                """
            }
        }
    }
}

stage('Deploy to Kubernetes') {
    steps {
        script {
            def deploymentFile = """
            if (params.DEPLOY_ENV == 'blue') {

```

```

        deploymentFile = 'app-deployment-blue.yml'

    } else {

        deploymentFile = 'app-deployment-green.yml'

    }

    withKubeConfig(caCertificate: '', clusterName: 'devopsshack-cluster', contextName: '', credentialsId: 'k8-token',
namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl: 'https://133C1D104C905CF3677143280A07BF49.gr7.ap-south-1.eks.amazonaws.com') {

        sh "kubectl apply -f ${deploymentFile} -n ${KUBE_NAMESPACE}"

    }

}

}

}

stage('Switch Traffic Between Blue & Green Environment') {

when {

    expression { return params.SWITCH_TRAFFIC }

}

steps {

script {

    def newEnv = params.DEPLOY_ENV

    withKubeConfig(caCertificate: '', clusterName: 'devopsshack-cluster', contextName: '', credentialsId: 'k8-token',
namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl: 'https://133C1D104C905CF3677143280A07BF49.gr7.ap-south-1.eks.amazonaws.com') {

        sh """
        kubectl patch service bankapp-service -p "{\"spec\":{\"selector\":{\"app\": \"bankapp\"}}, \"version\": \"$newEnv\"}" -n ${KUBE_NAMESPACE}
        """
        """

    }

    echo "Traffic has been switched to the ${newEnv} environment."
}

}

}

stage('Verify Deployment') {

steps {

script {

```

```
def verifyEnv = params.DEPLOY_ENV

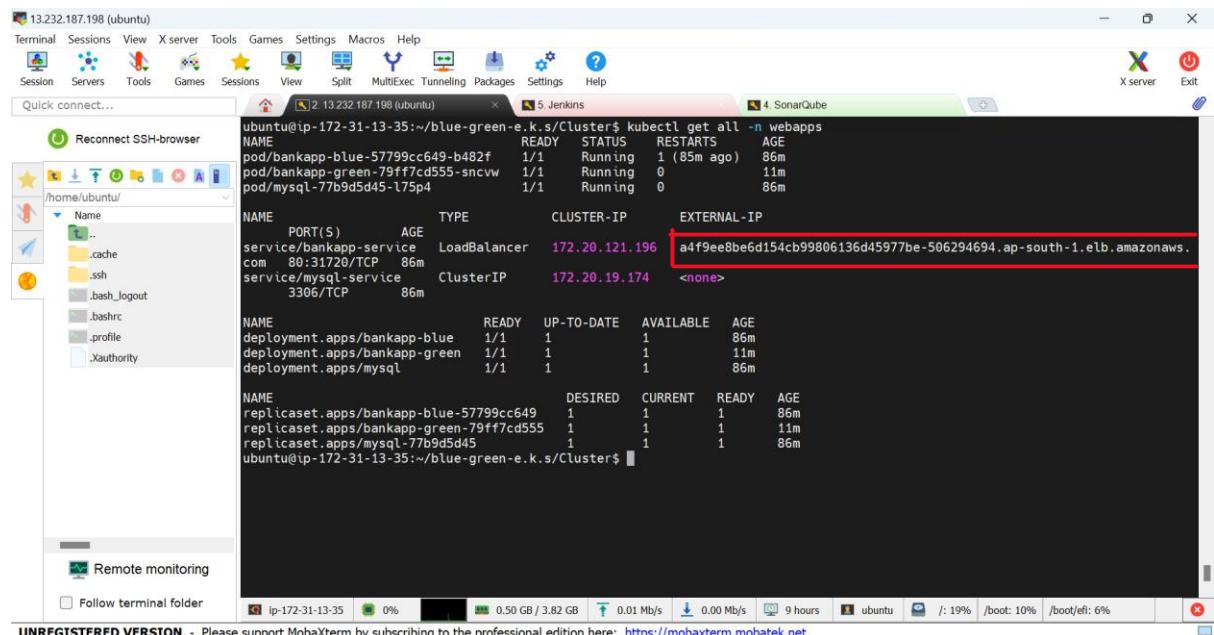
withKubeConfig(caCertificate: '', clusterName: 'devopsshack-cluster', contextName: '', credentialsId: 'k8-token',
namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl: 'https://133C1D104C905CF3677143280A07BF49.gr7.ap-south-1.eks.amazonaws.com') {

    sh """
        kubectl get pods -l version=${verifyEnv} -n ${KUBE_NAMESPACE}
        kubectl get svc bankapp-service -n ${KUBE_NAMESPACE}
    """

