

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>

AWS Documentation AWS Command Line Interface User Guide for Version 2

Contact Us English Return to the Console

Feedback Preferences

On this page

AWS CLI install and update instructions

Troubleshooting AWS CLI install and uninstall errors
Next steps

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

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.

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

Now, click on install <https://awscli.amazonaws.com/AWSCLIV2.msi>

Complete the installation process for AWSCLIV2

Downloads

Search Downloads

New + Details

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...

Downloads (105 items)

Select a single file to get more information and share your cloud content.

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 cedentials

The screenshot shows the AWS Academy Learner Lab interface. On the left, there's 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 has buttons for 'Start Lab', 'End Lab', 'AWS Details', 'Readme', and 'Reset'. The central area has tabs for Home, Modules, Discussions, Grades (with a red notification badge), and Lucid (Whiteboard). A terminal window shows a command prompt: 'eee_W_3429984@runweb141049:~\$ []'. To the right is a 'Learner Lab' panel with a dropdown for 'EN-US'. Below it is a 'Environment Overview' section with links to various AWS services and restrictions. At the bottom, it says 'Instructions last updated: 2024-08-06' and 'Environment Overview'.

This screenshot shows the same interface as above, but the 'AWS Details' section is expanded. It includes a 'Cloud Access' section with a 'Close' button, a 'Cloud Labs' section showing session details (started at 2024-10-18T10:50:58-0700, ended at 2024-10-18T14:50:58-0700, accumulated lab time: 1 day 06:28:00 (1828 minutes)), and a 'No running instance' message. It also includes download buttons for 'SSH key', 'Download PEM', 'Download PPK', 'AWS SSO', and 'Download URL'. At the bottom, there's a table showing AWSAccountID (856746069793) and Region (us-east-1).

The screenshot shows the AWS Academy Learner Lab interface. On the left, there's a sidebar with icons for Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area has a breadcrumb navigation: ALLv2EN-US... > Modules > AWS Academy... > Launch AWS Academy Learner Lab. The top right shows 'Used \$1.6 of \$50' and a timer '00:42'. Below the breadcrumb, there are tabs for Home, Modules (selected), Discussions, Grades, and Lucid (Whiteboard). A notification icon with the number '1' is visible next to the Grades tab. The central pane shows a terminal window with the command 'eee_l_3429984@runweb141049:~\$'. To the right is a 'Cloud Access' panel with a 'Close' button. It contains instructions: 'Copy and paste the following into ~/.aws/credentials'. Below this is a large text block of AWS CLI configuration parameters.

```
[default]
aws_access_key_id=ASTAA06QLR4QWTUDTJ4D
aws_secret_access_key=c0HyVzvWpgGYWLFIeFlei
tKgh5a1+gfyic1h5bC0
aws_session_token=I0q0lJ33pZ21uxX2VjEO
L/////////wEaCXVzLXd1c3QtMiJGMGQCIC6ot5d1X
YaUUDdgKzyS2jQlyG6+fd0xRkZvGU/gePFAiBzREYq
4yVkoSgrnRML/NtikKeoyS0FZkKoqddd4t8SiqyAgh
LEAEaDDg1Nj0NjAz0Tc5MyIMGVa9Rykq32BXqFPKKo
8C8108faEKk1Uy48FcBvU9V9IkCp13fNDzWky1s2
6C+k1cgeBLW19ttzVnX356AwkGXtgLwsjcr1lbyQP
amIYQreT355QY9Pmkid0aZm0501ez2Yss4CJFQP2nig
yi8mzAHWlphwCOAXRofJ3ZBz2AeK3qm3xNUvh+hjJNR
yF3PTCNjCf1FNyxwCpQ3Wk58g8/UomV907Yi9ucgH6
o7mbCcvQDVsdsebxz1w4NVHMyQgJVlqC9D3NKK4PbBm
MShi duilEc4euEljYkFyY5penPzt5vhvn27Ghn7DC
FPJ1GL0WjunYXD+Lw+O1Zf5v15CQozvCmmKoaJukG
dIM6DmQH8kaGyv+/VygkzCcMq4BqjeAZItoyQim5+CF
B1XhMYx7LUnbwMFDKzTA4AgjC40BqyF12KO1BC9gCh
E37rFPxle1zPzNBMEIVZJ2E41wkAzoxwX35yJ2U0c7d
tYjXWz0IE50U3qKIsd2MvQhrtksFy7ng2dU8IrEd8g
ZhMXzenEKYTX3n5/WABOXLpeIeagP3jf3Ih9glmJB8
```

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-05f408238af346b4f" # Amazon Linux 2 AMI
```

```
    instance_type = "t2.micro"
```

```
    key_name    = "AMjenkins"
```

```
    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-05f408238af346b4f" # Amazon Linux 2 AMI
    instance_type = "t2.medium"
    key_name     = "AMsonarqube"
    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
}

```

}

```

File Edit Selection View Go Run Terminal Help ⏪ ⏩ AdvDevOps_Practical
EXPLORER ADVDEVOPS_PRactical ...
main.tf x hello.py
main.tf
1 # Specify the AWS provider
2 provider "aws" {
3   region = "us-east-1" # Replace with your preferred region
4 }
5
6 # Jenkins instance
7 resource "aws_instance" "jenkins" {
8   ami      = "ami-05f408238af346b4f" # Amazon Linux 2 AMI
9   instance_type = "t2.micro"
10  key_name     = "AMjenkins"
11  tags = {
12    Name = "JenkinsServer"
13  }
14
15  # User data to install Jenkins with Java 17
16  user_data = <<-EOF
17  #!/bin/bash
18  sudo yum update -y
19  sudo dnf install -y java-17-amazon-corretto-devel # Install Java 17
20  sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
21  sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
22  sudo yum install -y jenkins
23  sudo systemctl start jenkins
24  sudo systemctl enable jenkins
25
26 } # Jenkins instance
27
28 # SonarQube instance
29 resource "aws_instance" "sonarqube" {
30   ami      = "ami-05f408238af346b4f" # Amazon Linux 2 AMI
31   instance_type = "t2.medium"
32   key_name     = "AMsonarqube"
33   tags = {
34     Name = "SonarQubeServer"
35   }
36
37  # User data to install SonarQube manually
38  user_data = <<-EOF
39  #!/bin/bash
40  sudo yum update -y
41  sudo su -
42  cd /opt

```

Step 4:

Now, to ensure and run the aws cli commands in vs code terminal, run the following commands

aws –version
aws configure

(Write the content as mentioned in the figure below)

The screenshot shows the Visual Studio Code interface. In the Explorer sidebar, there is a folder named 'ADVDEVOPS_PRACTICAL' containing files 'hello.py' and 'main.tf'. The 'hello.py' file has the following content:

```
1 print("Hello, World!")
```

In the Terminal tab, the output shows the AWS CLI being configured:

```
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]: ASIA4O6QLR4QWMA6EBFG
AWS Secret Access Key [None]: g101Lg880R5lgoNx0OE8oX3TNhxBcJ5u8fTUHuU
Default region name [None]: us-east-1
Default output format [None]: json
○ PS D:\Sem5_anuprita\AdvDevops_Practical>
```

The status bar at the bottom indicates the terminal is in line 1, column 23, with spaces 4, CRLF, Python 3.12.0 64-bit, Go Live, and Prettier.

Step 5:

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

```
$env:AWS_ACCESS_KEY_ID="ASIA4O6QLR4QWMA6EBFG"
$env:AWS_SECRET_ACCESS_KEY="g101Lg880R5lgoNx0OE8oX3TNhxBcJ5u8fTUHuU"
$env:AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjEO//////////wEaCXVzLXdIc3QtMiJIMEYCIQDFth+aigG/gP+Z6F3r+MqGoz
mnWIMrKNwWhAdlQybeowlhANM0Vv37FoO8JPYclXeF9WdzbBVNXuzCyL3CFTilbFITKrICCFgQARoMODU2NzQ2MDY5Nzk
zlgzfb754SQVtRN0LY6gqjwLRheg15jeclfhu956B4lw1wh7nh8uAbmksRgzTFn0f/XRdNDp8umD5361F3CB0Kw0y3u2iUiappL
GRgmAls6Ipwx91OaD/IziE+J6jqA9werjjE+vPSwJZDjBAI3AOsXEqrHHOxOCMnV8tDADefFFdq4/TVCpj7XrZZbr/bI9b33kA28Y
b02vxwFpXXwCjcuPtkae+kPC5v74lAb1S0gEXudqQv1okS7d+M6hg3qOqrIswTZJ52IZHDZYbeavRrU5qv8W+ky95bOhy1fb3
HKUSGKnpS6Pu2IDqjG97SiAFiB6YCxBzq4UxFJWVreSMILwMPzbL7nq87z3lbRNHbJ3RylWA0TLNBwNNKT92DPyqMP+jzbgG
OpwBBmh89toqaGEkit4lMz7vMppk0w4Q6pgBb/qWs5QtACBlxE6MMqUMBvMquTpP7t3u48g6Z7/ebnkNnf4xtfZ+MgcGY
pwWqkHhiBg3QoX4toPicq83phWThYfOaRop4D/V7h9en8dRRMLuYkEh545h55d+dHbzgop1JpKCLCvcDO8jh3UGzDhSlvEB
5BYjnwrn24CiA3bX4yHa/2o"
```

```

resource "aws_lambda_function" "hello" {
  function_name = "hello"
  runtime      = "python3.8"
  role          = "arn:aws:iam::123456789012:role/lambdaBasicExecutionRole"
  handler       = "app.handler"
  code          = "lambda_function.zip"
}

resource "aws_s3_bucket" "bucket" {
  bucket = "aws-tutorial-bucket"
}

resource "aws_lambda_permission" "lambda_s3" {
  action        = "LambdaInvokeFunction"
  function_name = aws_lambda_function.hello.function_name
  principal    = "s3.amazonaws.com"
  source_arn   = "arn:aws:s3:::aws-tutorial-bucket"
}

```

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
```

```

resource "aws_lambda_function" "hello" {
  function_name = "hello"
  runtime      = "python3.8"
  role          = "arn:aws:iam::123456789012:role/lambdaBasicExecutionRole"
  handler       = "app.handler"
  code          = "lambda_function.zip"
}

resource "aws_s3_bucket" "bucket" {
  bucket = "aws-tutorial-bucket"
}

resource "aws_lambda_permission" "lambda_s3" {
  action        = "LambdaInvokeFunction"
  function_name = aws_lambda_function.hello.function_name
  principal    = "s3.amazonaws.com"
  source_arn   = "arn:aws:s3:::aws-tutorial-bucket"
}

```

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
```

The screenshot shows the VSCode interface with the terminal tab active. The terminal window displays the AWS CLI command to list AMIs and its output, which lists numerous AMI IDs and names. The code editor shows a Terraform configuration file (main.tf) with a script block that uses wget to download SonarQube and sudo to start it.

