

Advance DevOps
Practical Exam

Introduction

Case Study Overview

The case study involves integrating static analysis tools with **Infrastructure-as-Code (IaC)** using Terraform. This project focuses on setting up a development environment that combines **Jenkins** for continuous integration and deployment, **SonarQube** for static code analysis, and **Terraform** for infrastructure provisioning. The goal is to create a robust pipeline that can automatically analyze Python applications for quality and security issues.

Key Features and Applications:

Key features of this case study include:

- Infrastructure deployment using Terraform
- Continuous Integration/Continuous Deployment (CI/CD) pipeline setup
- Static code analysis integration
- Cross-tool communication between Jenkins, SonarQube,

and Terraform. The practical applications of this setup are:

- Improved code quality through automated static analysis
- Enhanced security by identifying potential vulnerabilities early in the development cycle
- Streamlined development workflow with automated testing and reporting
- Scalable infrastructure management using Terraform

Step-by-Step

Explanation

Terraform

Step 1:

Install terraform and add it to environment variable. Now, download Amazon CLI by visiting the following website. Visit <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>



Search in this guide

Contact Us

English ▾

Return to the Console

AWS > Documentation > AWS Command Line Interface

> User Guide for Version 2

Feedback Preferences

AWS Command Line Interface

User Guide for Version 2

Recently added to this guide

Amazon ECR Public examples using AWS CLI
20 September 2024

Route 53 Profiles examples using AWS CLI
20 September 2024

Security Lake examples using AWS CLI
20 September 2024

About the AWS CLI

Get started

Prerequisites

Install/Update

Past releases

Build and install from source

Amazon ECR Public/Docker Setup

Configure the AWS CLI

Windows

Install and update requirements

- We support the AWS CLI on Microsoft-supported versions of 64-bit Windows.
- Admin rights to install software

Install or update the AWS CLI

To update your current installation of AWS CLI on Windows, download a new installer each time you update to overwrite previous versions. AWS CLI is updated regularly. To see when the latest version was released, see the [AWS CLI version 2 Changelog](#) on GitHub.

1. Download and run the AWS CLI MSI installer for Windows (64-bit):

<https://awscli.amazonaws.com/AWSCLIV2.msi>

Alternatively, you can run the `msiexec` command to run the MSI installer.

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
```

For various parameters that can be used with `msiexec`, see [msiexec](#) on the Microsoft Docs website. For example, you can use the `/qn` flag for a silent installation.

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi /qn
```

2. To confirm the installation, open the **Start** menu, search for `cmd` to open a command prompt

On this page

AWS CLI install and update instructions

Troubleshooting AWS CLI install and uninstall errors

Next steps

Now, click on install

<https://awscli.amazonaws.com/AWSCLIV2.msi> Complete

the installation process for AWSCLIV2

The screenshot shows a Windows File Explorer window titled "Downloads". The left sidebar shows navigation links like Home, Gallery, Anu - Personal, Desktop, and others. The main area displays a list of files in the "Downloads" folder. One file, "AWSCLIV2.msi", is highlighted. The file details are shown in the preview pane: Name: AWSCLIV2.msi, Date modified: 19-10-2024 02:06, Type: Windows Installer. A large green download icon is overlaid on the preview pane. On the right, there's a summary section for "Downloads (105 items)" with a note: "Select a single file to get more information and share your cloud content."

Name	Date modified	Type
AdvDevOps_practical	19-10-2024 02:45	Microsoft Word D...
AWSCLIV2	19-10-2024 02:06	Windows Installer ...
AMJenkins.pem	18-10-2024 19:23	PEM File
AMsonarqube.pem	18-10-2024 21:40	PEM File
Anuprita_Resume	13-10-2024 21:51	Microsoft Edge HT...
CNS_Assignment_4	14-10-2024 22:13	Microsoft Word D...
Siddhesh_Resume	17-10-2024 00:43	Adobe Acrobat D...
Anuprita_Resume	13-10-2024 21:55	Adobe Acrobat D...
CNS Assignment 4_Sneha	14-10-2024 22:14	Adobe Acrobat D...
CNS_Assignment_4	14-10-2024 22:13	Adobe Acrobat D...
DevOps_exp7	17-10-2024 02:44	Adobe Acrobat D...
DevOps_exp6	17-10-2024 23:34	Adobe Acrobat D...
DevOps_exp7	17-10-2024 02:43	Microsoft Word D...

Step 2:

Open AWS Academy and now click on AWS Details and then click on show button present in front of AWS CLI label. You will be shown with your credentials

The screenshot shows the AWS Academy interface. On the left is a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main navigation bar shows 'ALLv2EN-US...' > 'Modules' > 'AWS Acad...' > 'Launch AWS Academy Learner Lab'. The top right shows 'Used \$1.6 of \$50', '00:43', and buttons for 'Start Lab', 'End Lab', 'AWS Details', 'Readme', and 'Reset'. A dropdown menu shows 'EN-US'. The central area has a terminal window with the command 'eee_W_3429984@runweb141049:~\$'. To the right is a 'Learner Lab' panel with a 'Environment Overview' section containing links to various AWS services and a note that instructions were last updated on 2024-08-06.

This screenshot is similar to the one above, but the 'Cloud Access' panel is expanded. It contains a section titled 'AWS CLI:' with the instruction 'Copy and paste the following into ~/.aws/credentials'. Below this is a large block of AWS CLI configuration code. The rest of the interface is identical to the first screenshot.

```
[default]
aws_access_key_id=ASIASTK6MOBJNDGGSD5M
aws_secret_access_key=fAKG8vjAttCdl/T+6GCNE
w2xtuNTpm/ShUg1EE
aws_session_token=IqoJb3Jp22luX2VjEF1aCXvzI
Xd1c3QtMiJIMEYCIQDZY++GcCRp5qdsNvn6P1YYIKTr
IY3Db1MHR4LkbUfwA1hAL48020oLiavDT4lCoqRwm
dPLK3baqe9SPzA2zcCxYBKsECCLv/////////wEQAB
oMMTc40Tc10TYxHtewXgwXNBVHzIC5mpWz8q1QIMz
6H5S01h3zC7MSEgpJb7aV9BL8xNON4FnSKP580CrB0
tLSeluwd7ghxxUoQch6wK2QZ8QJVDc3dkLan8VuDQF
M7k1+glBkGhoDSuI2xfTPPEPjzbvCS/5tadSQ1NjXseI
cLNvXB/+74mXqkSXFr3QEsZGBc10/wr4j4RC0qEChz
I8q/LK10Z7xd/pp2w/KG2cD84Lev7jDeKFhzdHOrQ
dnRIuAtIUzjIYGf7Q+InQkZmj+XfkAdb+s+iFSYkFHB9
uHXG/1PtwiGFwiKX43GS60GQZd3k8jd8q/VDRBK3C
VF9GSpYSuN0UuaJfms9dZPF4r-RVKFq/dz01Jjy3evdb
```

Step 3:

Now, create a folder in VSCode and create a main.tf file in it with the following content.

```
# Specify the AWS provider
```

```
provider "aws" {
region = "us-east-1" # Replace with your preferred region
}
# Jenkins instance
resource "aws_instance" "jenkins" {
ami      = "ami-007868005aea67c54" # Amazon Linux 2 AMI
instance_type = "t2.micro"
key_name    = "SCjenkins"
tags = {
  Name = "JenkinsServer"
}
# User data to install Jenkins with Java 17
user_data = <<-EOF
#!/bin/bash
sudo yum update -y
sudo dnf install -y java-17-amazon-corretto-devel # Install Java 17
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
sudo yum install -y Jenkins
sudo systemctl start Jenkins
sudo systemctl enable Jenkins
EOF
}
# SonarQube instance
resource "aws_instance" "sonarqube" {
ami      = "ami-007868005aea67c54" # Amazon Linux 2 AMI
instance_type = "t2.medium"
key_name    = "SCsonarqube1"
tags = {
  Name = "SonarQubeServer"
}
# User data to install SonarQube manually
user_data = <<-EOF
#!/bin/bash
sudo yum update -y
sudo su -
cd /opt
wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-10.7.0.96327.zip
unzip sonarqube-10.7.0.96327.zip
sudo adduser sonar
sudo passwd sonar
sudo chown -R sonar:sonar /opt/sonarqube-10.7.0.96327
su - sonar -c "/opt/sonarqube-10.7.0.96327/bin/linux-x86-64/sonar.sh start"
EOF
}
```

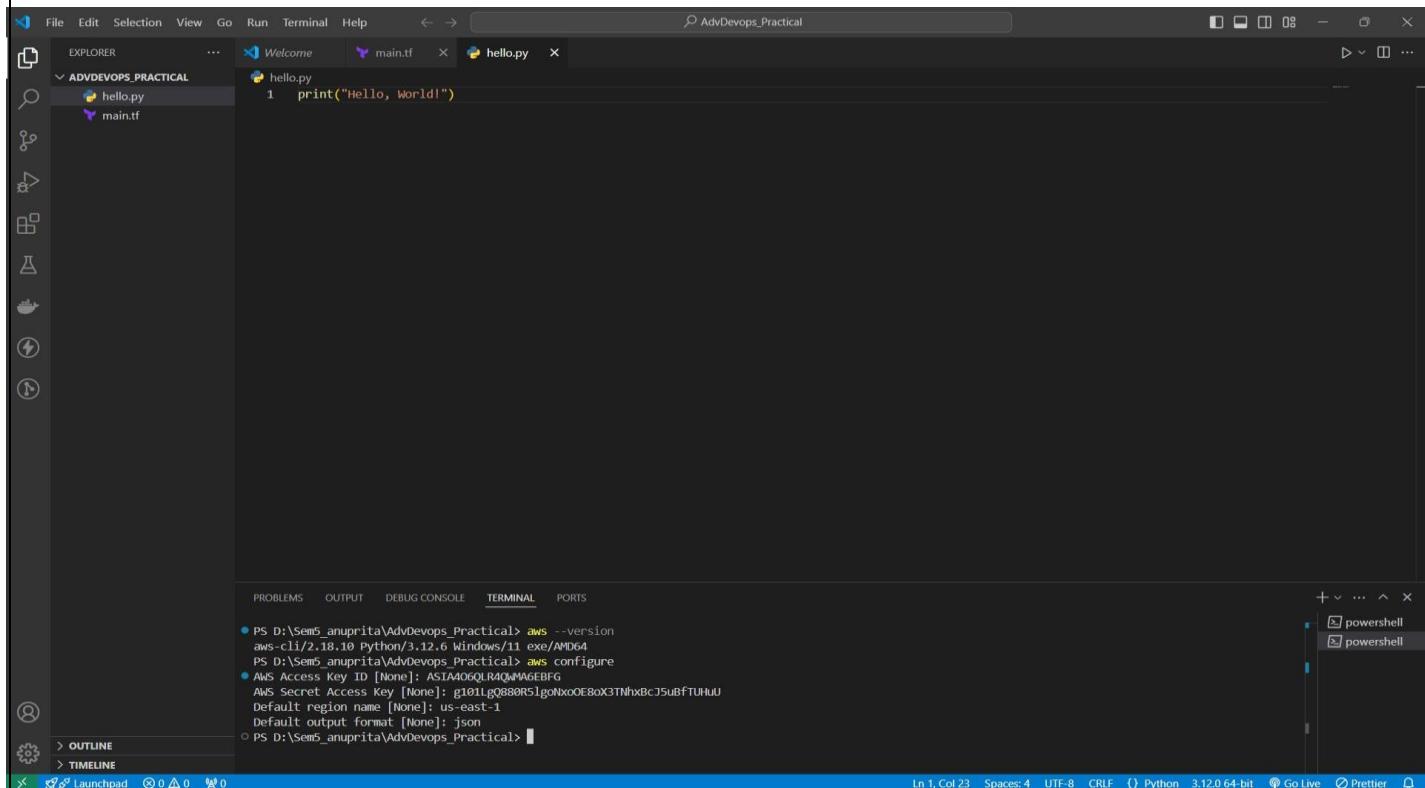
The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** SONARJENKIS-1 folder containing .terraform, .terraform.lock.hcl, main.tf, terraform.tfstate, and terraform.tfstate.backup.
- Editor:** The main window displays the `main.tf` file content.
- Code Content:**

```
1 # Specify the AWS provider
2 provider "aws" {
3   region = "us-east-1" # Replace with your preferred region
4 }
5 # Jenkins instance
6 resource "aws_instance" "jenkins" {
7   ami           = "ami-007868005aea67c54" # Amazon Linux 2 AMI
8   instance_type = "t2.micro"
9   key_name      = "SCjenkins"
10  tags = {
11    Name = "JenkinsServer"
12  }
13 # User data to install Jenkins with Java 17
14 user_data = <<-EOF
15 #!/bin/bash
16 sudo yum update -y
17 sudo dnf install -y java-17-amazon-corretto-devel # Install Java 17
18 sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
19 sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
20 sudo yum install -y Jenkins
21 sudo systemctl start Jenkins
22 sudo systemctl enable Jenkins
23 EOF
24 }
25 # SonarQube instance
26 resource "aws_instance" "sonarqube" {
27   ami           = "ami-007868005aea67c54" # Amazon Linux 2 AMI
28   instance_type = "t2.medium"
29   key_name      = "Scsonarqube1"
30   tags = {
31     Name = "SonarQubeServer"
32   }
33 # User data to install SonarQube manually
34 user_data = <<-EOF
35 #!/bin/bash
36 sudo yum update -y
37 sudo su -
```