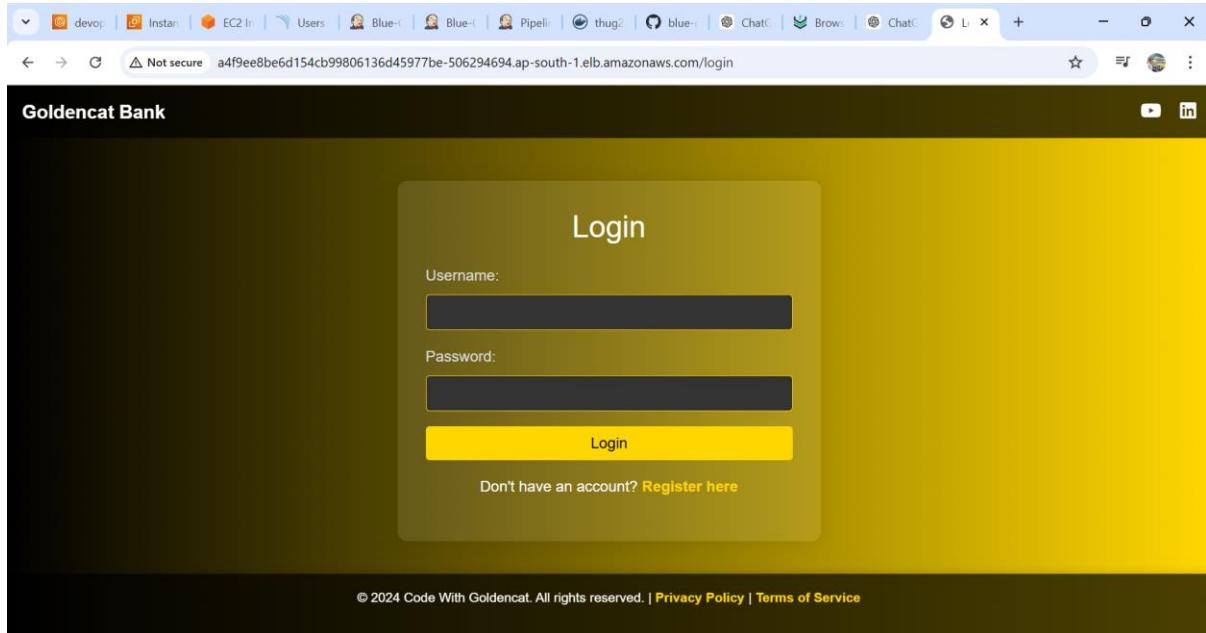
    }
}

}
}
```

```
kubectl get all -n webapps
```



Using external ip dashboard can be operated



## Nexus snapshots

The screenshot shows the Sonatype Nexus Repository interface. The left sidebar has a green 'Browse' tab selected. The main content area shows a tree view of Maven snapshot artifacts under 'com/example/bankapp'. Artifacts include versions like 0.0.1-SNAPSHOT, 0.0.1-20241212130006-1, and 0.0.1-20241212130639-2. There are also several .xml and .md5 files listed at the bottom of the artifact tree.

## SonarQube

The screenshot shows the SonarQube dashboard for the 'Multitier' project. The top navigation bar includes links for dev, Inst, EC2, User, Blue, Pipe, thug, blue, Chat, Brow, Chat, Log, and a plus sign. The main header shows the URL as 'Not secure 13.127.231.177:9000/dashboard?id=Multitier'. The dashboard features a dark header with 'sonarqube' logo and navigation tabs: Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. A message at the top right indicates 'Last analysis of this Branch had 3 warnings' on December 12, 2024 at 9:51 PM. Below this, the 'Project Settings' and 'Project Information' dropdowns are visible. The left sidebar has sections for Overview, Issues, Security Hotspots, Measures, Code, and Activity. The 'QUALITY GATE STATUS' section shows a large green box labeled 'Passed' with the message 'All conditions passed'. The 'MEASURES' section displays various metrics: 'New Code' (0 New Bugs, Reliability A), 'Overall Code' (0 New Vulnerabilities, Security A), 'New Security Hotspots' (0 Reviewed, Security Review A), 'Added Debt' (0 New Code Smells, Maintainability A). The overall interface is clean with a light gray background and green highlights for successful status.

## Docker hub

The screenshot shows the Docker Hub repository page for 'thug24/bankapp'. The top navigation bar includes links for dev, Inst, EC2, Use, Blue, Mul, Pipe, thug, blue, Chat, Brow, Chat, Log, and a plus sign. The main header shows the URL as 'hub.docker.com/r/thug24/bankapp'. The dashboard features a dark header with 'dockerhub' logo and navigation tabs: Explore, Repositories, Organizations, Usage, and a search bar. A message at the top right indicates 'New More Docker. Easy Access. New Streamlined Plans. Learn more.' Below this, the repository path 'Explore / thug24/bankapp' is shown. The repository card for 'thug24/bankapp' shows a blue cube icon, the repository name, 'By thug24 · Updated 4 minutes ago', and a 'Manage Repository' button. It also shows 'IMAGE', '0 stars', and '9 downloads'. The 'Overview' tab is selected, showing a placeholder message 'No overview available' with the note 'This repository doesn't have an overview'. To the right, a 'Docker Pull Command' section contains the command 'docker pull thug24/bankapp' with a 'Copy' button. The overall interface is modern with a blue and white color scheme.