```
PS D:\Sem6\anuprita\AdvDevops_Practical> 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
+-----+-----+
| ImageId | Name |
+-----+-----+
| ami-007868005aea67c54 | amzn2-ami-hvm-2.0.20230119.1-x86_64-gp2 |
| ami-0241b1d769b029352 | amzn2-ami-hvm-2.0.20240620.0-x86_64-gp2 |
| ami-01e3c4a339a264cc9 | amzn2-ami-hvm-2.0.20241014.0-x86_64-gp2 |
| ami-014d544cfe21b2d2 | amzn2-ami-hvm-2.0.20240223.0-x86_64-gp2 |
| ami-02d69c34f7a0bf56a | amzn2-ami-hvm-2.0.20230418.0-x86_64-gp2 |
| ami-02b972fec07f1e659 | amzn2-ami-hvm-2.0.20221103.3-x86_64-gp2 |
| ami-04c09ef2f505b609 | amzn2-ami-hvm-2.0.20240109.0-x86_64-gp2 |
| ami-03c951bbe993ea887 | amzn2-ami-hvm-2.0.20240131.0-x86_64-gp2 |
| ami-03d1b2fca19c17cf1 | amzn2-ami-hvm-2.0.20230118.2-x86_64-gp2 |
| ami-03b8c4c9d51a0a8999 | amzn2-ami-hvm-2.0.20221119.0-x86_64-gp2 |
| ami-046eeba8a7f7b0ef0 | amzn2-ami-hvm-2.0.20240124.0-x86_64-gp2 |
| ami-045602374a1982480 | amzn2-ami-hvm-2.0.20240412.0-x86_64-gp2 |
| ami-0476f2eeba10d899bc | amzn2-ami-hvm-2.0.20230719.0-x86_64-gp2 |
| ami-052b9fbbe949f883a | amzn2-ami-hvm-2.0.20240916.0-x86_64-gp2 |
| ami-05b5badc2f7ddd88d | amzn2-ami-hvm-2.0.20230221.0-x86_64-gp2 |
```

After this run the following command

```
terraform init
```

```
terraform apply
```

The screenshot shows the VSCode interface with the terminal tab active. The terminal window displays the Terraform apply command and its confirmation dialog. The code editor shows the Terraform configuration file (main.tf) with a resource block for an AWS Lambda function.

```
PS D:\Sem6\anuprita\AdvDevops_Practical> terraform apply
+-----+
|   + tags_all           = { |
|   |     + "Name"          = "SonarQubeServer" |
|   |     + tenancy          = (known after apply) |
|   |     + user_data         = "84e016249e5fd73535afcb3769cc27626514b358" |
|   |     + user_data_base64  = (known after apply) |
|   |     + user_data_replace_on_change = false |
|   |     + vpc_security_group_ids = (known after apply) |
|   |     + capacity_reservation_specification (known after apply) |
|   |     + cpu_options (known after apply) |
|   |     + ebs_block_device (known after apply) |
|   |     + enclave_options (known after apply) |
|   |     + ephemeral_block_device (known after apply) |
|   |     + instance_market_options (known after apply) |
|   |     + maintenance_options (known after apply) |
|   |     + metadata_options (known after apply) |
|   |     + network_interface (known after apply) |
|   |     + private_dns_name_options (known after apply) |
|   |     + root_block_device (known after apply) |
|   } |
|
| Plan: 2 to add, 0 to change, 0 to destroy.
|
| Do you want to perform these actions?
| Terraform will perform the actions described above.
| Only 'yes' will be accepted to approve.
|
| Enter a value: yes
```

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

- EXPLORER**: Shows the project structure under "ADDEVOPS PRACTICAL" with files: .terraform, .terraform.lock.hcl, hello.py, main.tf, terraform.tfstate, and terraform.tfstate.backup.
- TERMINAL**: Displays the Terraform command-line interface output. It shows the plan phase with 2 resources to add, and then applies the changes successfully.
- OUTPUT**: Shows the Terraform configuration code.
- DEBUG CONSOLE**: Shows the PowerShell session output.
- PORTS**: Shows the PowerShell session output.
- STATUS BAR**: Shows the file path as PS D:\Sem5_anuprita\AdDevops_Practical> and the terminal status as Ln 49, Col 1 Spaces: 2 UTF-8 Plain Text Go Live Prettier.

```

+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.sonarqube: Creating...
aws_instance.jenkins: Creating...
aws_instance.sonarqube: Still creating... [10s elapsed]
aws_instance.jenkins: Still creating... [10s elapsed]
aws_instance.sonarqube: Creation complete after 16s [id=i-09f5733c9d7c06309]
aws_instance.jenkins: Creation complete after 16s [id=i-0b2d8414cbfb34c5]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS D:\Sem5_anuprita\AdDevops_Practical>

```

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 with the following details:

- Left Sidebar**: Includes sections for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations.
- Instances Tab**: Shows 2 instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
JenkinsServer	i-05f12961a9cf8cf3e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-98-80
SonarQubeSer...	i-061f29e11e3fa1a8b	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c	ec2-54-21
- Bottom Status Bar**: Shows the copyright notice "© 2024, Amazon Web Services, Inc. or its affiliates.", links for Privacy, Terms, and Cookie preferences.

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 AMjenkins-security and give some description and add the inbounds rules given below.

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

The screenshot shows the AWS Management Console with the 'Security Groups' page open. The left sidebar includes sections for Events, Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups). The main content area displays a table titled 'Security Groups (23) Info' with columns: Name, Security group ID, Security group name, VPC ID, and Description. The table lists several security groups, such as 'launch-wizard-6', 'sg-01e6da10b21246fb', and 'allow_http'. A search bar at the top allows filtering by attribute or tag. Action buttons like 'Actions' and 'Create security group' are available at the top right.

The screenshot shows the 'Edit inbound rules' page for the security group 'sg-06766b363e07ace3b - AMjenkins-security'. The top navigation bar includes links for EC2, Security Groups, sg-06766b363e07ace3b - AMjenkins-security, and Edit inbound rules. Below the navigation, a message states: 'Inbound rules control the incoming traffic that's allowed to reach the instance.' The main content area is titled 'Inbound rules Info' and contains a table with columns: Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. Three rules are listed:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0d9d57127ff521991	Custom TCP	TCP	8080	Custom	49.36.105.96/32
sgr-057e1247e95e38c1c	SSH	TCP	22	Custom	0.0.0.0/0
sgr-07584f2ee4e6eebfaf	Custom TCP	TCP	8080	Custom	0.0.0.0/0

A button labeled 'Add rule' is located at the bottom left. A warning message at the bottom left states: '⚠️ 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.' At the bottom right are buttons for 'Cancel', 'Preview changes', and 'Save rules'.

EC2 > Security Groups > sg-04d93757f217a03e0 - AMsonarqube-security > Edit inbound rules

Edit inbound rules Info

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

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-0b145544ddb02dbd0	SSH	TCP	22	Custom	0.0.0.0/0 <small>X</small>
sgr-0ca54f064e2b116c9	Custom TCP	TCP	9000	Custom	49.36.105.96/32 <small>X</small>
sgr-0b9649cd8bb7a9401	Custom TCP	TCP	9000	Custom	0.0.0.0/0 <small>X</small>

[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.

[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.

EC2 Dashboard EC2 Global View Events Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations New Images AMIs AMI Catalog Elastic Block Store Volumes Snapshots Lifecycle Manager Network & Security Security Groups Elastic IPs Placement Groups

Instances (1/2) Info Last updated 1 minute ago Connect Instance state Actions Launch Instances

Find Instance by attribute or tag (case-sensitive) All states

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
JenkinsServer	i-05f12961a9cf8cf3e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-98-80-223-40
SonarQubeSer...	i-061f29e11e3fa1a8b	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c	ec2-54-21-11-111

i-05f12961a9cf8cf3e (JenkinsServer)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary Public IPv4 address 98.80.223.40 | open address Private IPv4 addresses 172.31.42.86

Instance ID i-05f12961a9cf8cf3e (JenkinsServer) Instance state Running Public IPv4 DNS ec2-98-80-223-40.compute-1.amazonaws.com | open address

IPv6 address - Private IP DNS name (IPv4 only) ip-172-31-42-86.ec2.internal

Hostname type IP name: ip-172-31-42-86.ec2.internal Answer private resource DNS name Instance type Elastic IP addresses

[https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance\\$instanceId=i-05f12961a9cf8cf3e](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance$instanceId=i-05f12961a9cf8cf3e)

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

S | Services | Search [Alt+S] | N. Virginia | voclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9795 ▾

EC2 > Instances > i-05f12961a9cf8cf3e > Connect to instance

Connect to instance Info

Connect to your instance i-05f12961a9cf8cf3e (JenkinsServer) using any of these options

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

Instance ID
 i-05f12961a9cf8cf3e (JenkinsServer)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is AMjenkins.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "AMjenkins.pem"
4. Connect to your instance using its Public DNS:
 ec2-98-80-223-40.compute-1.amazonaws.com

Example:
 ssh -i "AMjenkins.pem" ec2-user@ec2-98-80-223-40.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

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 2:

Open Git Bash and go to the directory which has the Key downloaded. If you don't have the key downloaded, create a key 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 AMjenkins.pem*

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

```
ec2-user@ip-172-31-42-155:~ 
User@DESKTOP-QOK15A MINGW64 ~ (master)
$ cd Downloads
User@DESKTOP-QOK15A MINGW64 ~/Downloads (master)
$ dir AMjenkins.pem*
AMjenkins.pem

User@DESKTOP-QOK15A MINGW64 ~/Downloads (master)
$ ssh -i "AMjenkins.pem" ec2-user@ec2-54-91-87-54.compute-1.amazonaws.com
The authenticity of host 'ec2-54-91-87-54.compute-1.amazonaws.com (54.91.87.54)' can't be established.
ED25519 key fingerprint is SHA256:LNZVrp4MguqGAg79Kdfgnah0moJppMk06Ke+1bzoEMM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added 'ec2-54-91-87-54.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
'          #_
  _\##_      Amazon Linux 2023
  _\###\_
  _\##\_
  _\#/   https://aws.amazon.com/linux/amazon-linux-2023
  _\#/   _/_
  _\#/   _/_
[ec2-user@ip-172-31-42-155 ~]$ |
```

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/>

The screenshot shows a Google search results page for "Jenkins". The first result is a link to the Jenkins website (<https://www.jenkins.io>). Below it are several other links related to Jenkins, such as "Download and deploy", "Installing Jenkins", "Jenkins User Documentation", "Tutorials overview", and "Pipeline". To the right of the search results is a detailed "Software" card for Jenkins. It includes a Jenkins logo, a screenshot of the Jenkins interface, and sections for "Programming language: Java", "Developer: Kohsuke Kawaguchi", "Initial release: 2 February 2011", "License: MIT License", "Platform: Java 11, Java 17, Java 21", "Repository: github.com/jenkinsci/jenkins", and "Stable release: 2.477 / 17 September 2024; 18 days ago".

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 root user.

The screenshot shows the Jenkins Redhat Packages page. It starts with a heading "Jenkins Redhat Packages" and a note: "To use this repository, run the following command:" followed by two terminal 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
```

Below this, a note says: "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."