Step 4:

Now, to ensure and run the aws cli commands in vs code terminal, run the following commandsaws –version
aws configure



The screenshot shows the VS Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** AdvDevops_Practical.
- Explorer:** Shows a folder named "ADDEVOPS_PRACTICAL" containing files "hello.py" and "main.tf".
- Terminal:** Active tab, showing command-line history:
 - PS D:\Sem5_anuprita\AdvDevops_Practical> aws --version
 - aws-cli/2.18.10 Python/3.12.6 Windows/11 exe/AMD64
 - PS D:\Sem5_anuprita\AdvDevops_Practical> aws configure
 - AWS Access Key ID [None]: ASIAA06QLRAQWMA6E9FG
 - AWS Secret Access Key [None]: gI01Lg0B8R5lgoNxOE8oX3TNhx8cJ5uBFTUhuU
 - Default region name [None]: us-east-1
 - Default output format [None]: json
- Bottom Status Bar:** Ln 1, Col 23, Spaces: 4, UTF-8, CRLF, Python 3.12.0 64-bit, Go Live, Prettier.

Step 5:

Now, run the following commands in the vs code terminal to set the credential secrets.

```
$env:AWS_ACCESS_KEY_ID="ASIASTK6MOBJERYLZBXP"
$env:AWS_SECRET_ACCESS_KEY="Pzclg4d5kf1dsk5ztX0vGeFDhWRD+XnI4FZNUqL4"
$env:AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjEO//////////wEaCXVzLXdIc3QtMiJIMEYCIQDFth+ai
gG/gP+Z6F3r+MqGoz
mnWIMrKNwWhAdlQybeowlhANM0Vv37FoO8JPYclXeF9WdzbBVNXuzCyL3CFTilbFITKrICCFgQARo
MODU2NzQ2MDY5Nzk
zlgfb754SQVtRN0LY6gqjwLRheg15jeclfhu956B4lw1wh7nh8uAbmksRgzTFn0f/XRdNDp8umD5361F
3CB0Kw0y3u2iUiappL
GRgmAIs6lpwx91OaD/IziE+J6jqA9werjjE+vPSwJZDjBAI3AOsXEqrHHOxOCMnV8tDADefFFdq4/TVcp
j7XrZZbr/bl9b33kA28Y
b02vxwFpXXwCjcuPtkae+kPC5v74IAb1S0gEXudqQv1okS7d+M6hg3qOqrIswTZJ52IZHDZYbeavRrU
5qv8W+ky95bOhy1fb3
HKUSGKnpS6Pu2IDqjG97SiAFiB6YCxBzq4UxFJWVreSMILwMPzbL7nq87z3lbRNHbJ3RyIWA0TLNB
wNNKT92DPyqMP+jzbgG
OpwBBmh89toqaGEkit4IMz7vMppk0w4Q6pgBb/qWs5QtACBIxE6MMqUMBvMquTpP7t3u48g6Z7/ebn
kNnf4xtfZ+MgcGY
pwWqkHhiBg3QoX4toPicq83phWThYfOaRop4D/V7h9en8dRRMLuYkEh545h55d+dHbzgop1JpKCLCv
cDO8jh3UGzDhSllvEB 5BYjnwrn24CiA3bX4yHa/2o"
```

```
PS C:\Users\PC\Desktop\Sonarjenkis-1> $env:AWS_ACCESS_KEY_ID="ASIASTK6MOBJERYLZBXP"
PS C:\Users\PC\Desktop\Sonarjenkis-1> $env:AWS_SECRET_ACCESS_KEY="Pzclg4d5kf1dsk5ztX0vGeFDhWRD+XnI4FZNUqL4"
PS C:\Users\PC\Desktop\Sonarjenkis-1> $env:AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjEE0aCXVzLXdIc3QtMiJHMEUCIEhbITmjMn+VLG0tfb2Q
eTETh903fyvLyIxP5EfA4PF3AiEA6qReIq6Jlx9wB2oxJVXRrI+izMH0RPFAxhAMfem8ezsqwQIItv//////////ARAAGgwXNzg5NzU5NjExNzAiDFd0oUOUof
p3Qv1moSqVApbMnA60Mo8w5l1l1qwlascx1YCz5imD4BodyvgIgCXBmZz+3czuFgehQ1BZ5b24xXBrgQmAvtb3uNhsjYjFy+ItKsmb1hz3VqpihFQ0tgmU9Bv1/4
Rc5kvdeYAmqmYMiFmBhnehQCaLJQQ+BVxZXZAiu2Ev7+7fGNPqb6LKdZLY6nzGdVYHAKhX8Z7soj/J6hrwhHwEl4xgiFl9NP4X2sodfQBjJp0YzekbnTuQgbg5
/YKi/mZ83jrawVvxHZX4M061P6nDoEYUjJhyQNURWY1VaA6l1v2Abh85bvsl2vCYza8qoQak/Axi/8LASArANu5onZqUor2eqK92T5BQ0QwzPHfNIIdPcpIbf
Tz5HIfNNZlEqLswsv3huAY6nQFiyfA9+n3uV3Qss5dkh4/6CLIjc713fdoh8zzsol08vUVISwlGM3kbKxAdvPoyqmnn1uw7fd8s8dRlf8NbUfGgH1PlEnoT+D7+
ciSPzlpYeaQnd1FnP436cGzbR6mK6AH5yAdeB6RT+r9EhuVp5RX5uAyjmAw/1uA07AyXp6Fetsc0KKbc1FAAyeg/yH0SQ+Lb1pkYQSeZIqnQ+L8Gn"
PS C:\Users\PC\Desktop\Sonarjenkis-1>
```

Step 6:

Now, to check whether the aws cli is connected to your aws account run the following command.
aws sts get-caller-identity
aws configure

```
PS C:\Users\PC\Desktop\Sonarjenkis-1> aws sts get-caller-identity
{
    "UserId": "AROASTK6MOBJK2PFZZWFI:user3396434=CHHABAIDIYA SONAM_SURESH_KUMAR",
    "Account": "178975961170",
    "Arn": "arn:aws:sts::178975961170:assumed-role/voclabs/user3396434=CHHABAIDIYA SONAM_SURESH_KUMAR"
}

PS C:\Users\PC\Desktop\Sonarjenkis-1> aws configure
AWS Access Key ID [*****ZBXP]: ASIASTK6MOBJERYLZBXP
AWS Secret Access Key [*****Uql4]: Pzclg4d5kf1dsk5ztX0vGeFDhWRD+XnI4FZNUqL4
Default region name [us-east-1]:
PS C:\Users\PC\Desktop\Sonarjenkis-1> aws configure
AWS Access Key ID [*****ZBXP]: ASIASTK6MOBJERYLZBXP
AWS Secret Access Key [*****Uql4]: Pzclg4d5kf1dsk5ztX0vGeFDhWRD+XnI4FZNUqL4
Default region name [us-east-1]: us-east-1
```

Step 7:

Now, to get the AMI ID run the following command and select any of the AMI ID and replace the AMI ID present in main.tf file in VSCode.

```
aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-ami-hvm-2.0.*-x86_64-gp2" --query "Images[*].[ImageId,Name]" --region us-east-1 --output table
```

```
Default region name [us-east-1]: us-east-1
PS C:\Users\PC\Desktop\Sonarjenkins-1> aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-ami-hvm-2.0.*-x86_64-gp2" --query "Images[*].[ImageId,Name]" --region us-east-1 --output table
+-----+
|             DescribeImages           |
+-----+
| ami-007868005aea67c54 | amzn2-ami-hvm-2.0.20230119.1-x86_64-gp2 |
| ami-01e3c4a339a264cc9 | amzn2-ami-hvm-2.0.20241014.0-x86_64-gp2 |
| ami-014d544cfef21b42d | amzn2-ami-hvm-2.0.20240223.0-x86_64-gp2 |
| ami-0241b1d769b029352 | amzn2-ami-hvm-2.0.20240620.0-x86_64-gp2 |
| 03.3-x86_64-gp2          |
| 18.0-x86_64-gp2          |
| 12.0-x86_64-gp2          |
+-- More --+
```

After this run the following

command

```
terraform init
```

```
terraform apply
```

```
PS C:\Users\PC\Desktop\Sonarjenkins-1> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.72.1...
- Installed hashicorp/aws v5.72.1 (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.
```

```
aws_instance.sonarqube: Creating...
aws_instance.jenkins: Creating...
aws_instance.jenkins: Creating...
aws_instance.sonarqube: Still creating... [10s elapsed]
aws_instance.jenkins: Still creating... [10s elapsed]
aws_instance.sonarqube: Still creating... [20s elapsed]
aws_instance.jenkins: Still creating... [20s elapsed]
aws_instance.sonarqube: Still creating... [10s elapsed]
aws_instance.jenkins: Still creating... [10s elapsed]
aws_instance.sonarqube: Still creating... [20s elapsed]
aws_instance.jenkins: Still creating... [20s elapsed]
aws_instance.sonarqube: Still creating... [30s elapsed]
aws_instance.jenkins: Still creating... [30s elapsed]
aws_instance.sonarqube: Still creating... [30s elapsed]
aws_instance.sonarqube: Creation complete after 33s [id=i-053beecb54b82ce6f]
aws_instance.sonarqube: Creation complete after 33s [id=i-053beecb54b82ce6f]
aws_instance.jenkins: Creation complete after 34s [id=i-0fe08e33b6d2aceee]
aws_instance.jenkins: Creation complete after 34s [id=i-0fe08e33b6d2aceee]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Users\PC\Desktop\Sonarjenkins-1>
```

Step 8:

After this Terraform will automatically create 2 EC2 instances on the EC2 Dashboard. To check the running instances: Visit AWS EC2 Dashboard.

The screenshot shows the AWS EC2 Instances dashboard. The left sidebar lists various EC2-related services like EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The main pane displays a table titled 'Instances (2) Info' with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4, and Elastic IP. Two instances are listed: 'JenkinsServer' (Instance ID i-0fe08e33b6d2aceee, t2.micro, 2/2 checks passed, us-east-1a, ec2-3-85-3-21.compute..., 3.85.3.21, -) and 'SonarQubeServer' (Instance ID i-053beccb54b82ce6f, t2.medium, 2/2 checks passed, us-east-1d, ec2-3-86-103-58.compute..., 3.86.103.58, -). A search bar at the top says 'Find Instance by attribute or tag (case-sensitive)' and a filter button says 'Clear filters'. The status bar at the bottom indicates 'Last updated 25 minutes ago'.

Set up Security Groups for the given two instances Step 1:

Go to EC2 Dashboard and select the Security Groups present in the left pane or sidebar and the click on the create security group.

- a. Create a security group with name SCjenkins-security and give some description and add the inbounds rules given below.
- b. Create a security group with name SCsonarqube1-security and give some description and add the inbounds rules given below.

The screenshot shows the AWS Security Groups dashboard. The left sidebar lists EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The main pane displays a table titled 'Security Groups (4) Info' with columns: Name, Security group ID, Security group name, VPC ID, Description, and Owner. Four security groups are listed: 'default' (sg-0b8e1d67d3e059ec0, default, vpc-07e15a80ff8cff437, default VPC security group, 178975961170), 'SCjenkins-security' (sg-04c78591a36acb9bd, SCjenkins-security, vpc-07e15a80ff8cff437, Allows SSH access to developers, 178975961170), 'launch-wizard-1' (sg-0ff00c0f2b4d69ee64, launch-wizard-1, vpc-07e15a80ff8cff437, launch-wizard-1 created 2024-10-13T..., 178975961170), and 'SCsonarqube-security' (sg-0066a4041317ab712, SCsonarqube-security, vpc-07e15a80ff8cff437, Allows SSH access to developers, 178975961170). A search bar at the top says 'Find resources by attribute or tag' and a button at the top right says 'Create security group'.