Then, it shows a terminal command:

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

A note states: "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]
```

Finally, a note says: "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
```

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

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

The screenshot shows a terminal session on an Amazon Linux 2023 system. The user is installing Jenkins and various Java-related packages using yum and dnf.

```
user@DESKTOP-00GK15A MINGW64 ~\Downloads (master)  
$ ssh -i "Amjenkins.pem" ec2-user@ec2-54-91-87-54.compute-1.amazonaws.com  
Amazon Linux 2023.09.0  
https://aws.amazon.com/linux/amazon-linux-2023  
Last login: Fri Oct 18 14:57:44 2024 from 49.36.105.208  
[ec2-user@ip-172-31-42-155 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo  
[ec2-user@ip-172-31-42-155 ~]$ curl -L https://pkg.jenkins.io/redhat-stable/jenkins.repo | tee /etc/yum.repos.d/jenkins.repo  
Resolving [pkg.jenkins.io (pkg.jenkins.io)]... 166.75.34.133, 2a04:4e42:78::645  
Connecting to pkg.jenkins.io (pkg.jenkins.io)[166.75.34.133]:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 85  
Saving to: '/etc/yum.repos.d/jenkins.repo'  
/etc/yum.repos.d/jenkins.repo 100%[=====] 85 --.KB/s in 0s  
2024-10-18 15:11:26 (3.60 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]  
[ec2-user@ip-172-31-42-155 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key  
[ec2-user@ip-172-31-42-155 ~]$ yum install fontconfig java-17-openjdk  
Error: This command has to be run with superuser privileges (under the root user on most systems).  
[ec2-user@ip-172-31-42-155 ~]$ sudo yum install fontconfig java-17-openjdk  
Jenkins-stable  
No match for argument: java-17-openjdk  
Error: Unable to find a match: java-17-openjdk  
[ec2-user@ip-172-31-42-155 ~]$ sudo dnf install java-17-amazon-corretto-devel  
Last metadata expiration check: 0:02:40 ago on Fri Oct 18 15:12:06 2024.  
Dependencies resolved.  
----  
| Package           | Architecture | Version      | Repository |  
|----|----|----|----|  
| Installing dependencies:  
| java-17-amazon-corretto-devel | x86_64       | 1:17.0.12+7-1.amzn2023.1 | amazonlinux | 142 k  
| alsa-lib          | x86_64       | 1:2.7.2-1.amzn2023.0.2   | amazonlinux | 504 k  
| cairo             | x86_64       | 1:17.6-2.amzn2023.0.1   | amazonlinux | 684 k  
| dejavu-sans-fonts | noarch      | 2.37-16.amzn2023.0.2    | amazonlinux | 1.3 M  
| dejavu-sans-mono-fonts | noarch      | 2.37-16.amzn2023.0.2    | amazonlinux | 467 k  
| dejavu-serif-fonts | noarch      | 2.37-16.amzn2023.0.2    | amazonlinux | 1.0 M  
| fontconfig        | x86_64       | 2.13.1-12.amzn2023.0.2   | amazonlinux | 32 k  
| fonts-filenamesystem | noarch      | 1:2.0.5-12.amzn2023.0.2   | amazonlinux | 9.5 k  
| freetype          | x86_64       | 2.13.2-5.amzn2023.0.1    | amazonlinux | 423 k  
| google-noto-fonts-common | noarch      | 20201206-2.amzn2023.0.2   | amazonlinux | 15 k  
| google-noto-sans-vf-fonts | noarch      | 20201206-2.amzn2023.0.2   | amazonlinux | 492 k  
| graphite2         | x86_64       | 1:3.3.14-7.amzn2023.0.2   | amazonlinux | 97 k  
| harfbuzz          | x86_64       | 7.0.0-0.amzn2023.0.1     | amazonlinux | 868 k  
| java-17-amazon-corretto-headless | x86_64       | 1:17.0.12+7-1.amzn2023.1 | amazonlinux | 91 M  
| jpackagestags-filesystem | noarch      | 6.0.0-0.7.amzn2023.0.6    | amazonlinux | 12 k  
| langpacks-core-font-en | noarch      | 3.0-21.amzn2023.0.4      | amazonlinux | 10 k  
| libXi              | x86_64       | 1:8.10-2.amzn2023.0.1    | amazonlinux | 659 k  
| libXext             | x86_64       | 1:1.3.1-1.amzn2023.0.1    | amazonlinux | 14 k  
| libXt              | x86_64       | 1:0.11-6.amzn2023.0.1    | amazonlinux | 33 k  
| libXrender          | x86_64       | 1:3.6-1.amzn2023.0.1     | amazonlinux | 42 k  
| libXtst             | x86_64       | 0.9.11-6.amzn2023.0.1    | amazonlinux | 29 k  
| libjpeg-turbo       | x86_64       | 1:0.9-4.amzn2023.0.2     | amazonlinux | 315 k  
| libpng              | x86_64       | 2:1.4-2.amzn2023.0.5     | amazonlinux | 190 k  
| libtiff             | x86_64       | 2:1.6.37-10.amzn2023.0.6 | amazonlinux | 128 k
```

```

ec2-user@ip-172-31-42-155:~ Transaction Summary
Install 27 Packages
Total download size: 100 M
Installed size: 261 M
Is this ok [y/N]: y
Downloading Packages:
(1/27): cairo-1.17.6-2.amzn2023.0.1.x86_64.rpm
(2/27): dejavu-sans-fonts-2.37-16.amzn2023.0.2.noarch.rpm
(3/27): alsa-lib-1.2.7.2-1.amzn2023.0.2.x86_64.rpm
(4/27): dejavu-serif-fonts-2.37-16.amzn2023.0.2.noarch.rpm
(5/27): fonts-filesystem-2.0.5-12.amzn2023.0.2.noarch.rpm
(6/27): dejavu-sans-mono-fonts-2.37-16.amzn2023.0.2.noarch.rpm
(7/27): fontconfig-2.13.94-2.amzn2023.0.2.x86_64.rpm
(8/27): google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch.rpm
(9/27): graphite2-1.3.14-7.amzn2023.0.2.x86_64.rpm
(10/27): freetype-2.13.7-5.amzn2023.0.1.x86_64.rpm
(11/27): harfbuzz-7.0.0-2.amzn2023.0.1.x86_64.rpm
(12/27): google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch.rpm
(13/27): java-17-amazon-corretto-devel-17.0.12+7-1.amzn2023.1.x86_64.rpm
(14/27): javapackages-filesystem-6.0.0-7.amzn2023.0.6.noarch.rpm
(15/27): langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch.rpm
(16/27): libpix11-1.8.10-2.amzn2023.0.1.x86_64.rpm
(17/27): libpix11-common-1.8.10-2.amzn2023.0.1.noarch.rpm
(18/27): libpixau-1.0.11-6.amzn2023.0.1.x86_64.rpm
(19/27): libxext-1.3.6-1.amzn2023.0.1.x86_64.rpm
(20/27): libxrender-0.9.11-6.amzn2023.1.x86_64.rpm
(21/27): libjpeg-turbo-2.1.4-7.amzn2023.0.5.x86_64.rpm
(22/27): libprotobuf-1.0.9-4.amzn2023.0.2.x86_64.rpm
(23/27): libpng-1.6.37-10.amzn2023.0.6.x86_64.rpm
(24/27): libxcb-1.17.0-1.amzn2023.0.1.x86_64.rpm
(25/27): xml-common-0.6.3-56.amzn2023.0.2.noarch.rpm
(26/27): pixman-0.43.4-1.amzn2023.0.4.x86_64.rpm
(27/27): java-17-amazon-corretto-headless-17.0.12+7-1.amzn2023.1.x86_64.rpm

total 54 MB/s | 100 MB 00:01

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : fonts-filesystem-2.0.5-12.amzn2023.0.2.noarch 2/27
Installing : dejavu-sans-mono-fonts-2.37-16.amzn2023.0.2.noarch 3/27
Installing : dejavu-sans-fonts-2.37-16.amzn2023.0.2.noarch 4/27
Installing : libpng-2.1.6-37-10.amzn2023.0.6.x86_64 5/27
Installing : dejavu-serif-fonts-2.37-16.amzn2023.0.2.noarch 6/27
Running scriptlet: xml-common-0.6.3-56.amzn2023.0.2.noarch 7/27
Installing : xkb-common-0.3.56.amzn2023.0.2.noarch 8/27
Installing : pixman-0.43.4-1.amzn2023.0.4.x86_64 9/27
Installing : libjpeg-turbo-2.1.4-7.amzn2023.0.5.x86_64 10/27
Installing : libprotobuf-1.0.9-4.amzn2023.0.2.x86_64 11/27
Installing : libpixau-1.0.11-6.amzn2023.0.1.x86_64 12/27
Installing : libxcb-1.17.0-1.amzn2023.0.1.x86_64 13/27
Installing : libpix11-common-1.8.10-2.amzn2023.0.1.noarch 14/27
Installing : libpix11-1.8.10-2.amzn2023.0.1.x86_64 15/27
Installing : libxext-1.3.6-1.amzn2023.0.1.x86_64 16/27
Installing : libxrender-0.9.11-6.amzn2023.1.x86_64 17/27
Installing : javapackages-filesystem-6.0.0-7.amzn2023.0.6.noarch 18/27
Installing : graphite2-1.3.14-7.amzn2023.0.2.x86_64 19/27
Installing : google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch 20/27
Installing : google-noto-fonts-vf-fonts-20201206-2.amzn2023.0.2.noarch 21/27
Installing : langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch 22/27

```

```
ec2-user@ip-172-31-42-155: ~
```

Installing : libhx11-1.8.10-2_amzn2023.0.1.x86_64 13/27
Installing : libhxext-1.3.6-1_amzn2023.0.1.x86_64 14/27
Installing : libixerender-0.9.11-6_amzn2023.0.1.x86_64 15/27
Installing : javapackages-fscommon-6.0.0-7_amzn2023.0.6.noarch 16/27
Installing : graphite2-1.3.14-7_amzn2023.0.2.x86_64 17/27
Installing : google-noto-fonts-common-20201206-2_amzn2023.0.2.noarch 18/27
Installing : google-noto-sans-vf-fonts-20201206-2_amzn2023.0.2.noarch 19/27
Installing : langpacks-core-font-en-3.0-21_amzn2023.0.4.noarch 20/27
Installing : cairo-1.17.6-2_amzn2023.0.1.x86_64 21/27
Installing : fontconfig-2.13.90-2_amzn2023.0.1.x86_64 22/27
Installing : freetype-2.13.2-5_amzn2023.0.1.x86_64 23/27
Installing : fontconfig-2.13.94-2_amzn2023.0.2.x86_64 24/27
Running scriptlet: fontconfig-2.13.94-2_amzn2023.0.2.x86_64 24/27
Installing : alsalib-1.2.7-2-1_amzn2023.0.2.x86_64 25/27
Installing : java-17-amazon-corretto-headless-1:17.0.12+7-1_amzn2023.1.x86_64 26/27
Running scriptlet: java-17-amazon-corretto-headless-1:17.0.12+7-1_amzn2023.1.x86_64 26/27
Installing : java-17-amazon-corretto-devel-1:17.0.12+7-1_amzn2023.1.x86_64 27/27
Running scriptlet: java-17-amazon-corretto-devel-1:17.0.12+7-1_amzn2023.1.x86_64 27/27
Running scriptlet: java-17-amazon-corretto-devel-1:17.0.12+7-1_amzn2023.1.x86_64 27/27
Running scriptlet: java-17-amazon-corretto-devel-1:17.0.12+7-1_amzn2023.1.x86_64 27/27
Verifying : alsalib-1.2.7.2-2_amzn2023.0.2.x86_64 1/27
Verifying : cairo-1.17.6-2_amzn2023.0.1.x86_64 2/27
Verifying : dejavu-sans-fonts-2.37-16_amzn2023.0.2.noarch 3/27
Verifying : dejavu-sans-mono-fonts-2.37-16_amzn2023.0.2.noarch 4/27
Verifying : dejavu-serif-fonts-2.37-16_amzn2023.0.2.noarch 5/27
Verifying : fontconfig-2.13.94-2_amzn2023.0.2.x86_64 6/27
Verifying : fonts-fscommon-1:2.0.0-12_amzn2023.0.2.noarch 7/27
Verifying : freetype-2.13.2-5_amzn2023.0.1.x86_64 8/27
Verifying : javapackages-fscommon-20201206-2_amzn2023.0.2.noarch 9/27
Verifying : google-noto-sans-vf-fonts-20201206-2_amzn2023.0.2.noarch 10/27
Verifying : graphite2-1.3.14-7_amzn2023.0.2.x86_64 11/27
Verifying : harfbuzz-0.7.0-2_amzn2023.0.1.x86_64 12/27
Verifying : java-17-amazon-corretto-devel-1:17.0.12+7-1_amzn2023.1.x86_64 13/27
Verifying : java-17-amazon-corretto-headless-1:17.0.12+7-1_amzn2023.1.x86_64 14/27
Verifying : javapackages-fscommon-6.0.0-7_amzn2023.0.6.noarch 15/27
Verifying : langpacks-core-font-en-3.0-21_amzn2023.0.4.noarch 16/27
Verifying : libhx11-1.8.10-2_amzn2023.0.1.x86_64 17/27
Verifying : libhxau-1.0.11-6_amzn2023.0.1.x86_64 18/27
Verifying : libhxext-1.3.6-1_amzn2023.0.1.x86_64 19/27
Verifying : libixerender-0.9.11-6_amzn2023.0.1.x86_64 20/27
Verifying : libbrotli-0.9.4_amzn2023.0.2.x86_64 21/27
Verifying : libjpeg-turbo-2.1.4-2_amzn2023.0.5.x86_64 22/27
Verifying : libpng-2:1.6.37-10_amzn2023.0.6.x86_64 23/27
Verifying : libxcb-1.17-0-1_amzn2023.0.1.x86_64 24/27
Verifying : pixman-0.43.4-1_amzn2023.0.4.x86_64 25/27
Verifying : xml-common-0.6.3-56_amzn2023.0.2.noarch 26/27
Installed:
alsalib-1.2.7.2-2_amzn2023.0.2.x86_64
cairo-1.17.6-2_amzn2023.0.1.x86_64
dejavu-sans-fonts-2.37-16_amzn2023.0.2.noarch

```
dejavu-sans-mono-fonts-2.37-16_amzn2023.0.2.noarch
fonts-filesystem-1:2.0-5.12_amzn2023.0.2.noarch
google-noto-sans-vf-fonts_20201206-2_amzn2023.0.2.noarch
java-17-amazon-corretto-devel-1:17.0.12-7_amzn2023.1.x86_64
langpacks-core-font-en-3-0.21_amzn2023.0.4.noarch
libXau-1.0.11-6_amzn2023.0.1.x86_64
librtrobj-1.0-9.4_amzn2023.0.2.x86_64
libxcb-1.17-0.1_amzn2023.0.1.x86_64
```

```
dejavu-serif-fonts-2.37-16_amzn2023.0.2.noarch
freetype-2.13-2_amzn2023.0.1.x86_64
graphite2-1.3.14-7_amzn2023.0.2.x86_64
java-17-amazon-corretto-headless-1:1.17.0.12+7-1_amzn2023.1.x86_64
jibcurl-1.1.8-10_amzn2023.1.x86_64
jibcurl-ext-1.3.6-1_amzn2023.0.1.x86_64
libjpeg-turbo-2.1.4-2_amzn2023.0.5.x86_64
pixman-0.43.4-1_amzn2023.0.4.x86_64
```

```
fontconfig-2.13.94-2_amzn2023.0.2_x86_64
google-fonts-common-20202016-2_amzn2023.0.2_noarch
harfbuzz-7.0.0-2_amzn2023.0.1_x86_64
javapackages-filesystem-6.0-0.7_amzn2023.0.6_noarch
libI18NCommon-1.8-10_amzn2023.0.1_noarch
libXrender-0.9.11-6_amzn2023.0.1_x86_64
libpng-2.1.6-37_amzn2023.0.6_x86_64
lxml-4.9.1-3_amzn2023.0.2_noarch
python3.8-3.8.12-1_amzn2023.0.2_noarch
python3.8-setuptools-63.4.3-1_amzn2023.0.2_noarch
python3.8-wheel-3.6.3-1_amzn2023.0.2_noarch
```

```
Complete!
[ec2-user@ip-172-31-42-155 ~]$ java --version
openjdk 17.0.12 2024-07-16 LTS
OpenJDK Runtime Environment Corretto-17.0.12.7.1 (build 17.0.12+7)
OpenJDK 64-Bit Server VM Corretto-17.0.12.7.1 (build 17.0.12+7)
```

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

```
ec2-user@ip-172-31-42-155:~
```

Package	Architecture	Version	Repository	
Installing:				
jenkins	noarch	2.462.3-1.1	jenkins	89 M
Transaction Summary				
Install 1 Package				
total download size: 89 M				
installed size: 89 M				
Is this ok [y/N]: y				
Downloading Packages:				
jenkins-2.462.3-1.1.noarch.rpm				13 MB/s 89 MB 00:06
Total				
Running transaction check				13 MB/s 89 MB 00:06
Transaction check succeeded.				
Running transaction test				
Transaction test succeeded.				
Running transaction				
Preparing transaction:				
Running scriptlet: jenkins-2.462.3-1.1.noarch				
Installing : jenkins-2.462.3-1.1.noarch				1/1
Running scriptlet: jenkins-2.462.3-1.1.noarch				1/1
Verifying : jenkins-2.462.3-1.1.noarch				1/1
Installed:				
jenkins-2.462.3-1.1.noarch				
Completed!				

Step 7:

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

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. The main content displays two instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
JenkinsServer	i-05f12961a9cf8cf3e	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-98-80-223-40
SonarQubeSer...	i-061f29e11e3fa1a8b	Running	t2.medium	2/2 checks passed	View alarms	us-east-1c	ec2-54-21-11-111

A modal window for the JenkinsServer instance is open, showing its details. The Public IPv4 address is highlighted and copied to the clipboard (98.80.223.40). Other details include Instance ID (i-05f12961a9cf8cf3e), Instance state (Running), Hostname type (IP name: ip-172-31-42-86.ec2.internal), Private IP DNS name (ip-172-31-42-86.ec2.internal), and Instance type (t2.micro).

Step 8:

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

The screenshot shows the Jenkins 'Unlock Jenkins' configuration page. The title is 'Getting Started' and the section is 'Unlock Jenkins'. It instructs the user to copy the password from the log file /var/lib/jenkins/secrets/initialAdminPassword. A text input field is provided for the 'Administrator password'.

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.

```

ec2-user@ip-172-31-42-86:~$ aliases: install-n, install-na, install-nevra
[ec2-user@ip-172-31-42-86 ~]$ sudo yum installclient_loop: send disconnect: Connection reset by peer
User@DESKTOP-DOOK15A MINGW64 ~/Downloads (master)
$ ssh -i "Ajenkins.pem" ec2-user@ec2-98-80-223-40.compute-1.amazonaws.com
Last login: Sat Oct 19 16:16:12 2024 from 49.36.105.96
.
.~_ ##### 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/
[ec2-user@ip-172-31-42-86 ~]$ sudo dnf install java-17-amazon-corretto-devel
sudo: dnf: command not found
[ec2-user@ip-172-31-42-86 ~]$ nc
[ec2-user@ip-172-31-42-86 ~]$ sudo yum install java-17-amazon-corretto-devel
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package java-17-amazon-corretto-devel-17.0.12+7-1.amzn2.1.x86_64 already installed and latest version
| 3.6 kB 00:00:00
No Match for argument: java
[ec2-user@ip-172-31-42-86 ~]$ sudo systemctl status jenkins
sudo: systemctl: command not found
[ec2-user@ip-172-31-42-86 ~]$ nc
[ec2-user@ip-172-31-42-86 ~]$ 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-172-31-42-86 ~]$ sudo systemctl enable jenkins
Failed to execute operation: No such file or directory
[ec2-user@ip-172-31-42-86 ~]$ sudo systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-172-31-42-86 ~]$ 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-172-31-42-86 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-42-86 ~]$ 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 Sat 2024-10-19 16:37:17 UTC; 10s ago
Main PID: 13065 (Java)
  CGroup: /system.slice/jenkins.service
          └─13065 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Oct 19 16:37:10 ip-172-31-42-86.ec2.internal jenkins[13065]: 174b824d318f44658530f65bebff39ff2
Oct 19 16:37:10 ip-172-31-42-86.ec2.internal jenkins[13065]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Oct 19 16:37:10 ip-172-31-42-86.ec2.internal jenkins[13065]: ****
Oct 19 16:37:10 ip-172-31-42-86.ec2.internal jenkins[13065]: 2024-10-19 16:37:17.478+0000 [id=32]      INFO  jenkins.InitReactorRunner$1#onAttained: Completed initialization
Oct 19 16:37:10 ip-172-31-42-86.ec2.internal jenkins[13065]: 2024-10-19 16:37:17.516+0000 [id=24]      INFO  hudson.lifecycle.Lifecycle$OnReady: Jenkins is fully up and running
Oct 19 16:37:10 ip-172-31-42-86.ec2.internal jenkins[13065]: Started Jenkins Continuous Integration Server.
Oct 19 16:37:10 ip-172-31-42-86.ec2.internal jenkins[13065]: 2024-10-19 16:37:17.584+0000 [id=48]      INFO  h.m.DownloadService$Downloadable$load: Obtained the updated data file for huds...venInstaller
Oct 19 16:37:10 ip-172-31-42-86.ec2.internal jenkins[13065]: 2024-10-19 16:37:17.584+0000 [id=48]      INFO  hudson.util.Retry$#start: Performed the action check updates server successfully attempt #1
Hint: Some lines were ellipsized, use ... to show in full.
[ec2-user@ip-172-31-42-86 ~]$ sudo more /var/lib/jenkins/secrets/initialAdminPassword
174b824d318f44658530f65bebff39ff2
[ec2-user@ip-172-31-42-86 ~]$ nc
[ec2-user@ip-172-31-42-86 ~]$ |

```

Not secure | 98.80.223.40:8080/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

.....

[Continue](#)

Step 10:

Select install suggested plugins and complete the installation and initial configurations.

Not secure | 98.80.223.40:8080

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.462.3

Not secure | 98.80.223.40:8080

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	** Ionicons API Folders OWASP Markup Formatter ** ASH API ** JSON Path API ** Structs ** Pipeline: Step API ** Token Macro Build Timeout ** Pipeline: Pipeline API ** Credentials ** Plain Credentials ** Variant ** SSH Credentials Credentials Binding ** GH API ** Pipeline: API ** commons-lang3 v3.x Jenkins API Timestamper ** Caffeine API ** Script Security ** JavaBeans Activation Framework (JAF) API ** JAXB ** SnakeYAML API ** JSON Api ** Jackson 2 API ** - required dependency
✓ Timestamper	⌚ Workspace Cleanup	⌚ Ant	⌚ Gradle	
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline Graph View	
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ PAM Authentication	
⌚ LDAP	⌚ Email Extension	⌚ Mailer	⌚ Dark Theme	