S E C U R I T Y G R O U P S

EC2 > Security Groups > sg-04c78591a36acb9bd - SCjenkins-security > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-044cf5ad2e5762f5e	Custom TCP	TCP	8080	Custom	<input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>
sgr-0f613c46c2b50f522	SSH	TCP	22	Custom	<input type="text" value="125.99.93.18/32"/> <input type="button" value="X"/>

Add rule

⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. **X**

Cancel **Preview changes** **Save rules**

S E C U R I T Y G R O U P S

EC2 > Security Groups > sg-00d6a4041317ab712 - Scsonarqube-security > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-0893791bd7dc83ddb	Custom TCP	TCP	9000	Custom	<input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>
sgr-09ab6da74b2465339	SSH	TCP	22	Custom	<input type="text" value="125.99.93.18/32"/> <input type="button" value="X"/>

Add rule

⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. **X**

Cancel **Preview changes** **Save rules**

Installation for Jenkins

Reference Video: <https://www.youtube.com/watch?v=bNuAS52ebLs>

Step 1:

Click on the JenkinsServer and click on connect.

The screenshot shows the AWS EC2 Instances page. There are two instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP
JenkinsServer	i-0fe08e33b6d2aceee	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-3-85-3-21.compute...	3.85.3.21
SonarQubeSer...	i-053beeb54b82ce6f	Running	t2.medium	2/2 checks passed	View alarms	us-east-1d	ec2-3-86-103-58.comp...	3.86.103.58

Details for the JenkinsServer instance:

- Public IPv4 address: 3.85.3.21 | [open address](#)
- Private IP4 addresses: 172.31.25.217
- Public IP DNS name (IPv4 only): ec2-3-85-3-21.compute-1.amazonaws.com | [open address](#)
- Instance state: Running
- Hostname type: IP name: ip-172-31-25-217.ec2.internal

The screenshot shows the 'Connect to instance' dialog for the JenkinsServer instance. The SSH client tab is selected.

EC2 Instance Connect | Session Manager | **SSH client** | EC2 serial console

Instance ID: i-0fe08e33b6d2aceee (JenkinsServer)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is SCjenkins.pem.
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "SCjenkins.pem"
4. Connect to your instance using its Public DNS:
ec2-3-85-3-21.compute-1.amazonaws.com

Example:
ssh -i "SCjenkins.pem" ec2-user@ec2-3-85-3-21.compute-1.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Step 2:

Open Git Bash and go to the directory which has the Key downloaded. If you don't have the key downloaded, create akey pair and download the .pem file for the key.

Since, I have the key downloaded in Downloads directory, I used the following commands:

```
cd Downloads
```

```
dir AMjenkins.pem*
```

```
ssh -i "AMjenkins.pem" ec2-user@ec2-98-80-223-40.compute-1.amazonaws.com
```

```
MINGW64:/c/Users/PC/Downloads
```

```
PC@ASUS MINGW64 ~ (master)
$ cd ~/Downloads

PC@ASUS MINGW64 ~/Downloads (master)
$ ssh -i "SCjenkins.pem" ec2-user@ec2-3-80-36-179.compute-1.amazonaws.com
The authenticity of host 'ec2-3-80-36-179.compute-1.amazonaws.com (3.80.36.179)' can't be established.
ED25519 key fingerprint is SHA256:+4IZQUb0AaK6xoSYyEyWzhDcMq9o3v68J7k0JtCzw60.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-80-36-179.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      #_
      ~\_ ####_          Amazon Linux 2
      ~~ \_\#\#\#/\_
      ~~   \#\#\|          AL2 End of Life is 2025-06-30.
      ~~     \#/_
      ~~       V~' '-->
      ~~~      /_
      ~~~-.  /_
      _/-' /_
      _/m/'-'_ A newer version of Amazon Linux is available!
      _/-' /_
      _/m/'-'_ Amazon Linux 2023, GA and supported until 2028-03-15.
                  https://aws.amazon.com/linux/amazon-linux-2023/
```

Step 3:

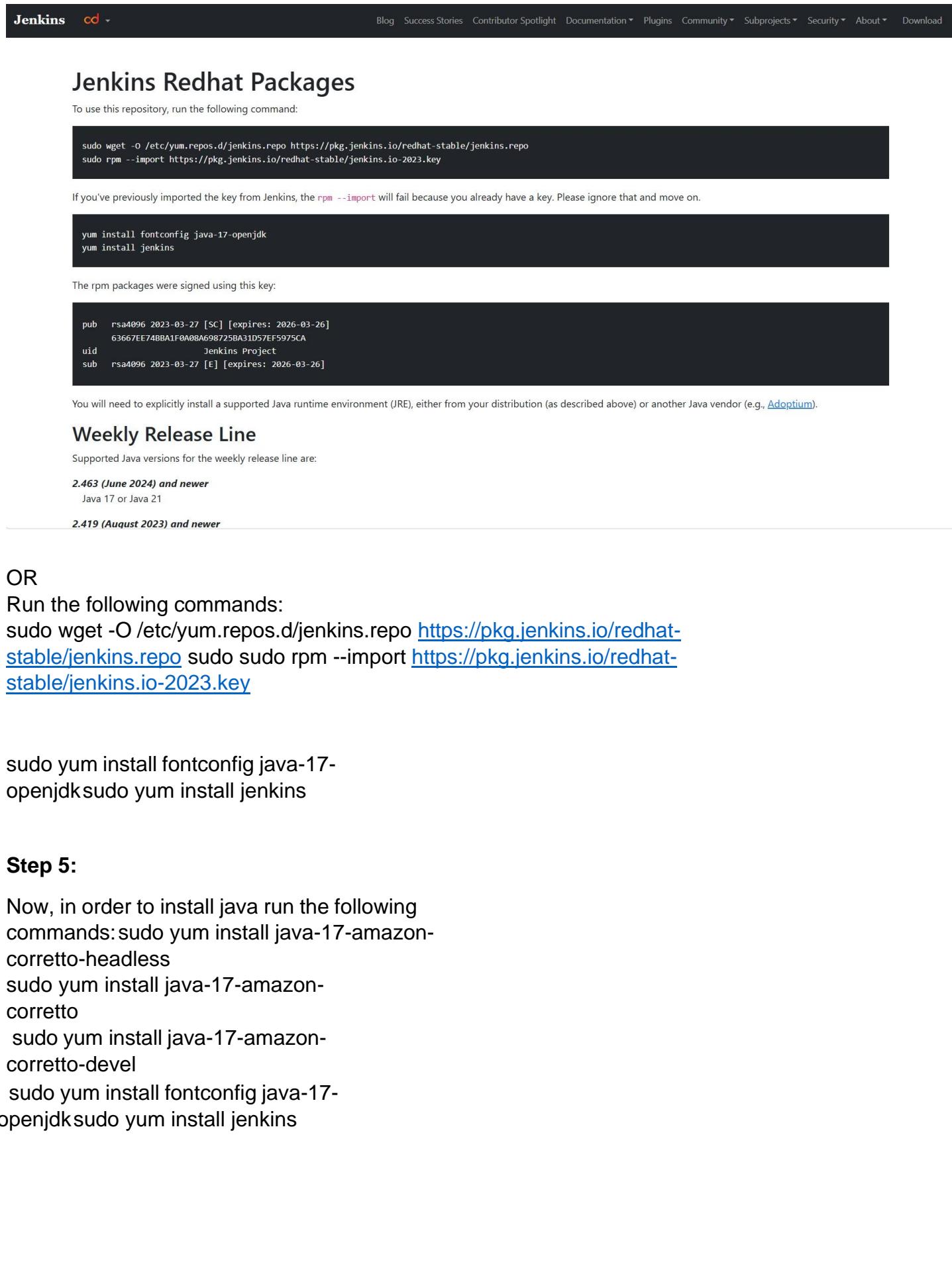
Go to google and search for Jenkins and then click on the Download and Deploy Link. Else, navigate using the following link: <https://pkg.jenkins.io/redhat-stable/>

Google search results for 'jenkins':

- Jenkins**
https://www.jenkins.io
Jenkins – an open source automation server which enables developers around the world to reliably build, test, and deploy their software.
 - Download and deploy**
Jenkins – an open source automation server which ...
 - Installing Jenkins**
Jenkins – an open source automation server which ...
 - Jenkins User Documentation**
Installing Jenkins - Pipeline - User Handbook Overview - Blue Ocean
 - Tutorials overview**
Jenkins – an open source automation server which ...
 - Pipeline**
A suite of plugins which supports implementing and integrating ...
 - More results from jenkins.io »
- Jenkins**
Software
 Jenkins is an open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration, and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat. [Wikipedia](#)

Step 4:

Now, run the initial 2 commands as it is and then run the next 2 commands using sudo word in the first; to run as rootuser.



The screenshot shows the Jenkins Redhat Packages page. At the top, there's a navigation bar with links for Blog, Success Stories, Contributor Spotlight, Documentation, Plugins, Community, Subprojects, Security, About, and Download. Below the navigation bar, the main title is "Jenkins Redhat Packages". A sub-section titled "To use this repository, run the following command:" contains the following code block:

```
sudo wget -o /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

Below this, a note states: "If you've previously imported the key from Jenkins, the `rpm --import` will fail because you already have a key. Please ignore that and move on." Another code block follows:

```
yum install fontconfig java-17-openjdk
yum install jenkins
```

Below these code blocks, a note says: "The rpm packages were signed using this key:" followed by a public key fingerprint:

```
pub rsa4096 2023-03-27 [SC] [expires: 2026-03-26]
  63667EE74BBA1F0A08A698725BA31D57EF5975CA
uid Jenkins Project
sub rsa4096 2023-03-27 [E] [expires: 2026-03-26]
```

At the bottom of the page, there's a note: "You will need to explicitly install a supported Java runtime environment (JRE), either from your distribution (as described above) or another Java vendor (e.g., [Adoptium](#))."

Weekly Release Line

Supported Java versions for the weekly release line are:

- 2.463 (June 2024) and newer**
Java 17 or Java 21
- 2.419 (August 2023) and newer**

OR

Run the following commands:

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

Step 5:

Now, in order to install java run the following commands:

```
sudo yum install java-17-amazon-corretto-headless
sudo yum install java-17-amazon-corretto
sudo yum install java-17-amazon-corretto-devel
sudo yum install fontconfig java-17-openjdk
sudo yum install jenkins
```

```
[ec2-user@ip-172-31-25-217 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-10-23 06:06:07-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:79::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

100%[=====] 2024-10-23 06:06:07 (7.47 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[ec2-user@ip-172-31-25-217 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[ec2-user@ip-172-31-25-217 ~]$ sudo yum install fontconfig java-17-openjdk
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No package java-17-openjdk available.
Resolving Dependencies
--> Running transaction check
--> Package fontconfig.x86_64 0:2.13.0-4.3.amzn2 will be installed
--> Processing Dependency: dejavu-sans-fonts for package: fontconfig-2.13.0-4.3.amzn2.x86_64
--> Processing Dependency: fontpackages-filesystem for package: fontconfig-2.13.0-4.3.amzn2.x86_64
--> Running transaction check
--> Package dejavu-sans-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Processing Dependency: dejavu-fonts-common = 2.33-6.amzn2 for package: dejavu-sans-fonts-2.33-6.amzn2.noarch
--> Package fontpackages-filesystem.noarch 0:1.44-8.amzn2 will be installed
--> Running transaction check
--> Package dejavu-fonts-common.noarch 0:2.33-6.amzn2 will be installed
--> Finished Dependency Resolution
```

```
Complete!
[ec2-user@ip-172-31-25-217 ~]$ sudo yum install java-17-amazon-corretto-headless
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package java-17-amazon-corretto-headless.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Processing Dependency: jpackage-utils for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-serif-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-mono-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: alsa-lib for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: log4j-cve-2021-44228-cve-mitigations for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Running transaction check
--> Package alsa-lib.x86_64 0:1.1.4.1-2.amzn2 will be installed
--> Package dejavu-sans-mono-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Package dejavu-serif-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Package javapackages-tools.noarch 0:3.4.1-11.amzn2 will be installed
--> Processing Dependency: python-javapackages = 3.4.1-11.amzn2 for package: javapackages-tools-3.4.1-11.amzn2.noarch
--> Processing Dependency: libxslt for package: javapackages-tools-3.4.1-11.amzn2.noarch
--> Package log4j-cve-2021-44228-hotpatch.noarch 0:1.3-7.amzn2 will be installed
--> Running transaction check
--> Package libxslt.x86_64 0:1.1.28-6.amzn2 will be installed
--> Package python-javapackages.noarch 0:3.4.1-11.amzn2 will be installed
--> Processing Dependency: python-lxml for package: python-javapackages-3.4.1-11.amzn2.noarch
--> Running transaction check
--> Package python-lxml.x86_64 0:3.2.1-4.amzn2.0.6 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