Jenkins 2.462.3

Not secure | 98.80.223.40:8080

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

Jenkins 2.462.3

Skip and continue as admin Save and Continue

Password: Anuprita@4321

Not secure | 98.80.223.40:8080

Getting Started

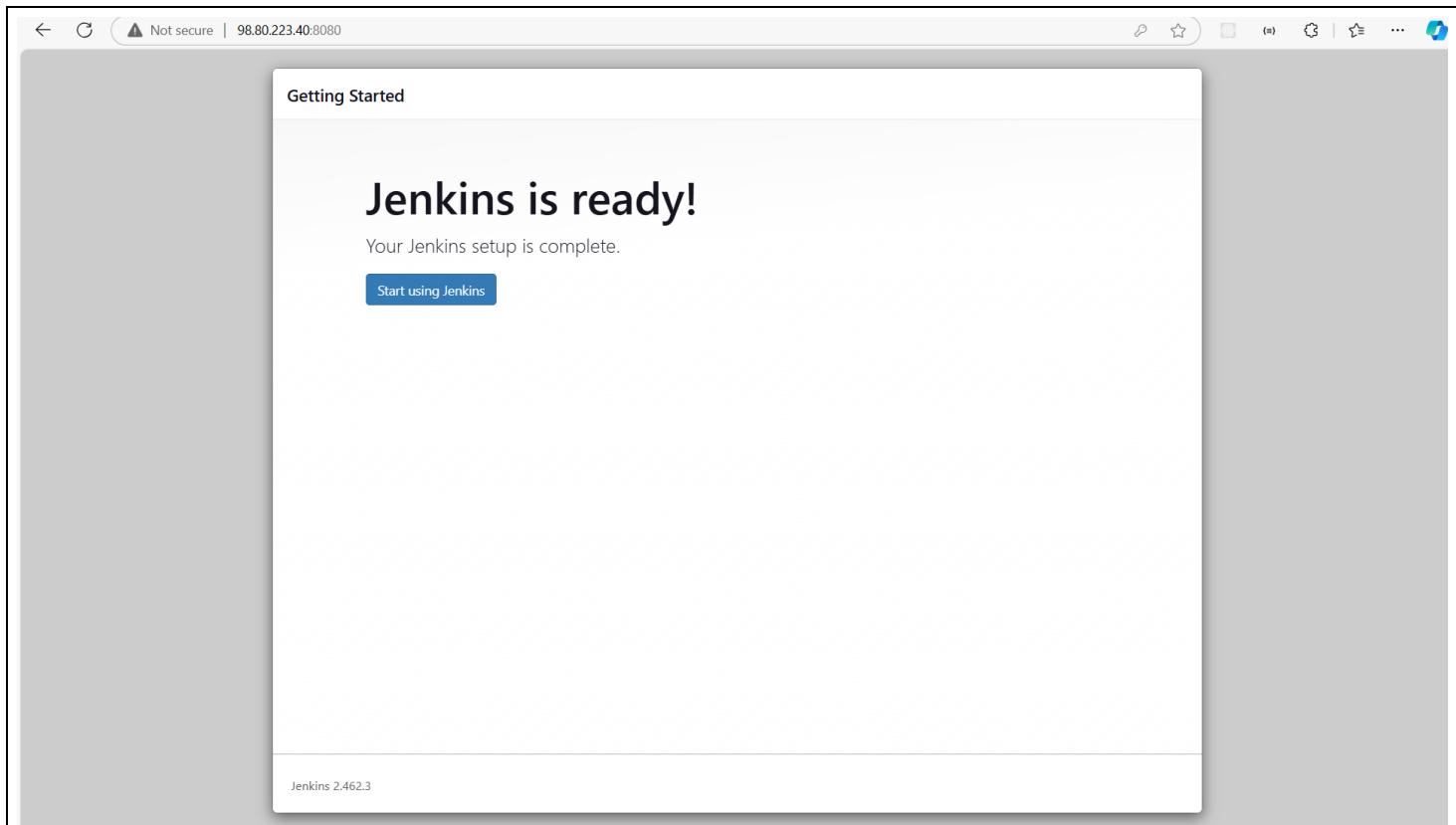
Instance Configuration

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



Step 11:

After proper initial configuration you will be redirected to this page.

A screenshot of the Jenkins Dashboard page. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'Anuprita'. The dashboard features several sections: 'Build Queue' (No builds in the queue), 'Build Executor Status' (1 Idle, 2 Idle), 'Welcome to Jenkins!' (a main heading with a subtext about displaying Jenkins jobs), 'Start building your software project' (a button to 'Create a job'), 'Set up a distributed build' (links to 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'), and footer links for 'REST API' and 'Jenkins 2.462.3'.

Step 12:

Install more plugins which will be required for this experiment.

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

Not secure | 98.80.223.40:8080/manage/

Jenkins

Dashboard > Manage Jenkins

+ New Item Build History Manage Jenkins My Views

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

[Set up agent](#) [Set up cloud](#) [Dismiss](#)

Operating system end of life monitor

You are running Jenkins on Amazon Linux 2. Jenkins stopped supporting Amazon Linux 2 as of **2023-11-16**. Please upgrade to a supported operating system. Refer to [the documentation](#) for details.

[More Info](#) [Ignore](#)

System Configuration

[System](#) Configure global settings and paths.

[Tools](#) Configure tools, their locations and automatic installers.

[Plugins](#) Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

[Nodes](#) Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

[Clouds](#) Add, remove, and configure cloud instances to provision agents on-demand.

[Appearance](#) Configure the look and feel of Jenkins

98.80.223.40:8080/manage/pluginManager Security

Not secure | 98.80.223.40:8080/manage/pluginManager/available

Jenkins

Dashboard > Manage Jenkins > Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Search: sonarqube

Install Name Released

Install	Name	Released
<input checked="" type="checkbox"/>	SonarQube Scanner 2.17.2	8 mo 3 days ago
<input type="checkbox"/>	External Site/Tool Integrations	
<input type="checkbox"/>	Build Reports	
	This plugin allows an easy integration of SonarQube , the open source platform for Continuous Inspection of code quality.	
<input type="checkbox"/>	Sonar Gerrit 388.v9b_ffcb_e42306	4 mo 17 days ago
<input type="checkbox"/>	External Site/Tool Integrations	
<input type="checkbox"/>	Build Reports	
	This plugin allows to submit issues from SonarQube to Gerrit as comments directly.	
<input type="checkbox"/>	SonarQube Generic Coverage 1.0	5 yr 2 mo ago
<input type="checkbox"/>	External Site/Tool Integrations	
<input type="checkbox"/>	Build Reports	
	TODO	

REST API Jenkins 2.462.3

Not secure | 98.80.223.40:8080/manage/pluginManager/available

Plugins

Search (CTRL+K)

Install

Available plugins

Released

Install	Name	Released
<input type="checkbox"/>	Pipeline: REST API 2.34 User Interface Provides a REST API to access pipeline and pipeline run data.	11 mo ago
<input checked="" type="checkbox"/>	Pipeline: Stage View 2.34 User Interface Pipeline Stage View Plugin.	11 mo ago
<input type="checkbox"/>	Docker Pipeline 580.vc0c340686b_54 pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	5 mo 0 days ago
<input type="checkbox"/>	Lockable Resources 1320.v1f0dff578476 pipeline Cluster Management Agent Management This plugin allows to define external resources (such as printers, phones, computers) that can be locked by builds. If a build requires an external resource which is already locked, it will wait for the resource to be free.	3 days 22 hr ago
<input type="checkbox"/>	Pipeline: Deprecated Groovy Libraries 612.v55f2f80781ef Miscellaneous Hosting of Pipeline Groovy libraries inside a Jenkins Git server. Deprecated . Use Pipeline: Groovy Libraries instead. If you see this plugin installed just because you upgraded, you can probably uninstall it now. This plugin should only be used if you have historically <i>pushed</i> libraries to a Git server inside Jenkins.	9 mo 4 days ago

This plugin is deprecated. In general, this means that it is either obsolete, no longer being developed, or may no longer work. Learn more

Not secure | 98.80.223.40:8080/manage/pluginManager/updates/



Please wait while Jenkins is restarting ...

Your browser will reload automatically when Jenkins is ready.

Safe Restart
Builds on agents can usually continue.

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.

Not secure 98.80.223.40:8080/manage/configureTools/

Dashboard > Manage Jenkins > Tools

SonarQube Scanner installations

SonarQube Scanner installations ^ Edited

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

The screenshot shows the Jenkins configuration interface for SonarQube Scanner installations. A new instance is being created with the name 'sonarqube'. The 'Install automatically' option is selected. The 'Install from Maven Central' section specifies the version 'SonarQube Scanner 6.2.1.4610'. There are 'Save' and 'Apply' buttons at the bottom.

Sonarqube installation

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

Step 1:

Click on the SonarQubeServer and click on connect.

AWS Services Search [Alt+S] N. Virginia vclabs/user3413602=MHAPANKAR_ANUPRITA_ANAND @ 8567-4606-9793 ▾

EC2 Dashboard EC2 Global View Events Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations New Images AMIs AMI Catalog Elastic Block Store Volumes Snapshots Lifecycle Manager Network & Security Security Groups Elastic IPs Placement Groups CloudShell Feedback

Instances (1/2) Info Last updated 15 minutes ago Connect Actions ▾ Launch instances ▾

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
JenkinsServer	i-05f12961a9cf8cf3e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-98-80-
SonarQubeSer...	i-061f29e11e3fa1a8b	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c	ec2-54-21-

i-061f29e11e3fa1a8b (SonarQubeServer)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID: i-061f29e11e3fa1a8b (SonarQubeServer)	Public IPv4 address: 54.210.13.192 open address	Private IPv4 addresses: 172.31.92.157
IPv6 address: -	Instance state: Running	Public IPv4 DNS: ec2-54-210-13-192.compute-1.amazonaws.com open address
Hostname type: IP name: ip-172-31-92-157.ec2.internal	Private IP DNS name (IPv4 only): ip-172-31-92-157.ec2.internal	Elastic IP addresses:
Answer private resource DNS name	Instance type	

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS EC2 Instances page with the instance `i-061f29e11e3fa1a8b` selected. The top navigation bar includes the AWS logo, services dropdown, search bar, and account information. Below the navigation is a breadcrumb trail: [EC2](#) > [Instances](#) > [i-061f29e11e3fa1a8b](#) > Connect to instance. The main content area is titled "Connect to instance" with an "Info" link. It instructs the user to connect to the instance using one of several methods: EC2 Instance Connect, Session Manager, SSH client (selected), or EC2 serial console. The "SSH client" section contains steps to use an SSH client, locate a private key file named `AMsonarqube.pem`, run a command to change its permissions (`chmod 400 AMsonarqube.pem`), and connect using the Public DNS (`ec2-54-210-13-192.compute-1.amazonaws.com`). An "Example:" section shows the command `ssh -i "AMsonarqube.pem" ec2-user@ec2-54-210-13-192.compute-1.amazonaws.com`. A note box states: "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." At the bottom right are "Cancel" and "Next Step" buttons.

Step 2:

Open Git Bash and go to the directory which has the Key downloaded. If you don't have the key downloaded, create a key 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 AMsonarqube.pem*

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

```
ec2-user@ip-172-31-92-157:~
```

```
User@DESKTOP-QQGK15A MINGW64 ~ (master)
$ cd Downloads

User@DESKTOP-QQGK15A MINGW64 ~/Downloads (master)
$ dir AMsonarqube.pem*
AMsonarqube.pem

User@DESKTOP-QQGK15A MINGW64 ~/Downloads (master)
$ ssh -i "AMsonarqube.pem" ec2-user@ec2-54-210-13-192.compute-1.amazonaws.com
The authenticity of host 'ec2-54-210-13-192.compute-1.amazonaws.com (54.210.13.192)' can't be established.
ED25519 key fingerprint is SHA256:1npBgiuhtDDBAlldNoAcPyZQhmxxvYIRxxskaitXGQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-210-13-192.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!
          / \
          /m/  /   Amazon Linux 2023, GA and supported until 2028-03-15.
          / \
          /m/  /   https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-92-157 ~]$ sudo wget -O /etc/yum.repos.d/sonar.repo http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo
--2024-10-19 16:56:06-- 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?viafs=1 [following]
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 0s

2024-10-19 16:56:07 (16.7 MB/s) - '/etc/yum.repos.d/sonar.repo' saved [93/93]

[ec2-user@ip-172-31-92-157 ~]$ sudo yum install sonar -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
am2-core
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      Size
=====
Installing:
  sonar                noarch    7.1-1          sonar        142 M
=====
Transaction Summary
=====
Install  1 Package
```

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-172-31-42-155:~$ ssh -i "jenkins.pem" ec2-user@ec2-54-91-87-54.compute-1.amazonaws.com
Last login: Fri Oct 18 14:57:44 2024 from 49.36.105.208
[ec2-user@ip-172-31-42-155 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-10-18 15:11:26 - https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85 [text/html]
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenkins.repo
100%[=====] 85 --.-KB/s in 0s

2024-10-18 15:11:26 (3.60 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[ec2-user@ip-172-31-42-155 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[ec2-user@ip-172-31-42-155 ~]$ yum install fontconfig java-17-openjdk
Error: You must run this command as root or with sudo/su.
[ec2-user@ip-172-31-42-155 ~]$ sudo yum install fontconfig java-17-openjdk
Dependencies resolved.
No match for argument: java-17-openjdk
Error: Unable to find a match: java-17-openjdk
[ec2-user@ip-172-31-42-155 ~]$ sudo dnf install java-17-amazon-corretto-devel
Last metadata expiration check: 0:02:40 ago on Fri Oct 18 15:12:06 2024.
Dependencies resolved.