Package	Arch	Version	Repository
Installing:			
java-17-amazon-corretto-headless	x86_64	1:17.0.12+7-1.amzn2.1	amzn2-co
Installing for dependencies:			
alsa-lib	x86_64	1.1.4.1-2.amzn2	amzn2-co
dejavu-sans-mono-fonts	noarch	2.33-6.amzn2	amzn2-co
dejavu-serif-fonts	noarch	2.33-6.amzn2	amzn2-co
javapackages-tools	noarch	3.4.1-11.amzn2	amzn2-co
libxslt	x86_64	1.1.28-6.amzn2	amzn2-co
log4j-cve-2021-44228-hotpatch	noarch	1.3-7.amzn2	amzn2-co
python-javapackages	noarch	3.4.1-11.amzn2	amzn2-co
python-lxml	x86_64	3.2.1-4.amzn2.0.6	amzn2-co

```
Installed:
  java-17-amazon-corretto.x86_64 1:17.0.12+7-1.amzn2.1
```

```
Dependency Installed:
  giflib.x86_64 0:4.1.6-9.amzn2.0.2      libICE.x86_64 0:1.0.9-9.amzn2.0.2      libSM.x86_64 0:1.2.2-2.amzn2.0.2      libX11.x86_64 0:1.6.7-3.amzn2.0.5
  libXau.x86_64 0:1.0.8-2.1.amzn2.0.2     libXext.x86_64 0:1.3.3-3.amzn2.0.2      libXi.x86_64 0:1.7.9-1.amzn2.0.2      libXinerama.x86_64 0:1.3-2.1.amzn2.0.2
  libXrender.x86_64 0:0.9.10-1.amzn2.0.2   libXt.x86_64 0:1.1.5-3.amzn2.0.2      libXtst.x86_64 0:1.2.3-1.amzn2.0.2    libxcb.x86_64 0:1.12-1.amzn2.0.2
```

Complete!

```
[ec2-user@ip-172-31-25-217 ~]$ sudo dnf install java-17-amazon-corretto-devel
sudo: dnf: command not found
[ec2-user@ip-172-31-25-217 ~]$ sudo dnf install java-17-amazon-corretto-devel
sudo: dnf: command not found
[ec2-user@ip-172-31-25-217 ~]$ ^C
[ec2-user@ip-172-31-25-217 ~]$ sudo yum install -y java-17-amazon-corretto-devel
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package java-17-amazon-corretto-devel.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

Package	Arch	Version	Repository
Installing:			
java-17-amazon-corretto-devel	x86_64	1:17.0.12+7-1.amzn2.1	amzn2-co
Transaction Summary			
Install 1 Package			
Total download size: 146 k			
Installed size: 940 k			
Downloading packages:			
java-17-amazon-corretto-devel-17.0.12+7-1.amzn2.1.x86_64.rpm			
Running transaction check			
Running transaction test			
Transaction test succeeded			
Running transaction			

```

Complete!
[ec2-user@ip-172-31-25-217 ~]$ java --version
openjdk 17.0.12 2024-07-16 LTS
OpenJDK Runtime Environment Corretto-17.0.12.7.1 (build 17.0.12+7-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.12.7.1 (build 17.0.12+7-LTS, mixed mode, sharing)
[ec2-user@ip-172-31-25-217 ~]$ sudo yum install Jenkins
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No package Jenkins available.
 * Maybe you meant: jenkins
Error: Nothing to do
[ec2-user@ip-172-31-25-217 ~]$ sudo yum install jenkins
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package jenkins-2.462.3-1.1.noarch already installed and latest version
Nothing to do
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
      Active: inactive (dead)
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
      Active: inactive (dead)
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
      Active: active (running) since Wed 2024-10-23 06:11:41 UTC; 27s ago
        Main PID: 12266 (java)
       CGroup: /system.slice/jenkins.service
               └─12266 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

```

Step 6:

Run the following commands:
 sudo yum
 install Jenkins
 sudo systemctl status Jenkins
 sudo systemctl enable jenkins
 sudo systemctl status Jenkins
 sudo systemctl start Jenkins
 sudo systemctl status jenkins

```

Error: Nothing to do
[ec2-user@ip-172-31-25-217 ~]$ sudo yum install jenkins
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package jenkins-2.462.3-1.1.noarch already installed and latest version
Nothing to do
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
      Active: inactive (dead)
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
      Active: inactive (dead)
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-25-217 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
    Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
      Active: active (running) since Wed 2024-10-23 06:11:41 UTC; 27s ago
        Main PID: 12266 (java)
       CGroup: /system.slice/jenkins.service
               └─12266 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Oct 23 06:11:35 ip-172-31-25-217.ec2.internal jenkins[12266]: dbac51ecc97420ebd2b99e3d9f5a46e
Oct 23 06:11:35 ip-172-31-25-217.ec2.internal jenkins[12266]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Oct 23 06:11:35 ip-172-31-25-217.ec2.internal jenkins[12266]: ****
Oct 23 06:11:35 ip-172-31-25-217.ec2.internal jenkins[12266]: ****
Oct 23 06:11:35 ip-172-31-25-217.ec2.internal jenkins[12266]: ****
Oct 23 06:11:41 ip-172-31-25-217.ec2.internal jenkins[12266]: 2024-10-23 06:11:41.844+0000 [id=33]      INFO  jenkins.InitReactorRunner$1#onAttained: completed
Oct 23 06:11:41 ip-172-31-25-217.ec2.internal jenkins[12266]: 2024-10-23 06:11:41.868+0000 [id=24]      INFO  hudson.lifecycle.Lifecycle#onReady: Jenkins is
Oct 23 06:11:41 ip-172-31-25-217.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Oct 23 06:11:41 ip-172-31-25-217.ec2.internal jenkins[12266]: 2024-10-23 06:11:41.960+0000 [id=48]      INFO  h.m.DownloadService$Downloadable#load: obtained
Oct 23 06:11:41 ip-172-31-25-217.ec2.internal jenkins[12266]: 2024-10-23 06:11:41.961+0000 [id=48]      INFO  hudson.util.Retriger#start: Performed the action
Hint: Some lines were ellipsized. Use -l to show in full.