Package                                         Architecture   Version          Repository      Size
Installing:
java-17-amazon-corretto-devel                  x86_64        1:17.0.12+7-1.amzn2023.1                               amazonlinux    142 k
Installing dependencies:
alsa-lib                                         x86_64        1:2.7.2-1.amzn2023.0.2                                amazonlinux    504 k
cairo                                           x86_64        1:17.6-2.amzn2023.0.1                                amazonlinux    684 k
dejavu-sans-fonts                             noarch       2.37-16.amzn2023.0.2                                amazonlinux    1.3 M
dejavu-sans-mono-fonts                         noarch       2.37-16.amzn2023.0.2                                amazonlinux    467 k
dejavu-serif-fonts                            noarch       2.37-16.amzn2023.0.2                                amazonlinux    1.0 M
fontconfig                                     x86_64        2.13.94-2.amzn2023.0.2                                amazonlinux    273 k
fonts-filesystem                           noarch       1:2.3.5-12.amzn2023.0.2                            amazonlinux    9.5 k
freetype                                         x86_64        2.13.94-2.amzn2023.0.1                                amazonlinux    423 k
google-noto-fonts-common                      noarch       20201206-2.amzn2023.0.2                            amazonlinux    15 k
google-noto-sans-vf-fonts                     noarch       20201206-2.amzn2023.0.2                            amazonlinux    492 k
graphite2                                       x86_64        1.3.14-7.amzn2023.0.2                                amazonlinux    97 k
harfbuzz                                         x86_64        7.0.0-2.amzn2023.0.1                                amazonlinux    868 k
java-17-amazon-corretto-headless              noarch       1:17.0.12+7-1.amzn2023.1                               amazonlinux    91 M
javapackages-file-system                       noarch       6.0.0-7.amzn2023.0.6                                amazonlinux    12 k
langpacks-core-font-en                         noarch       3.0-21.amzn2023.0.4                                amazonlinux    10 k
libX11                                         x86_64        1.8.10-2.amzn2023.0.1                                amazonlinux    659 k
libX11-common                                  noarch       1.8.10-2.amzn2023.0.1                                amazonlinux    147 k
libXau                                         x86_64        1.0.0-6.amzn2023.0.1                                amazonlinux    33 k
libXext                                         x86_64        1.1.6-1.amzn2023.0.1                                amazonlinux    4 k
libXrender                                      x86_64        0.9.11-6.amzn2023.0.1                                amazonlinux    29 k
libjpeg-turbo                                   x86_64        1.0.9-4.amzn2023.0.2                                amazonlinux    315 k
libpng-turbo                                   x86_64        2.1.4-2.amzn2023.0.5                                amazonlinux    190 k
libpng                                           x86_64        2:1.6.37-10.amzn2023.0.6                            amazonlinux    128 k

ec2-user@ip-172-31-42-155:~$
```

```
Transaction Summary
Install 27 Packages

Total download size: 100 M
Installed size: 261 M
Is this ok [y/N]: y
Downloading Packages:
(1/27): cairo-1.17.6-2.amzn2023.0.1.x86_64.rpm
(2/27): dejavu-sans-fonts-2.37-16.amzn2023.0.2.noarch.rpm
(3/27): alsa-lib-1.2.7-2.1.amzn2023.0.2.x86_64.rpm
(4/27): dejavu-serif-fonts-2.37-16.amzn2023.0.2.noarch.rpm
(5/27): fonts-filesystem-2.0-2.amzn2023.0.2.noarch.rpm
(6/27): google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch.rpm
(7/27): fontconfig-2.13.94-2.amzn2023.0.2.x86_64.rpm
(8/27): google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch.rpm
(9/27): graphite2-1.3.14-7.amzn2023.0.2.x86_64.rpm
(10/27): freetype-2.13.2-5.amzn2023.0.1.x86_64.rpm
(11/27): harfbuzz-7.0-0.2.amzn2023.0.1.x86_64.rpm
(12/27): google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch.rpm
(13/27): java-17-amazon-corretto-devel-17.0.12+7-1.amzn2023.1.x86_64.rpm
(14/27): javapackages-file-system-6.0-0.7.amzn2023.0.6.noarch.rpm
(15/27): langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch.rpm
(16/27): libX11-1.8.10-2.amzn2023.0.1.x86_64.rpm
(17/27): libX11-common-1.8.10-2.amzn2023.0.1.noarch.rpm
(18/27): libXau-1.0.11-6.amzn2023.0.1.x86_64.rpm
(19/27): libXext-1.3.6-1.amzn2023.0.1.x86_64.rpm
(20/27): libXrender-0.9.11-6.amzn2023.0.1.x86_64.rpm
(21/27): libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64.rpm
(22/27): libpng-tl-1.0.9-4.amzn2023.0.2.x86_64.rpm
(23/27): libpng-1.6.37-10.amzn2023.0.6.x86_64.rpm
(24/27): libxcb-1.17.0-1.amzn2023.0.1.x86_64.rpm
(25/27): xml-common-0.6.3-56.amzn2023.0.2.noarch.rpm
(26/27): pixman-0.43.4-1.amzn2023.0.4.x86_64.rpm
(27/27): java-17-amazon-corretto-headless-17.0.12+7-1.amzn2023.1.x86_64.rpm

Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch
Installing : dejavu-sans-mono-fonts-2.37-16.amzn2023.0.2.noarch
Installing : dejavu-sans-fonts-2.37-16.amzn2023.0.2.noarch
Installing : dejavu-serif-fonts-2.37-16.amzn2023.0.2.noarch
Running scriptlet: xml-common-0.6.3-56.amzn2023.0.2.noarch
Installing : xml-common-0.6.3-56.amzn2023.0.2.noarch
Installing : pixman-0.43.4-1.amzn2023.0.4.x86_64
Installing : libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64
Installing : libbrotli-1.0.9-4.amzn2023.0.2.x86_64
Installing : libxa-1.0.11-6.amzn2023.0.1.x86_64
Installing : libxcb-1.17.0-1.amzn2023.0.1.x86_64
Installing : libX11-1.8.10-2.amzn2023.0.1.noarch
Installing : libX11-common-1.8.10-2.amzn2023.0.1.noarch
Installing : libXext-1.3.6-1.amzn2023.0.1.x86_64
Installing : libXrender-0.9.11-6.amzn2023.0.1.x86_64
Installing : javapackages-file-system-6.0-0.7.amzn2023.0.6.noarch
Installing : graphite2-1.3.14-7.amzn2023.0.2.x86_64
Installing : google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch
Installing : google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch
Installing : langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch

54 MB/s | 100 MB 00:01

ec2-user@ip-172-31-42-155:~$
```

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
```

```
ec2-user@ip-172-31-92-157:~
```

```
User@DESKTOP-0OGK15A MINGW64 ~ (master)
$ cd Downloads
User@DESKTOP-0OGK15A MINGW64 ~/Downloads (master)
$ dir AMsonarqube.pem*
AMsonarqube.pem

User@DESKTOP-0OGK15A MINGW64 ~/Downloads (master)
$ ssh -i "AMsonarqube.pem" ec2-user@ec2-54-210-13-192.compute-1.amazonaws.com
The authenticity of host 'ec2-54-210-13-192.compute-1.amazonaws.com (54.210.13.192)' can't be established.
ED25519 key fingerprint is SHA256:Inp8qjisuHtD8AUdnOaCPyzQhmxxvwYHxxXisKaItxQQ.
This key is not known by any other name.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added 'ec2-54-210-13-192.compute-1.amazonaws.com (ED25519)' to the list of known hosts.
#
~\   #####      Amazon linux 2
~~ \#####\
~~ \##\      AL2 End of Life is 2025-06-30.
~~ \#\
~~ \V-->      A newer version of Amazon Linux is available!
~~ \_ /      Amazon Linux 2023, GA and supported until 2028-03-15.
~/m/      https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-92-157 ~]$ sudo wget -O /etc/yum.repos.d/sonar.repo http://downloads.sourceforge.net/project/sonar-pkg/rpm/sonar.repo
--2024-10-19 16:56:06-- 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?visaf=1
[2024-10-19 16:56:06] [downloaded 0.00B]
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 0s

2024-10-19 16:56:07 (16.7 MB/s) - '/etc/yum.repos.d/sonar.repo' saved [93/93]

[ec2-user@ip-172-31-92-157 ~]$ sudo yum install sonar -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
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        Size
=====
Installing:
sonar             noarch    7.1-1              sonar           142 M

Transaction Summary
Install 1 Package
```

Step 5:

Now, run the following commands to install the sonarqube

sudo su

```
cd /opt
||

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
```

```
MINGW64:/c/Users/User/Downloads
inflate: sonarqube-10.7.0.96327/web/WEB-INF/web.xml
creating: sonarqube-10.7.0.96327/lib/jdbc/
creating: sonarqube-10.7.0.96327/lib/jdbc/mssql/
inflate: sonarqube-10.7.0.96327/lib/jdbc/mssql/msql-jdbc-12.6.3.jrell.jar
creating: sonarqube-10.7.0.96327/lib/jdbc/postgresql/
inflate: sonarqube-10.7.0.96327/lib/jdbc/postgresql/postgresql-42.7.3.jar
creating: sonarqube-10.7.0.96327/lib/jdbc/h2/
inflate: sonarqube-10.7.0.96327/lib/jdbc/h2/h2-2.2.224.jar
inflate: sonarqube-10.7.0.96327/lib/sonar-shutdowner-10.7.0.96327.jar
creating: sonarqube-10.7.0.96327/elasticsearch/plugins/
[root@ip-172-31-82-31 opt]# ll
total 741384
drwxr-xr-x 4 root root 33 Oct 10 21:23 aws
drwxr-xr-x 11 sonar sonar 141 Sep 27 15:13 bin
drwxr-xr-x 12 root root 184 Sep 27 15:16 sonarqube-10.7.0.96327
-rw-r--r-- 1 root root 75917572 Sep 27 15:52 sonarqube-10.7.0.96327.zip
[root@ip-172-31-82-31 opt]# cd sonarqube-10.7.0.96327
[root@ip-172-31-82-31 sonarqube-10.7.0.96327]# ll
total 112
-rw-r--r-- 1 root root 7651 Sep 27 15:13 COPYING
drwxr-xr-x 6 root root 117 Sep 27 15:13 bin
drwxr-xr-x 2 root root 30 Sep 27 15:13 conf
drwxr-xr-x 2 root root 24 Sep 27 15:13 data
-rw-r--r-- 1 root root 73376 Sep 27 15:14 dependency-license.json
drwxr-xr-x 2 root root 132 Sep 27 15:16 elasticsearch
drwxr-xr-x 4 root root 40 Sep 27 15:13 extensions
drwxr-xr-x 2 root root 16384 Sep 27 15:16 logs
drwxr-xr-x 5 root root 134 Sep 27 15:16 temp
drwxr-xr-x 6 root root 16384 Sep 27 15:16 web
[root@ip-172-31-82-31 sonarqube-10.7.0.96327]# cd conf
[root@ip-172-31-82-31 conf]# ll
total 24
-rw-r--r-- 1 root root 21761 Sep 27 15:13 sonar.properties
[root@ip-172-31-82-31 conf]# cat sonar.properties
-----
# IMPORTANT:
# This file will *not* be reloaded when using the api/system/restart endpoint.
# In order for any change made to this file to be taken into account, you must perform a full
# restart of the main SonarQube service.
-----

# Property values can:
# - be overridden by environment variables. The name of the corresponding environment variable is the
# upper-cased name of the property where all the dot ('.') and dash ('-') characters are replaced by
# underscores ('_'). For example, to override 'sonar.web.systemPasscode' use 'SONAR_WEB_SYSTEMPASSCODE'.
# - be encrypted. See https://docs.sonarsource.com/sonarqube/latest/instance-administration/security/#settings-encryption
-----
# DATABASE
# IMPORTANT:
# - The embedded H2 database is used by default. It is recommended for tests but not for
# production use. Supported databases are Oracle, PostgreSQL and Microsoft SQLServer.
# - Changes to database connection URL (sonar.jdbc.url) can affect SonarSource licensed products.

# User credentials.
# Permissions to create tables, indices and triggers must be granted to JDBC user.
# The schema must be created first.
#sonar.jdbc.username=
#sonar.jdbc.password=

#----- Embedded Database (default)
# H2 embedded database server listening port, defaults to 9092
```

```
MINGW64:/c/Users/User/Downloads
----- Oracle 19c/21c
# The Oracle JDBC driver must be copied into the directory extensions/jdbc-driver/oracle/.
# Only the thin client is supported, and we recommend using the latest Oracle JDBC driver. See
# https://jira.sonarsource.com/browse/SONAR-9758 for more details.
# If you need to set the schema, please refer to http://jira.sonarsource.com/browse/SONAR-5000
#sonar.jdbc.url=jdbc:oracle:thin:@localhost:1521/XE

----- PostgreSQL 11 or greater
# By default the schema named "public" is used. It can be overridden with the parameter "currentSchema".
#sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube?currentSchema=my_schema

----- Microsoft SQLServer 2014/2016/2017/2019/2022 and SQL Azure
# A database named sonar must exist and its collation must be case-sensitive (CS) and accent-sensitive (AS)
# Use the following connection string if you want to use integrated security with Microsoft Sql Server
# Do not set sonar.jdbc.username or sonar.jdbc.password property if you are using Integrated Security
# For Integrated Security to work, you have to install the Microsoft SQL JDBC Auth package
# Please refer to the online documentation https://docs.sonarsource.com/sonarqube
# for the exact procedure for this version of SonarQube.
#sonar.jdbc.url=jdbc:sqlServer://localhost;databaseName=sonar;integratedSecurity=true

# Use the following connection string if you want to use SOL Auth while connecting to MS Sql Server.
# Set the sonar.jdbc.username and sonar.jdbc.password appropriately.
#sonar.jdbc.url=jdbc:sqlServer://localhost;databaseName=sonar

#----- Connection pool settings
# The maximum number of active connections that can be allocated
# at the same time, or negative for no limit.
# The recommended value is 1.2 x max sizes of HTTP pools. For example if HTTP ports are
# enabled with max sizes (50, see property sonar.web.http.maxThreads)
[root@ip-172-31-82-31 opt]# cd ..
[root@ip-172-31-82-31 sonarqube-10.7.0.96327]# ll
total 112
-rw-r--r-- 1 root root 7651 Sep 27 15:13 COPYING
drwxr-xr-x 6 root root 117 Sep 27 15:13 bin
drwxr-xr-x 2 root root 30 Sep 27 15:13 conf
drwxr-xr-x 2 root root 24 Sep 27 15:13 data
-rw-r--r-- 1 root root 73376 Sep 27 15:14 dependency-license.json
drwxr-xr-x 7 root root 132 Sep 27 15:16 elasticsearch
drwxr-xr-x 2 root root 40 Sep 27 15:13 extensions
drwxr-xr-x 2 root root 16384 Sep 27 15:16 logs
drwxr-xr-x 5 root root 134 Sep 27 15:16 temp
drwxr-xr-x 2 root root 24 Sep 27 15:13 logs
drwxr-xr-x 2 root root 24 Sep 27 15:13 temp
drwxr-xr-x 6 root root 16384 Sep 27 15:16 web
[root@ip-172-31-82-31 sonarqube-10.7.0.96327]# cd bin
[root@ip-172-31-82-31 bin]# ll
total 4
-rwxr-xr-x 1 root root 101 Jun 10 23:34 elasticsearch
drwxr-xr-x 2 root root 22 Sep 27 15:13 linux-x86-64
drwxr-xr-x 3 root root 22 Sep 27 15:13 macosx-universal-64
drwxr-xr-x 3 root root 23 Sep 27 15:13 windows-x86-64
drwxr-xr-x 2 root root 25 Sep 27 15:13 winsw-license
[root@ip-172-31-82-31 bin]# cd linux-x86-64
[root@ip-172-31-82-31 linux-x86-64]# ll
total 8
-rwxr-xr-x 1 root root 7192 Sep 27 15:13 sonar.sh
[root@ip-172-31-82-31 linux-x86-64]# ./sonar.sh
/usr/bin/java
Usage: ./sonar.sh { console | start | stop | force-stop | restart | status | dump }
[root@ip-172-31-82-31 linux-x86-64]# ./sonar.sh start
/usr/bin/java
Starting sonarqube...
```

```

MINGW64:/c/Users/User/Downloads
[root@ip-172-31-82-31 bin]# cd linux-x86-64
[root@ip-172-31-82-31 linux-x86-64]# ll
total 8
-rw-r--r-x. 1 root root 7192 Sep 27 15:13 sonar.sh
[root@ip-172-31-82-31 linux-x86-64]# ./sonar.sh
/usr/bin/java
Usage: ./sonar.sh { console | start | stop | force-stop | restart | status | dump }
[root@ip-172-31-82-31 linux-x86-64]# ./sonar.sh start
/usr/bin/java
Starting SonarQube...
Started SonarQube.
[root@ip-172-31-82-31 linux-x86-64]# ./sonar.sh status
/usr/bin/java
Removed stale pid file: ./SonarQube.pid
SonarQube is not running.
[root@ip-172-31-82-31 logs]# cat sonar.log
2024.10.18 19:13:39 INFO app[]{o.s.a.AppFilesystem} Cleaning or creating temp directory /opt/sonarqube-10.7.0.96327/temp
2024.10.18 19:13:39 INFO app[]{o.s.a.es.EsSettings} Elasticsearch listening on [HTTP: 127.0.0.1:9001, TCP: 127.0.0.1:45529]
2024.10.18 19:13:39 INFO ::UserSerialGC -Dcli.name=server -Dcli.script=/bin/elasticsearch -Dcli.lib$lib/tools/server-cli -Des.path.home=/opt/sonarqube-10.7.0.96327/elasticsearch -Des.path.conf=/opt/sonarqube-10.7.0.96327/elasticsearch/lib/cli-launcher/* org.elasticsearch.launcher.CliToolLauncher
2024.10.18 19:13:39 INFO ::UserSerialGC -Dcli.name=server -Dcli.script=/bin/elasticsearch -Dcli.lib$lib/tools/server-cli -Des.path.home=/opt/sonarqube-10.7.0.96327/elasticsearch -Des.path.conf=/opt/sonarqube-10.7.0.96327/elasticsearch/lib/cli-launcher/* org.elasticsearch.launcher.CliToolLauncher
2024.10.18 19:13:43 WARN app[]{o.s.a.s.SchedulerImpl} Waiting for Elasticsearch to be up and running
2024.10.18 19:13:43 INFO app[]{o.s.a.s.SchedulerImpl} Process exited with exit value [ElasticSearch]: 1
2024.10.18 19:13:43 INFO app[]{o.s.a.s.SchedulerImpl} Process[ElasticSearch] is stopped
2024.10.18 19:13:43 INFO app[]{o.s.a.s.SchedulerImpl} Process[ElasticSearch] is stopped
[root@ip-172-31-82-31 logs]# cat /opt/sonarqube-10.7.0.96327/logs/es.log
2024.10.18 19:13:43 INFO es[]{o.e.n.NativeEsCase} Using [jna] native provider and native methods for [Linux]
2024.10.18 19:13:43 ERROR es[]{o.e.b.Elasticsearch} fatal exception while booting Elasticsearch
java.lang.RuntimeException: can not run elasticsearch as root
    at org.elasticsearch.bootstrap.Bootstrap$Elasticsearch.initializeNatives(Bootstrap.java:286) ~[elasticsearch-8.14.1.jar:?:~]
    at org.elasticsearch.bootstrap.Bootstrap$Elasticsearch.initPhase2(Bootstrap.java:169) ~[elasticsearch-8.14.1.jar:?:~]
    at org.elasticsearch.bootstrap.Bootstrap$Elasticsearch.main(Bootstrap.java:74) ~[elasticsearch-8.14.1.jar:?:~]
[root@ip-172-31-82-31 logs]# cd ..
[root@ip-172-31-82-31 sonarqube-10.7.0.96327]# cd ..
[root@ip-172-31-82-31 opt]# cd ..
[root@ip-172-31-82-31 ~]# /etc/init.d/sonar start
adduser: user 'sonar' already exists
[root@ip-172-31-82-31 ~]# sudo passwd sonar
Changing password for user sonar.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
[root@ip-172-31-82-31 ~]# sudo passwd sonar
Changing password for user sonar.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-82-31 ~]# sudo chown -R sonar:sonar /opt/sonarqube-10.7.0.96327
[root@ip-172-31-82-31 ~]# su - sonar
Last login: Fri Oct 18 18:26:31 UTC 2024 on pts/1
[sonar@ip-172-31-82-31 ~]$ cd /opt/sonarqube-10.7.0.96327/bin/linux-x86-64
[sonar@ip-172-31-82-31 Linux-x86-64]$ ./sonar.sh start
/usr/bin/java
Starting SonarQube...
Started SonarQube.
[sonar@ip-172-31-82-31 Linux-x86-64]$ ./sonar.sh status
/usr/bin/java
SonarQube is running (29499).
[sonar@ip-172-31-82-31 Linux-x86-64]$ client_loop: send disconnect: Connection reset by peer
User@DESKTOP-QOGK15A MINGW64 ~/Downloads (master)
$ ...