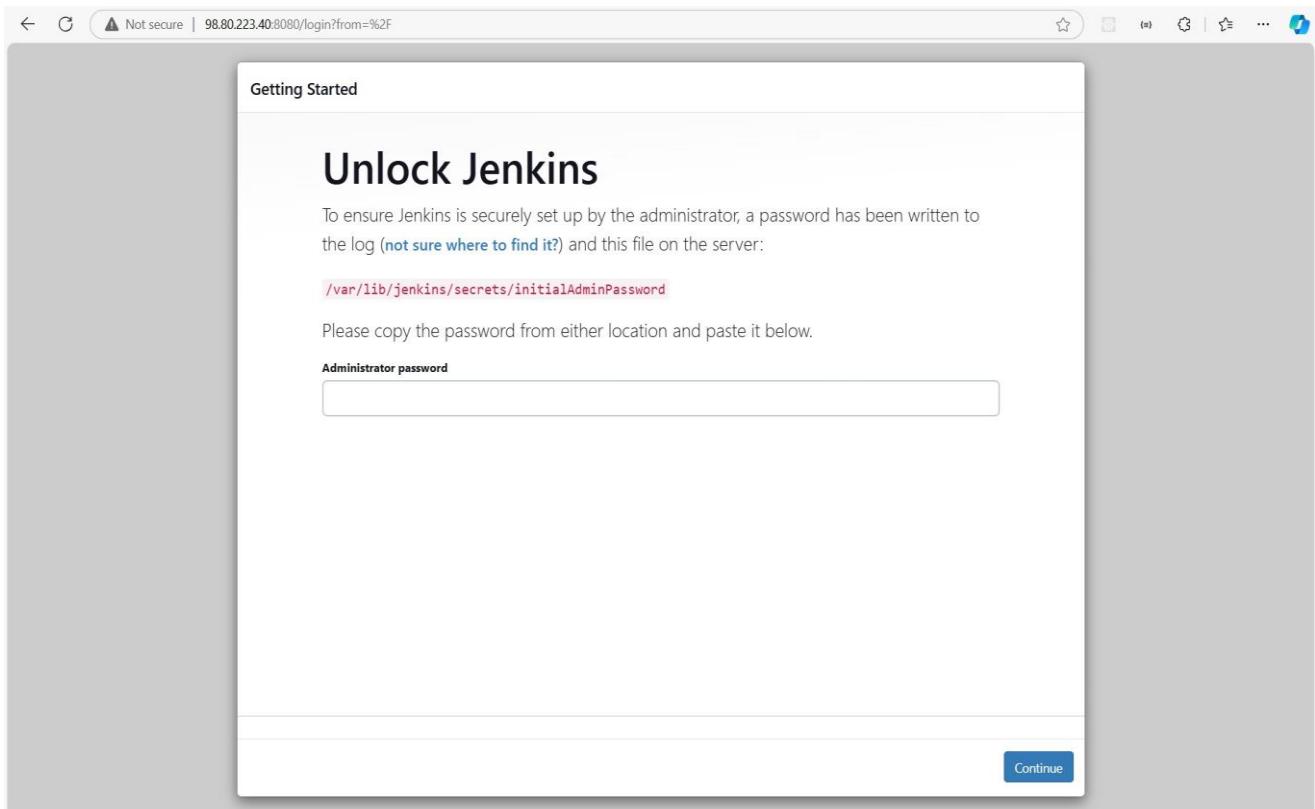
```

Step 7:

Now, go to EC2 dashboard and select Jenkins server and copy its public address and visit <http://<public-address>:8080>

Step 8:

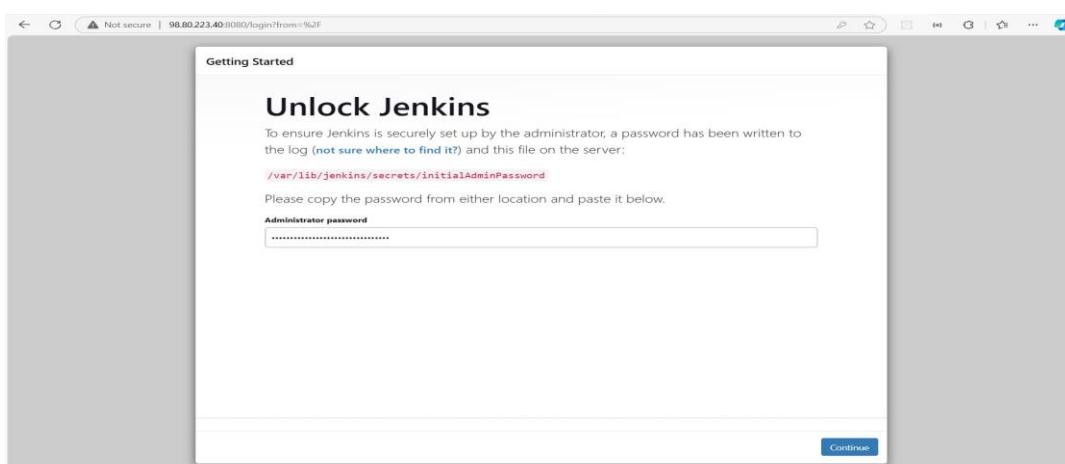
You will be redirected to this page on successful installation of Jenkins and visiting the public address url with port 8080



Step 9:

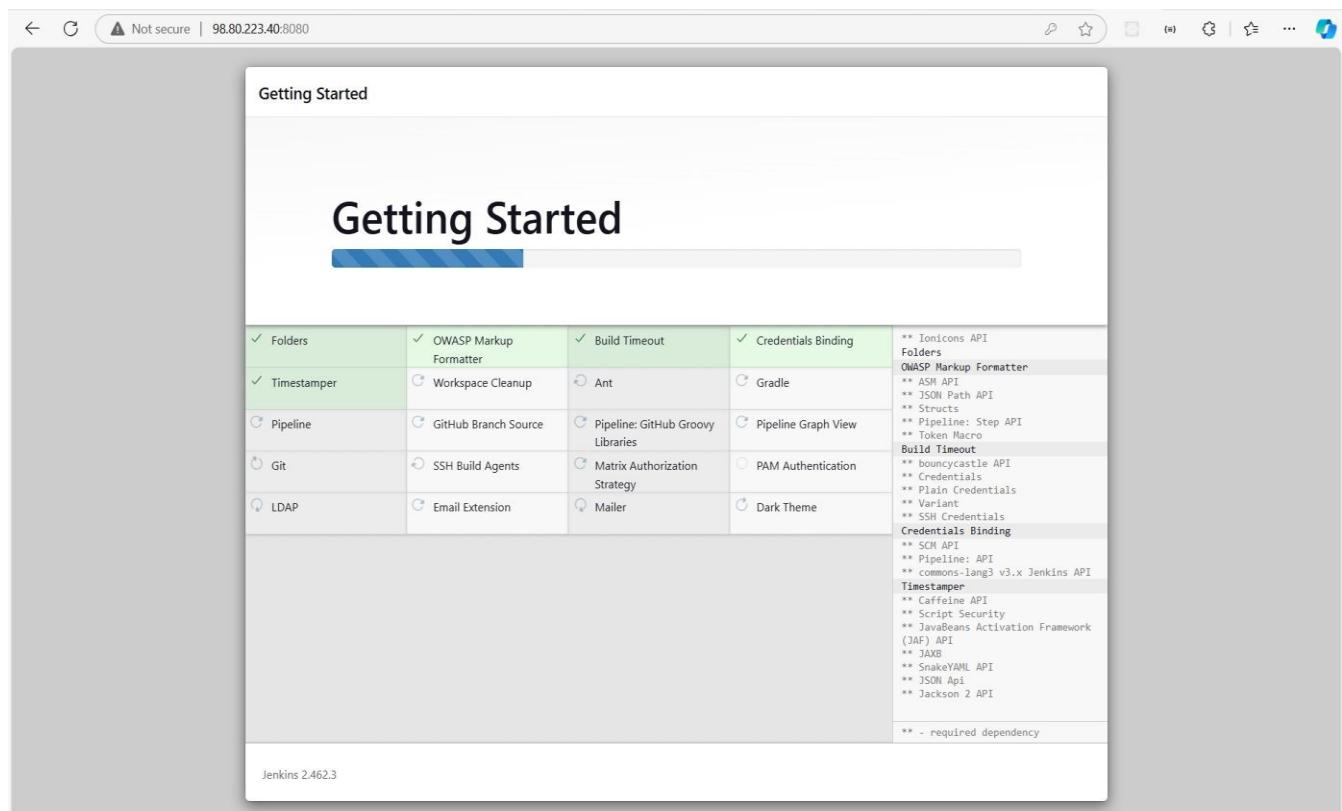
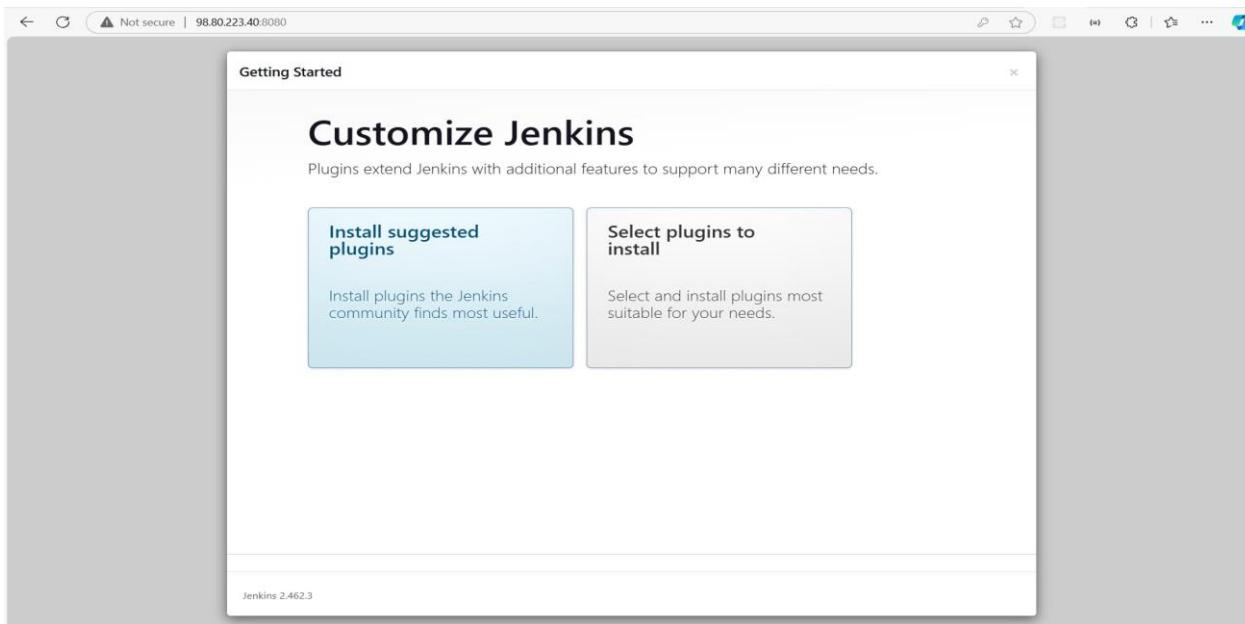
Now, come back to gitbash run the command
sudo more /var/lib/jenkins/secrets/initialAdminPassword

And, copy the content in the output and paste it in the input of Administrator password



Step 10:

Select install suggested plugins and complete the installation and initial configurations



Getting Started

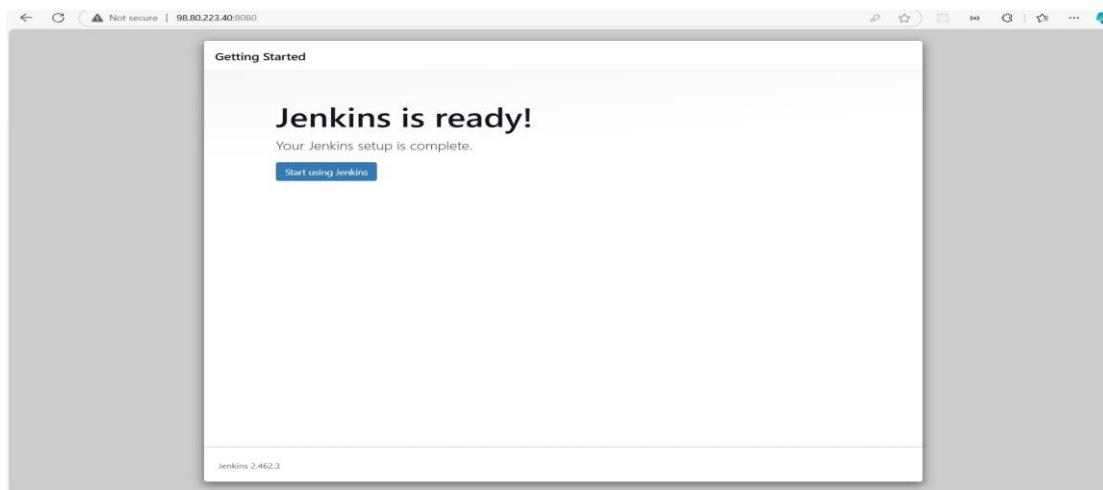
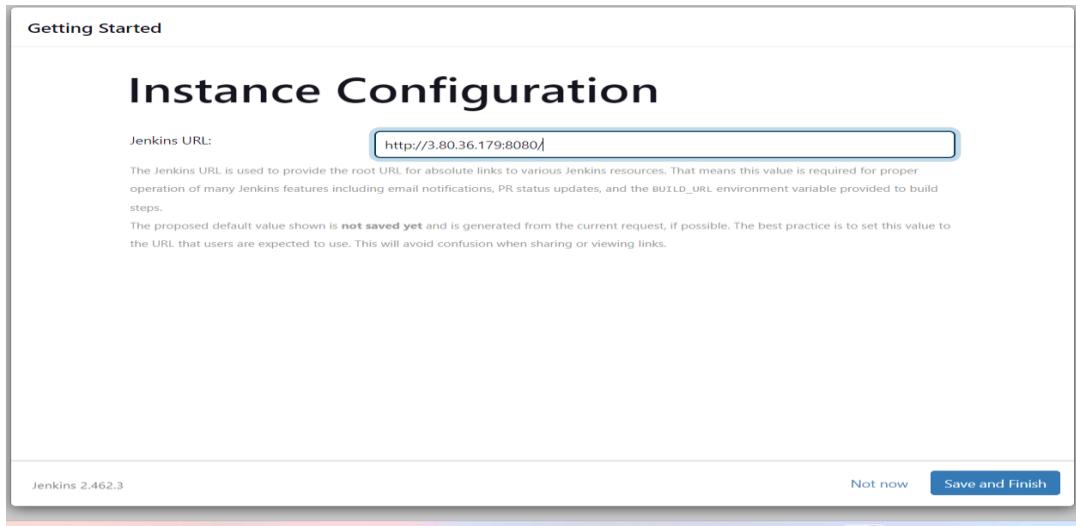
Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.462.3 Not now Save and Finish



Getting Started

Jenkins is ready!

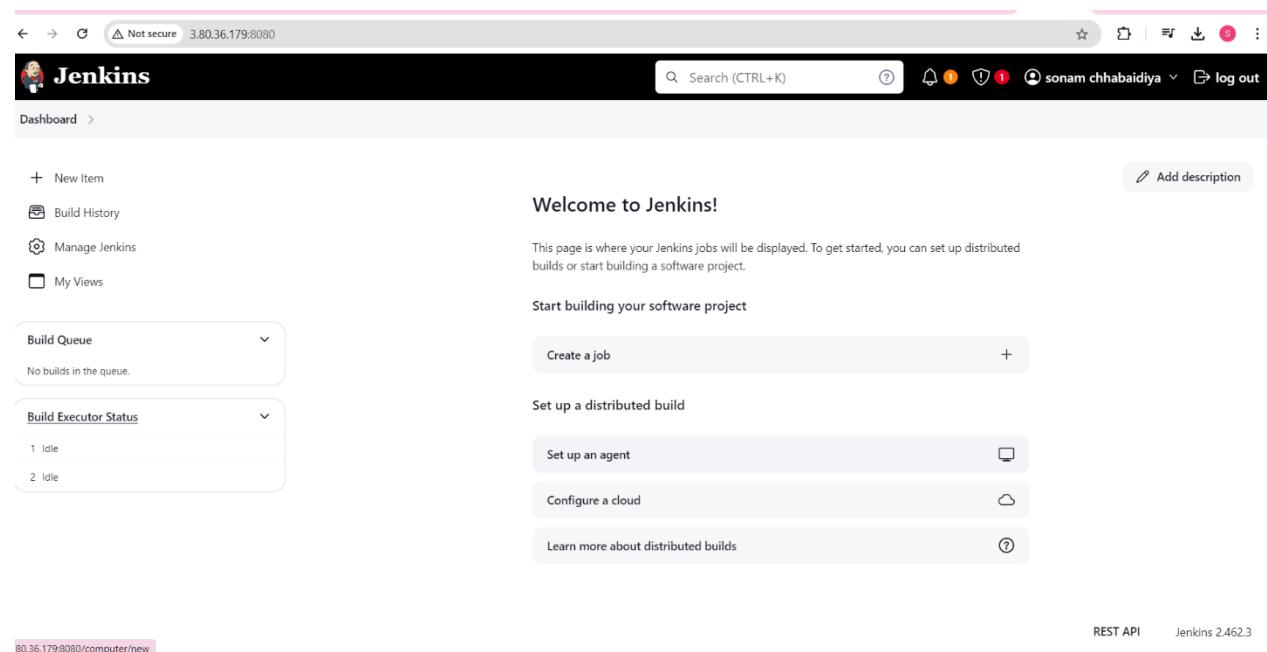
Your Jenkins setup is complete.

Start using Jenkins

Jenkins 2.462.3

Step 11:

After proper initial configuration you will be redirected to this page



Dashboard >

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

New Item

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

REST API Jenkins 2.462.3

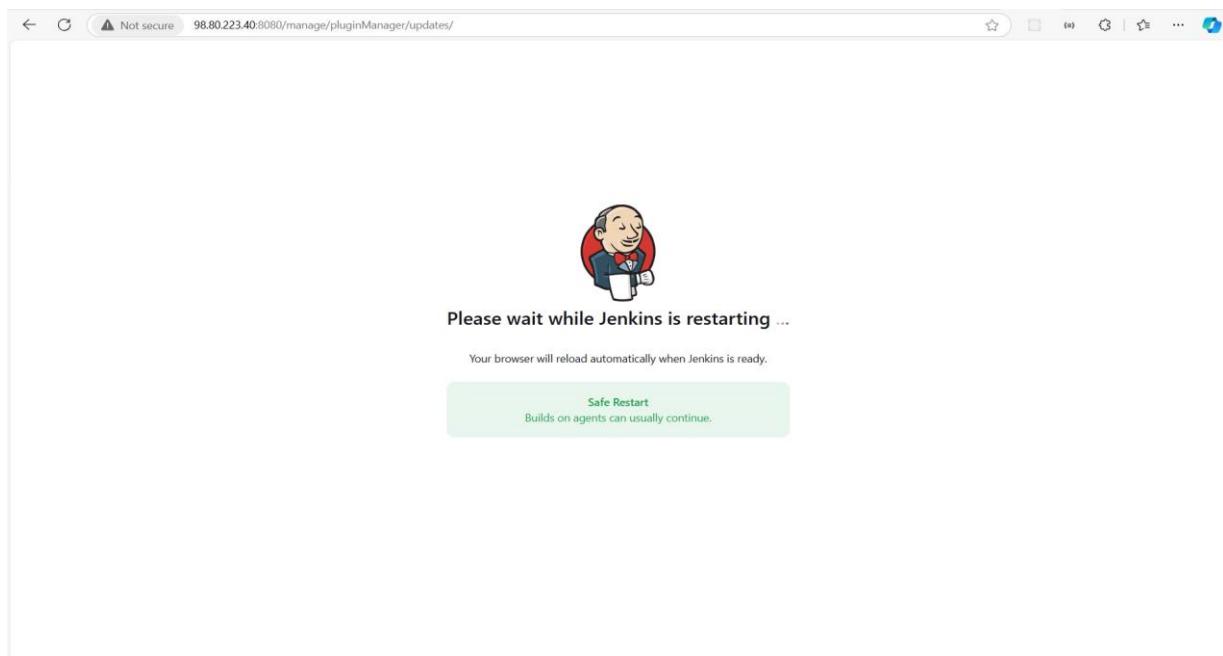
80.36.179:8080/computer/new

Step 12:

Install more plugins which will be required for this experiment.

- a. SonarQube Scanner
- b. Pipeline: Stage View

The screenshot shows the Jenkins plugin manager interface. The search bar at the top contains the text 'sonar'. Below the search bar, there is a list of available plugins. The first plugin listed is 'SonarQube Scanner 2.17.2', which is categorized under 'External Site/Tool Integrations' and 'Build Reports'. Its description states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.' The second plugin listed is 'Sonar Quality Gates 328.vf4369b_da_d3c2', which is categorized under 'Library plugins (for use by other plugins)', 'analysis', and 'Other Post-Build Actions'. Both plugins have an 'Install' button next to them. On the left sidebar, there are links for 'Updates', 'Available plugins' (which is selected), 'Installed plugins', 'Advanced settings', and 'Download progress'.



. Step 13:

Go to Manage Jenkins > Tools. Scroll down to SonarQube Scanner installations and add the SonarQube Scanner and then click on the save button.

SonarQube Scanner installations

Add SonarQube Scanner

SonarQube Scanner

Name: sonarqube

Install automatically:

Install from Maven Central

Version: SonarQube Scanner 6.2.1.4610

Add Installer

Add SonarQube Scanner

Save Apply

Sonarqube installation

Reference video: <https://www.youtube.com/watch?v=E5hMOGeBT-o&t=38s>

Step 1:

Click on the SonarQubeServer and click on connect.

EC2 > Instances > i-053beecb54b82ce6f > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-053beecb54b82ce6f (SonarQubeServer) using any of these options

EC2 Instance Connect Session Manager **SSH client** EC2 serial console

Instance ID: [i-053beecb54b82ce6f \(SonarQubeServer\)](#)

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is `SCsonarqube1.pem`.
- Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "SCsonarqube1.pem"`
- Connect to your instance using its Public DNS:
`ec2-3-86-103-58.compute-1.amazonaws.com`

Example:
`ssh -i "SCsonarqube1.pem" ec2-user@ec2-3-86-103-58.compute-1.amazonaws.com`

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Step 2:

Open Git Bash and go to the directory which has the Key downloaded. If you don't have the key downloaded, create akey pair and download the .pem file for the key.

Since, I have the key downloaded in Downloads directory, I used the following commands:

```
cd Download
```

```
dir SCsonarqube1.pem*
```

```
ssh -i "SCjsonarqube1.pem" ec2-user@ec2-98-80-223-40.compute-1.amazonaws.com
```

```
PC@ASUS MINGW64 ~/Downloads (master)
$ ssh -i "SCsonarqube1.pem" ec2-user@ec2-98-80-223-40.compute-1.amazonaws.com
The authenticity of host 'ec2-98-80-223-40.compute-1.amazonaws.com (98.82.4.97)' can't be established.
ED25519 key fingerprint is SHA256:mgKtPzi1+a00Le9b4w3K8HnpMxoI6PArI+G7mbQqrjI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-98-80-223-40.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      #_
  ~\_ #####_          Amazon Linux 2
  ~\_ \#####\_
  ~\_ \|##|          AL2 End of Life is 2025-06-30.
  ~\_ \|/_           A newer version of Amazon Linux is available!
  ~\_ \|/_/          Amazon Linux 2023, GA and supported until 2028-03-15.
  ~\_ \|/_/          https://aws.amazon.com/linux/amazon-linux-2023/
```

Step 3:

Now, in order to install java run the following

commands:

```
sudo su
```

```
sudo yum install java-17-amazon-corretto-headless
```

```
sudo yum install java-17-amazon-corretto
```

```
sudo dnf install java-17-amazon-corretto-devel
```

```
[ec2-user@ip-172-31-87-220 ~]$ sudo su
[root@ip-172-31-87-220 ec2-user]# sudo yum install java-17-amazon-corretto-headless
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package java-17-amazon-corretto-headless.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Processing Dependency: jpackage-utils for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: fontconfig for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-serif-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-mono-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: alsa-lib for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: log4j-cve-2021-44228-cve-mitigations for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Running transaction check
--> Package alsa-lib.x86_64 0:1.1.4.1-2.amzn2 will be installed
--> Package dejavu-sans-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Processing Dependency: dejavu-fonts-common = 2.33-6.amzn2 for package: dejavu-sans-fonts-2.33-6.amzn2.noarch
--> Package dejavu-sans-mono-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Package dejavu-serif-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Package fontconfig.x86_64 0:2.13.0-4.3.amzn2 will be installed
--> Processing Dependency: fontpackages-filesystem for package: fontconfig-2.13.0-4.3.amzn2.x86_64
--> Package javapackages-tools.noarch 0:3.4.1-11.amzn2 will be installed
--> Processing Dependency: python-javapackages = 3.4.1-11.amzn2 for package: javapackages-tools-3.4.1-11.amzn2.noarch
--> Processing Dependency: libxml2 for package: javapackages-tools-3.4.1-11.amzn2.noarch
--> Package log4j-cve-2021-44228-hotpatch.noarch 0:1.3-7.amzn2 will be installed
--> Running transaction check
--> Package dejavu-fonts-common.noarch 0:2.33-6.amzn2 will be installed
--> Package fontpackages-filesystem.noarch 0:1.44-8.amzn2 will be installed
--> Package libxml2.x86_64 0:1.1.28-6.amzn2 will be installed
--> Package python-javapackages.noarch 0:3.4.1-11.amzn2 will be installed
```

```

Complete!
[root@ip-172-31-87-220 ec2-user]# sudo yum install java-17-amazon-corretto
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package java-17-amazon-corretto.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Processing Dependency: libX11 for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXi for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXinerama for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXt for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXrender for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXrandr for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXtst for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: giflib for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Running transaction check
---> Package giflib.x86_64 0:4.1.6-9.amzn2.0.2 will be installed
--> Processing Dependency: libICE.so.6()(64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
--> Processing Dependency: libSM.so.6()(64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
---> Package libX11.x86_64 0:1.6.7-3.amzn2.0.5 will be installed
--> Processing Dependency: libX11-common >= 1.6.7-3.amzn2.0.5 for package: libX11-1.6.7-3.amzn2.0.5.x86_64
--> Processing Dependency: libxcb.so.1()(64bit) for package: libX11-1.6.7-3.amzn2.0.5.x86_64
---> Package libxi.x86_64 0:1.7.9-1.amzn2.0.2 will be installed
--> Processing Dependency: libXext.so.6()(64bit) for package: libxi-1.7.9-1.amzn2.0.2.x86_64
---> Package libXinerama.x86_64 0:1.1.3-2.1.amzn2.0.2 will be installed
---> Package libXrandr.x86_64 0:1.5.1-2.amzn2.0.3 will be installed
---> Package libXrender.x86_64 0:0.9.10-1.amzn2.0.2 will be installed
---> Package libXt.x86_64 0:1.1.5-3.amzn2.0.2 will be installed
---> Package libXtst.x86_64 0:1.2.3-1.amzn2.0.2 will be installed
--> Running transaction check
---> Package libICE.x86_64 0:1.0.9-9.amzn2.0.2 will be installed

```

```

[root@ip-172-31-87-220 ec2-user]# sudo dnf install java-17-amazon-corretto-devel
sudo: dnf: command not found
[root@ip-172-31-87-220 ec2-user]# sudo yum install -y java-17-amazon-corretto-devel
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package java-17-amazon-corretto-devel.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

=====
Package           Arch      Version            Repository      Size
=====
Installing:
java-17-amazon-corretto-devel   x86_64    1:17.0.12+7-1.amzn2.1          amzn2-core       146 k
Transaction Summary
=====
Install 1 Package

Total download size: 146 k
Installed size: 940 k
Downloading packages:
java-17-amazon-corretto-devel-17.0.12+7-1.amzn2.1.x86_64.rpm | 146 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:java-17-amazon-corretto-devel-17.0.12+7-1.amzn2.1.x86_64 1/1
  Verifying  : 1:java-17-amazon-corretto-devel-17.0.12+7-1.amzn2.1.x86_64 1/1
Installed:
  java-17-amazon-corretto-devel.x86_64 1:17.0.12+7-1.amzn2.1

Complete!
[root@ip-172-31-87-220 ec2-user]# sudo wget -O /etc/yum.repos.d/sonar.repo http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo
--2024-10-23 06:28:19-- http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://psychz.dl.sourceforge.net/project/sonar-pkg/rpm/sonar.repo?viasf=1 [following]
--2024-10-23 06:28:20-- http://psychz.dl.sourceforge.net/project/sonar-pkg/rpm/sonar.repo?viasf=1
Resolving psychz.dl.sourceforge.net (psychz.dl.sourceforge.net)... 208.87.241.191
Connecting to psychz.dl.sourceforge.net (psychz.dl.sourceforge.net)|208.87.241.191|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 93 [application/octet-stream]
Saving to: '/etc/yum.repos.d/sonar.repo'


```

Step 4:

Now, run the following command to install sonarqube:

```

sudo wget -O /etc/yum.repos.d/sonar.repo http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo
sudo yum install sonar -y

```

```

Complete!
[root@ip-172-31-87-220 ec2-user]# sudo wget -O /etc/yum.repos.d/sonar.repo http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo
--2024-10-23 06:28:19-- http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 204.68.111.105
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|204.68.111.105|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://psychz.dl.sourceforge.net/project/sonar-pkg/rpm/sonar.repo?viasf=1 [following]
--2024-10-23 06:28:20-- http://psychz.dl.sourceforge.net/project/sonar-pkg/rpm/sonar.repo?viasf=1
Resolving psychz.dl.sourceforge.net (psychz.dl.sourceforge.net)... 208.87.241.191
Connecting to psychz.dl.sourceforge.net (psychz.dl.sourceforge.net)|208.87.241.191|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 93 [application/octet-stream]
Saving to: '/etc/yum.repos.d/sonar.repo'

100%[=====] 93 --.-K/s  in

2024-10-23 06:28:20 (14.9 MB/s) - '/etc/yum.repos.d/sonar.repo' saved [93/93]

[root@ip-172-31-87-220 ec2-user]# sudo yum install sonar -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
sonar
sonar/primary_db
Resolving Dependencies
--> Running transaction check
---> Package sonar.noarch 0:7.1-1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

=====
Package           Arch      Version       Repository
=====
Installing:      sonar      noarch    7.1-1          sonar

Transaction Summary
Install 1 Package

total download size: 142 M
installed size: 169 M
Downloading packages:
sonar-7.1-1.noarch.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : sonar-7.1-1.noarch
  Verifying  : sonar-7.1-1.noarch

| 142 MB 00:00

```

Step 5:

Now, run the following commands to install the sonarqube

sudo su

cd

/opt/ll

wget <https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-10.7.0.96327.zip> unzip sonarqube-10.7.0.96327.zip

||

cd sonarqube-10.7.0.96327||

cd

conf||

cat

sonar.properties

cd ..

cd bin

cd linux-x86-

64||

./sonar.sh start

sudo adduser

sonar

sudo passwd sonar

sudo chown -R sonar:sonar /opt/sonarqube-10.7.0.96327

su – sonar

cd /opt/sonarqube-10.7.0.96327/bin/linux-x86-64/

./sonar.sh start

./sonar.sh status

```

[Complete!
[root@ip-172-31-87-220 ec2-user]# sudo su
[root@ip-172-31-87-220 ec2-user]# cd /opt
[root@ip-172-31-87-220 opt]# ll
total 741384
drwxr-xr-x  2 root root    33 Jan 24 2023 aws
drwxr-xr-x  2 root root    6 Aug 16 2018 rh
drwxr-xr-x 11 sonar sonar   141 Oct 23 06:28 sonar
drwxr-xr-x 12 sonar sonar   184 Sep 27 15:16 sonarqube-10.7.0.96327
-rw-r--r--  1 root root 759175712 Sep 27 15:52 sonarqube-10.7.0.96327.zip
[root@ip-172-31-87-220 opt]# wget https://binaries.sonarsource.com/distribution/sonarqube/sonarqube-10.7.0.96327.zip
--2024-10-23 06:29:38- https://binaries.sonarsource.com/distribution/sonarqube/sonarqube-10.7.0.96327.zip
Resolving binaries.sonarsource.com (binaries.sonarsource.com)... 99.84.191.23, 99.84.191.75, 99.84.191.71, ...
Connecting to binaries.sonarsource.com (binaries.sonarsource.com) [99.84.191.23]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 759175712 (724M) [binary/octet-stream]
Saving to: 'sonarqube-10.7.0.96327.zip' [=====] 759,175,712 83.4MB/s  in 8.6s

2024-10-23 06:29:46 (84.4 MB/s) - 'sonarqube-10.7.0.96327.zip' saved [759175712/759175712]

[root@ip-172-31-87-220 opt]# unzip sonarqube-10.7.0.96327.zip
Archive: sonarqube-10.7.0.96327.zip
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_linux_hotspot_17.0.11.9.tar.gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: ll
error: invalid response [ll]
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_linux_hotspot_17.0.11.9.tar.gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_linux_hotspot_17.0.11.9.tar.gz
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_aarch64_linux_hotspot_17.0.11.9.tar.gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_aarch64_linux_hotspot_17.0.11.9.tar.gz
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_alpine-linux_hotspot_17.0.11.9.tar.gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
error: invalid response [ENTER]
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_windows_hotspot_17.0.11.9.zip? [y]es, [n]o, [A]ll, [N]one, [r]ename: o
error: invalid response [ENTER]
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_windows_hotspot_17.0.11.9.zip? [y]es, [n]o, [A]ll, [N]one, [r]ename: es
error: invalid response [es]
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_windows_hotspot_17.0.11.9.zip? [y]es, [n]o, [A]ll, [N]one, [r]ename: Read from remote host ec2-98-82-4-97.compute-1.amazonaws.com: Connection reset by peer
Connection to ec2-98-82-4-97.compute-1.amazonaws.com closed.
client_loop: send disconnect: connection reset by peer

```

```

[root@ip-172-31-87-220 opt]# unzip sonarqube-10.7.0.96327.zip
Archive: sonarqube-10.7.0.96327.zip
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_linux_hotspot_17.0.11.9.tar.gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_aarch64_linux_hotspot_17.0.11.9.tar.gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: o
error: invalid response [o]
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_aarch64_linux_hotspot_17.0.11.9.tar.gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: no
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_alpine-linux_hotspot_17.0.11.9.tar.gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
error: invalid response [n]o
replace sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_linux_hotspot_17.0.11.9.tar.gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  inflating: sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_alpine-linux_hotspot_17.0.11.9.tar.gz
  inflating: sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_windows_hotspot_17.0.11.9.zip
  inflating: sonarqube-10.7.0.96327/jres/OpenJDK17U-jre_x64_mac_hotspot_17.0.11.9.tar.gz
  inflating: sonarqube-10.7.0.96327/dependency-license.json
  inflating: sonarqube-10.7.0.96327/COPYING
  inflating: sonarqube-10.7.0.96327/bin/windows-x86-64/SonarService.bat
  inflating: sonarqube-10.7.0.96327/bin/windows-x86-64/lib/SonarServiceWrapper.exe
  inflating: sonarqube-10.7.0.96327/bin/windows-x86-64/lib/find_java.bat
  inflating: sonarqube-10.7.0.96327/bin/winsw-license/LICENSE.txt
  inflating: sonarqube-10.7.0.96327/data/README.txt
  inflating: sonarqube-10.7.0.96327/extensions/jdbc-driver/oracle/README.txt
  inflating: sonarqube-10.7.0.96327/extensions/plugins/README.txt
  inflating: sonarqube-10.7.0.96327/logs/README.txt
  inflating: sonarqube-10.7.0.96327/temp/README.txt
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-preallocate-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-x-content-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-124-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-cli-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-vec-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-native-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-core-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-logging-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-secure-sm-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-geo-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-plugin-analysis-api-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-plugin-api-8.14.1.jar
  inflating: sonarqube-10.7.0.96327/elasticsearch/lib/elasticsearch-grok-8.14.1.jar

```

```

[root@ip-172-31-87-220 opt]# ll
total 2224152
drwxr-xr-x  4 root root    33 Jan 24 2023 aws
drwxr-xr-x  2 root root    6 Aug 16 2018 rh
drwxr-xr-x 11 sonar sonar   141 Oct 23 06:28 sonar
drwxr-xr-x 12 sonar sonar   184 Oct 23 06:37 sonarqube-10.7.0.96327
-rw-r--r--  1 root root 759175712 Sep 27 15:52 sonarqube-10.7.0.96327.zip.1
-rw-r--r--  1 root root 759175712 Sep 27 15:52 sonarqube-10.7.0.96327.zip.2
[root@ip-172-31-87-220 opt]# cd sonarqube-10.7.0.96327
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]# ll
total 88
drwxr-xr-x  6 sonar sonar   117 Oct 23 06:37 bin
drwxr-xr-x  2 sonar sonar   30 Oct 23 06:37 conf
-rw-r--r--  1 root root 7651 Sep 27 15:13 COPYING
drwxr-xr-x  2 sonar sonar   24 Oct 23 06:37 data
-rw-r--r--  1 root root 73376 Sep 27 15:14 dependency-license.json
drwxr-xr-x  7 sonar sonar   132 Oct 23 06:37 elasticsearch
drwxr-xr-x  4 sonar sonar   40 Sep 27 15:13 extensions
drwxr-xr-x  2 sonar sonar  4096 Oct 23 06:37 jres
drwxr-xr-x  5 sonar sonar   134 Oct 23 06:37 lib
drwxr-xr-x  2 sonar sonar   24 Oct 23 06:37 logs
drwxr-xr-x  2 sonar sonar   24 Oct 23 06:37 temp
drwxr-xr-x  6 sonar sonar  4096 Oct 23 06:37 web
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]# ll
total 88
drwxr-xr-x  6 sonar sonar   117 Oct 23 06:37 bin
drwxr-xr-x  2 sonar sonar   30 Oct 23 06:37 conf
-rw-r--r--  1 root root 7651 Sep 27 15:13 COPYING
drwxr-xr-x  2 sonar sonar   24 Oct 23 06:37 data
-rw-r--r--  1 root root 73376 Sep 27 15:14 dependency-license.json
drwxr-xr-x  7 sonar sonar   132 Oct 23 06:37 elasticsearch
drwxr-xr-x  4 sonar sonar   40 Sep 27 15:13 extensions
drwxr-xr-x  2 sonar sonar  4096 Oct 23 06:37 jres
drwxr-xr-x  5 sonar sonar   134 Oct 23 06:37 lib
drwxr-xr-x  2 sonar sonar   24 Oct 23 06:37 logs
drwxr-xr-x  2 sonar sonar   24 Oct 23 06:37 temp
drwxr-xr-x  6 sonar sonar  4096 Oct 23 06:37 web
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]# cd conf
[root@ip-172-31-87-220 conf]# ll
total 24
-rw-r--r-- 1 root root 21761 Sep 27 15:13 sonar.properties

```

```

# Delay in seconds between processing of notification queue. Default is 60 seconds.
sonar.notifications.delay=60

# Paths to persistent data files (embedded database and search index) and temporary files.
# Can be absolute or relative to installation directory.
# Defaults are respectively <installation home>/data and <installation home>/temp
sonar.path.data=data
sonar.path.temp=temp

# Telemetry - Share anonymous SonarQube statistics
# By sharing anonymous SonarQube statistics, you help us understand how SonarQube is used so we can improve the product to work even better for you.
# We don't collect source code or IP addresses. And we don't share the data with anyone else.
sonar.telemetry.enable=true
[root@ip-172-31-87-220 conf]\# cd /opt/sonarqube-10.7.0.96327/bin/linux-x86-64
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]\# ls
bin          conf          logs          sonar        temp
bin          conf          logs          sonar        temp
total 88
drwxr-xr-x  6 sonar sonar  117 Oct 23 06:37 bin
drwxr-xr-x  2 sonar sonar  30 Oct 23 06:37 conf
-rw-r--r--  1 root  root  7651 Sep 27 15:14 COPYING
drwxr-xr-x  3 sonar sonar  108 Oct 23 06:37 logs
drwxr-xr-x  1 root  root  73376 Sep 27 15:14 dependency-license.json
drwxr-xr-x  7 sonar sonar 132 Oct 23 06:37 elasticsearch
drwxr-xr-x  4 sonar sonar  40 Sep 27 15:13 extensions
drwxr-xr-x  2 sonar sonar 4096 Oct 23 06:37 jres
drwxr-xr-x  5 sonar sonar 134 Oct 23 06:37 lib
drwxr-xr-x  2 sonar sonar  24 Oct 23 06:37 logs
drwxr-xr-x  6 sonar sonar 4096 Oct 23 06:37 web
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]\# ./sonar.sh start
bash: ./sonar.sh: No such file or directory
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]\# sudo adduser sonar
adduser: user "sonar" already exists
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]\# sudo passwd sonar
Changing password for user sonar.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]\# sudo chown -R sonar:sonar /opt/sonarqube-10.7.0.96327
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]\# su - sonar
su: user - does not exist
[root@ip-172-31-87-220 sonarqube-10.7.0.96327]\# cd /opt/sonarqube-10.7.0.96327/bin/linux-x86-64/
[root@ip-172-31-87-220 linux-x86-64]\# ./sonar.sh start
/bin/java
Starting SonarQube...
started SonarQube...

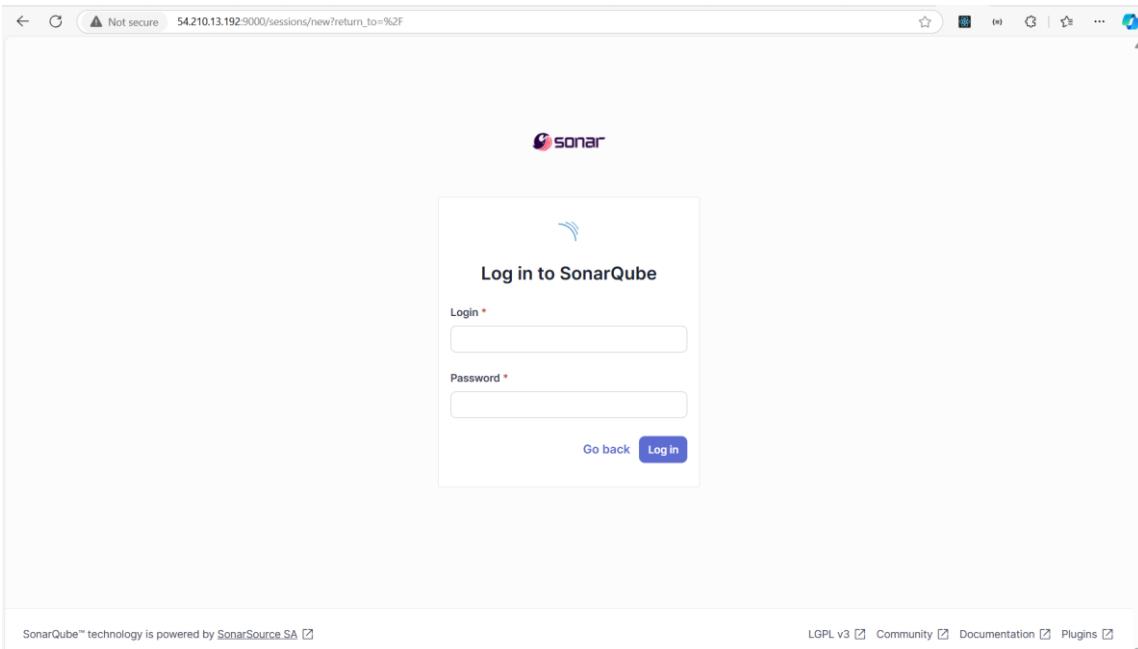
```

Step 6:

Now, go to EC2 dashboard and select SonarQubeServer and copy its public address and visit <http://<public-address>:9000>

Step 7:

You will be redirected to this page on successful installation of SonarQube and visiting the public address url with port9000. Login the username=admin and password=admin.



Step 8:

Now, set up the initial configurations by setting up new password.password=Sonamsonam@2004

Update your password

⚠ This account should not use the default password.

Enter a new password
All fields marked with * are required

Old Password *
.....

Password *
.....

Confirm Password *
.....

Update

Step 9:

Now, click on the Create a local project link and name the project Hello-World and choose use the global setting

The screenshot shows the 'Create a local project' page of SonarQube. At the top, there's a heading 'How do you want to create your project?'. Below it, a note says: '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.' A sub-note below that says: 'First, you need to set up a DevOps platform configuration.' There are six 'Import' buttons arranged in two rows of three: 'Import from Azure DevOps', 'Import from Bitbucket Cloud', 'Import from Bitbucket Server', 'Import from GitHub', 'Import from GitLab', and 'Import from Local'. Each button has a 'Setup' link next to it. At the bottom left, there's a note: 'Are you just testing or have an advanced use-case? Create a local project.' Below that is a 'Create a local project' button. At the very bottom, there's a note: '⚠ Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.' The footer of the page includes the SonarQube logo, version information (Community Edition v10.7 (96327) ACTIVE), and links to Community, Documentation, Plugins, and Web API.

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

The name of your project's default branch [Learn More](#)

[Cancel](#) [Next](#)

⚠️ Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA [Community Edition v10.7 \(96327\) ACTIVE](#) [LGPL v3](#) [Community](#) [Documentation](#) [Plugins](#) [Web API](#)

Step 10:

Click on Administration > Security > Users. And then give a token name and click on the generate token button and copy the token number and save it somewhere. For now, I have saved it in notepad

Not secure | 54.210.13.192:9000/admin/settings

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

Administration

Configuration Security Projects System Marketplace

General

- Users
- Groups
- Global Permissions
- Permission Templates

Duplications

Cross project duplication detection

DEPRECATED - By default, SonarQube detects duplications at project level. This means that a block duplicated on two different projects won't be reported. Setting this parameter to "true" allows to detect duplicates across projects. Note that activating this property will significantly increase each SonarQube analysis time, and therefore badly impact the performances of report processing as more and more projects are getting involved in this cross project duplication mechanism.

Key: sonar.cpd.cross_project

54.210.13.192:9000/admin/users

Tokens of Administrator

Generate Tokens

Name	Expires in
AdvDevops-CaseStudy	30 days

Generate

Name	Type	Project	Last use	Created	Expiration

Close

Tokens of Administrator

Generate Tokens

Name	Expires in
Enter Token Name	30 days

Generate

New token "AdvDevops-CaseStudy" has been created. Make sure you copy it now, you won't be able to see it again!

squ_8a382e533eb6be59b3c627ce8b99250a8ee0e5ce

Close

Githu

b

Step

1:

Make a repository and upload your files in the repository.

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Sonampython Public

main 1 Branch Tags

sonamcc Initial commit 849a1c6 · 2 hours ago 1 Commit

README.md Initial commit 2 hours ago

README

Sonampython

hello-world python

Readme Activity 0 stars 1 watching 0 forks

No releases published Create a new release

Packages

Pipeline

Step

1:

Open the git bash for Jenkins and run the following commands in the terminal to install git.
sudo yum install git
git --version

```
[ec2-user@ip-172-31-25-217 ~]$ sudo yum install git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl-Git = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Git) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
--> Package git-core-doc.noarch 0:2.40.1-1.amzn2.0.3 will be installed
--> Package perl-Git.noarch 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: perl(Git) for package: perl-Git-2.40.1-1.amzn2.0.3.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

=====
Package           Arch      Version            Repository
=====
Installing:
git              x86_64   2.40.1-1.amzn2.0.3          amzn2-core
Installing for dependencies:
git-core          x86_64   2.40.1-1.amzn2.0.3          amzn2-core
git-core-doc     noarch   2.40.1-1.amzn2.0.3          amzn2-core
perl-Error        noarch   1:0.17020-2.amzn2          amzn2-core
perl-Git          noarch   2.40.1-1.amzn2.0.3          amzn2-core
perl-TermReadKey x86_64   2.30-20.amzn2.0.2         amzn2-core

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

```
Complete!
[ec2-user@ip-172-31-25-217 ~]$ git -version
git: '-version' is not a git command. See 'git --help'.
[ec2-user@ip-172-31-25-217 ~]$ git --version
git version 2.40.1
[ec2-user@ip-172-31-25-217 ~]$ Read from remote host ec2-3-80-36-179.compute-1.amazonaws.com: Connection reset by peer
Connection to ec2-3-80-36-179.compute-1.amazonaws.com closed.
client_loop: send disconnect: Connection reset by peer
```

Step 2:

Go to Manage Jenkins > System. Scroll down to SonarQube Servers section and name it as SonarQube Server and copy the <http://<public-address-of-sonarqube>:9000>
Also, copy the token as secret here in secret text.

Not secure 98.80.223.40:8080/manage/configure

Dashboard > Manage Jenkins > System >

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables

SonarQube installations

List of SonarQube installations

Name: SonarQube Server

Server URL: Default is http://localhost:9000
http://54.210.13.192:9000

Server authentication token:
SonarQube authentication token. Mandatory when anonymous access is disabled.
- none -
+ Add
Advanced

Save Apply

Not secure 98.80.223.40:8080/manage/configure

Dashboard > Manage Jenkins > System >

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: Secret text

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

Secret:

ID:

Description:

Save Apply

SonarQube installations

List of SonarQube installations

Name: SonarQube Server

Server URL: Default is http://localhost:9000
http://98.82.4.97:9000/

Server authentication token:
SonarQube authentication token. Mandatory when anonymous access is disabled.
- none -
+ Add
Advanced

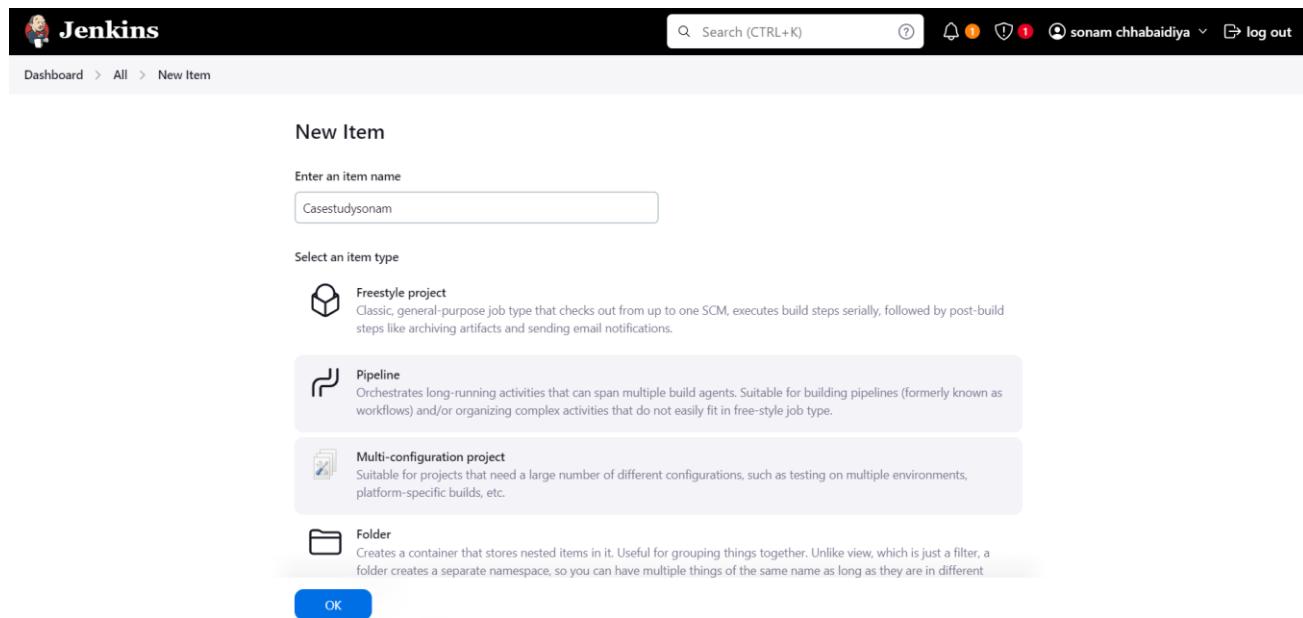
Save Apply

Step 3:

Go to Manage Jenkins > Credentials. Copy the id and you will need to paste in the Pipeline Script later.

Step 4:

Create a new pipeline and name it



New Item

Enter an item name
Casestudysonam

Select an item type

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different

OK

Step 5:

Now select on the Git project and paste your GitHub url.

Step 6:

Now, write the following Pipeline Script.pipeline {

```
agent
any
stages {
    stage('Clone
        Repository') {
            git branch: 'main', url: 'https://github.com/Anuprita2022-26/helloworld_python.git'
        }
    }
    stage('SonarQube
        Analysis') {
            environment {
```

```

        scannerHome = tool 'SonarQubeScanner' // Ensure SonarQube Scanner is
        installed
    }

    steps {
        withSonarQubeEnv('SonarQube') { // Name of SonarQube server configured in
        Jenkins
            withCredentials([string(credentialsId: '6e0ad648-6931-48d0-a2eb-938a55db6234',
            variable:'SONAR_TOKEN')]) {
                sh "${scannerHome}/bin/sonar-scanner -Dsonar.projectKey=AdvDevops -
                CaseStudy-Dsonar.sources=. -Dsonar.login=$SONAR_TOKEN"
            }
        }
    }

}

}

post {
    always {
        echo 'Pipeline completed'
    }
}
}

```

Step 7:

Build run the pipeline. It gives success. Also, check the console.

	Clone Repository	SonarQube Analysis	Declarative: Post Actions
Average stage times: (Average full run time: ~36s)	996ms	4s	91ms
#7 Oct 23 13:37 No Changes	1s	28s	106ms
#6 Oct 23 13:31 No Changes	348ms	205ms failed	77ms
#5 Oct 23 13:20 No Changes	331ms	197ms failed	89ms
#4 Oct 23 13:18 No Changes	326ms	1s failed	76ms

The screenshot shows the Jenkins dashboard. At the top, there's a search bar with 'Search (CTRL+K)' and a user icon for 'sonam chhabaidiya'. Below the header, a navigation bar includes links for 'Dashboard', 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', and 'Restart Safely'. A central table displays build information for 'Casestudysonam': Last Success was 1 min 7 sec ago (#7), Last Failure was 7 min 28 sec ago (#6), and Last Duration was 36 sec. Icons for status (green checkmark), workspace (cloud), and last build (yellow triangle) are shown. Below the table, sections for 'Build Queue' (empty) and 'Build Executor Status' (1 Idle, 2 Idle) are visible. At the bottom right, there are links for 'REST API' and 'Jenkins 2.462'.

SonarQube Analysis

and Results Step 1:

Visit back to the <http://<public-address-of-scanarqube>:9000>

Now, go to projects section and you can see the analysis of the python project.

The screenshot shows the SonarQube 'Projects' page. The URL in the browser is '98.82.4.97:9000/projects'. The page has a navigation bar with 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', 'More', and a search bar. On the left, there are filters for 'My Favorites' (All), 'Quality Gate' (Passed: 1, Failed: 0), 'Security' (Info issues: 1, Minor: 0, Major: 0, Critical: 0, Blocker: 0), and 'Reliability'. The main area shows a single project: 'AdvDevOps-CaseStudy' (PUBLIC). It indicates 'Last analysis: 1 minute ago' and notes 'The main branch of this project is empty.' A message at the bottom states 'Embedded database should be used for evaluation purposes only' and provides a warning about its limitations. The status of the project is 'Passed'.

Guidelines for User:

- Check the Security group.
Ensure your AWS secret tokens are accessible