```

Step 6:

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

Instances (1/2) Info							
Last updated less than a minute ago							
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
JenkinsServer	i-05f12961a9cf8cf3e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-98-80-11-111.us-east-1.compute.amazonaws.com
SonarQubeServer	i-061f29e11e3fa1a8b	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c	ec2-54-210-13-192.us-east-1.compute.amazonaws.com

i-061f29e11e3fa1a8b (SonarQubeServer)

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

Instance summary

- Instance ID: i-061f29e11e3fa1a8b (SonarQubeServer)
- IPv6 address: -
- Hostname type: IP name: ip-172-31-92-157.ec2.internal
- Answer private resource DNS name: -
- Private IP DNS name (IPv4 only): ip-172-31-92-157.ec2.internal
- Instance type: t2.medium

Public IPv4 address copied

54.210.13.192 | [open address](#)

Private IP addresses

172.31.92.157

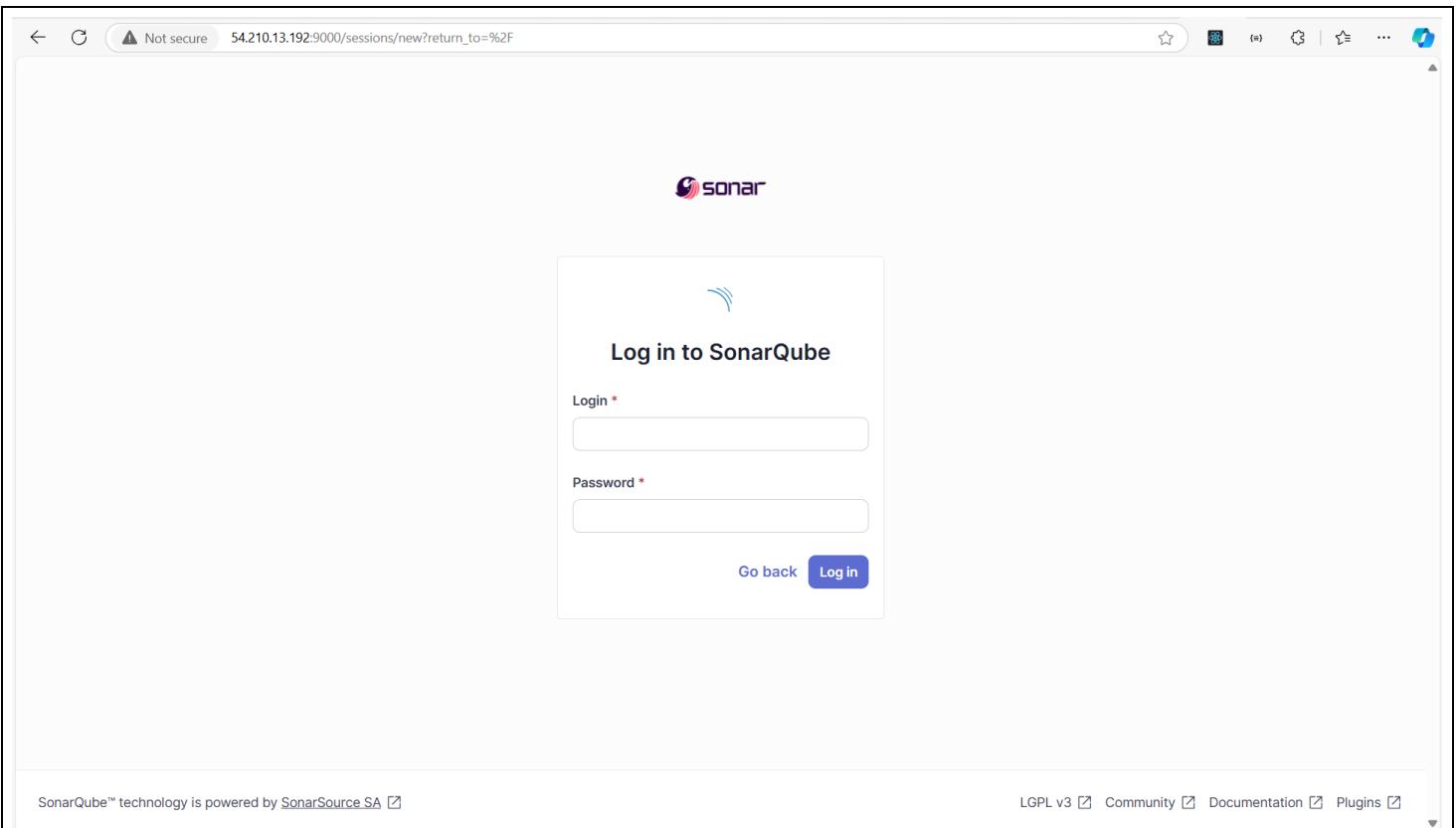
Public IPv4 DNS

ec2-54-210-13-192.compute-1.amazonaws.com | [open address](#)

Elastic IP addresses

Step 7:

You will be redirected to this page on successful installation of SonarQube and visiting the public address url with port 9000. Login the username=admin and password=admin.



Step 8:

Now, set up the initial configurations by setting up new password.
password=Anuprita@4321

A screenshot of a web browser showing the "Update your password" page. The title bar indicates the URL is 54.210.13.192:9000/account/reset_password. The page has a header "Update your password" with a warning message: "⚠ This account should not use the default password." Below this is a section titled "Enter a new password" with the note "All fields marked with * are required". It contains three input fields: "Old Password *", "Password *", and "Confirm Password *". Each field has a green border and a small checkmark icon. At the bottom is a blue "Update" button.

Step 9:

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

Not secure 54.210.13.192:9000/projects/create

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More ? A

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.

Import from Azure DevOps Setup

Import from Bitbucket Cloud Setup

Import from Bitbucket Server Setup

Import from GitHub Setup

Import from GitLab Setup

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

Create a local project

⚠️ **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

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

Not secure 54.210.13.192:9000/projects/create?mode=manual&setncd=true

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

2 of 2 Set up project for Clean as You Code

The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

Choose the baseline for new code for this project

Use the global setting

Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Define a specific setting for this project

Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.
Recommended for projects following continuous delivery.

Reference branch
Choose a branch as the baseline for the new code.

Not secure 54.210.13.192:9000/tutorials?id=Hello-World

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

Hello World / main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Analysis Method

Use this page to manage and set-up the way your analyses are performed.

How do you want to analyze your repository?

With Jenkins

With GitHub Actions

With Bitbucket Pipelines

With GitLab CI

With Azure Pipelines

Other CI
SonarQube integrates with your workflow no matter which CI tool you're using.

Locally
Use this for testing or advanced use-case. Other modes are recommended to help you set up your CI environment.

⚠️ 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

Edit global se Global Permissions

Find in Permission Templates

Analysis Scope Authentication DevOps Platform Integrations Email Notification External Analyzers General Housekeeping JaCoCo Languages

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

Not secure | 54.210.13.192:9000/admin/users

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

Administration

Configuration Security

Users

Create and administer individual users

Search by login or name

Name	Type	Project	Last use	Created	Expiration	Actions
A Administrator admin	User		Never	October 19, 2024	November 18, 2024	Revoke

Tokens of Administrator

Generate Tokens

Name: Enter Token Name Expires in: 30 days [Generate](#)

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

squ_c395eea8275923706da6ddb29f77d673a9a51668 [Copy](#)

Name	Type	Project	Last use	Created	Expiration	Actions
admin-token	User		Never	October 19, 2024	November 18, 2024	Revoke

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](#)

GitHub

Step 1:

Make a repository and upload your files in the repository.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** AdvDevOps_Practical
- Explorer:** Shows a tree view of the project structure:
 - ADVDEVOPS_PRactical
 - .terraform
 - .terraform.lock.hcl
 - hello.py
 - main.tf
 - terraform.tfstate
- Code Editor:** A file named "hello.py" is open, containing the code:

```
1 print("Hello, World!")
```
- Bottom Status Bar:** Ln 1, Col 23, Spaces: 4, UTF-8, CRLF, Python 3.12.0 64-bit, Go Live, Prettier.

The screenshot shows a GitHub repository page for "helloworld_python".

Repository Details:

- Name:** helloworld_python
- Owner:** Anuprita2022-26
- Branch:** main
- Last Commit:** 28b08a8 · yesterday
- Commits:** 1 Commit
- Activity:** 0 stars, 1 watching, 0 forks

Files:

- hello.py

README:

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

About:

No description, website, or topics provided.

Releases:

No releases published

[Create a new release](#)

Packages:

No packages published

[Publish your first package](#)

Languages:

Python 100.0%

Suggested workflows:

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

```
MINGW64:/c/Users/User/Downloads
$ ssh -i "Mjenkins.pem" ec2-user@ec2-98-80-223-40.compute-1.amazonaws.com
Last login: Sat Oct 19 17:56:46 2024 from 49.36.105.96
      #
      #### 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/
[ec2-user@ip-172-31-42-86 ~]$ sudo yum install git
Loaded plugins: extras_suggestions, Tangacks, 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(Error) 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 | size |
|=====             | =====| =====          | =====       | ===== |
| Installing:      |       |               |            |       |
| git              | x86_64 | 2.40.1-1.amzn2.0.3 | amzn2-core | 54 k |
|=====             |       |               |            |       |
| Installing for dependencies: |       |               |            |       |
| git-core          | x86_64 | 2.40.1-1.amzn2.0.3 | amzn2-core | 10 M |
| git-core-doc     | noarch | 2.40.1-1.amzn2.0.3 | amzn2-core | 3.0 M |
| perl-Error       | noarch | 1:0.17020-2.amzn2 | amzn2-core | 32 k |
| perl-Git          | noarch | 2.40.1-1.amzn2.0.3 | amzn2-core | 42 k |
| perl-TermReadkey | x86_64 | 2.30-20.amzn2.0.2 | amzn2-core | 31 k |
|=====             |       |               |            |       |

Transaction Summary
=====
| Install 1 Package (+5 Dependent packages) |       |               |            |       |
|=====             |       |               |            |       |
Total download size: 13 M
installed size: 44 M
Is this ok [y/d/N]: y
Downloading packages:
(1/6): git.x86_64 0:2.40.1-1.amzn2.0.3.x86_64.rpm | 54 kB 00:00:00
(2/6): git-core-doc-2.40.1-1.amzn2.0.3.noarch.rpm | 3.0 MB 00:00:00
(3/6): perl-Error-0.17020-2.amzn2.noarch.rpm | 32 kB 00:00:00
(4/6): git-core-2.40.1-1.amzn2.0.3.x86_64.rpm | 10 MB 00:00:00
(5/6): perl-Git-2.40.1-1.amzn2.0.3.noarch.rpm | 42 kB 00:00:00
(6/6): perl-TermReadkey-2.30-20.amzn2.0.2.x86_64.rpm | 31 kB 00:00:00
```

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 >

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

Environment variables

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**

Not secure 98.80.223.40:8080/manage/configure

Dashboard > Manage Jenkins > System >

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

Server authentication token: SonarQube authentication token. Mandatory when anonymous access is disabled.

+ Add

Advanced ▾

Add SonarQube

Save **Apply**

Step 3:

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

Not secure | 98.80.223.40:8080/manage/credentials/

Credentials

T	P	Store ↓	Domain	ID	Name
File icon	User icon	System	(global)	6e0ad648-6931-48d0-a2eb-938a55db6234	Secret text

Stores scoped to Jenkins

P	Store ↓	Domains
User icon	System	(global)

Icon: S M L

REST API Jenkins 2.46.3

Step 4:

Create a new pipeline and name it pipeline1.

Not secure | 98.80.223.40:8080/view/all/newJob

New Item

Enter an item name

Select an item 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.

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.

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 folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

OK

Step 5:

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

Not secure | 98.80.223.40:8080/job/pipeline1/configure

Jenkins

Dashboard > pipeline1 > Configuration

Configure General

Enabled

General

Advanced Project Options

Pipeline

Description

Plain text [Preview](#)

Discard old builds [?](#)

Do not allow concurrent builds

Do not allow the pipeline to resume if the controller restarts

GitHub project

Project url [?](#)

https://github.com/Anuprita2022-26/helloworld_python/

Advanced [▼](#)

Pipeline speed/durability override [?](#)

[Save](#) [Apply](#)

Step 6:

Now, write the following Pipeline Script.

```
pipeline {  
    agent any  
    stages {  
        stage('Clone Repository') {  
            steps {  
                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 "${scannerHome}/bin/sonar-scanner -Dsonar.projectKey=Hello-World -Dsonar.sources=. -Dsonar.login=$SONAR_TOKEN"  
                    }  
                }  
            }  
        }  
    }  
}
```

```

}

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

The screenshot shows the Jenkins Pipeline configuration page for a pipeline named 'pipeline1'. The pipeline script is defined as follows:

```

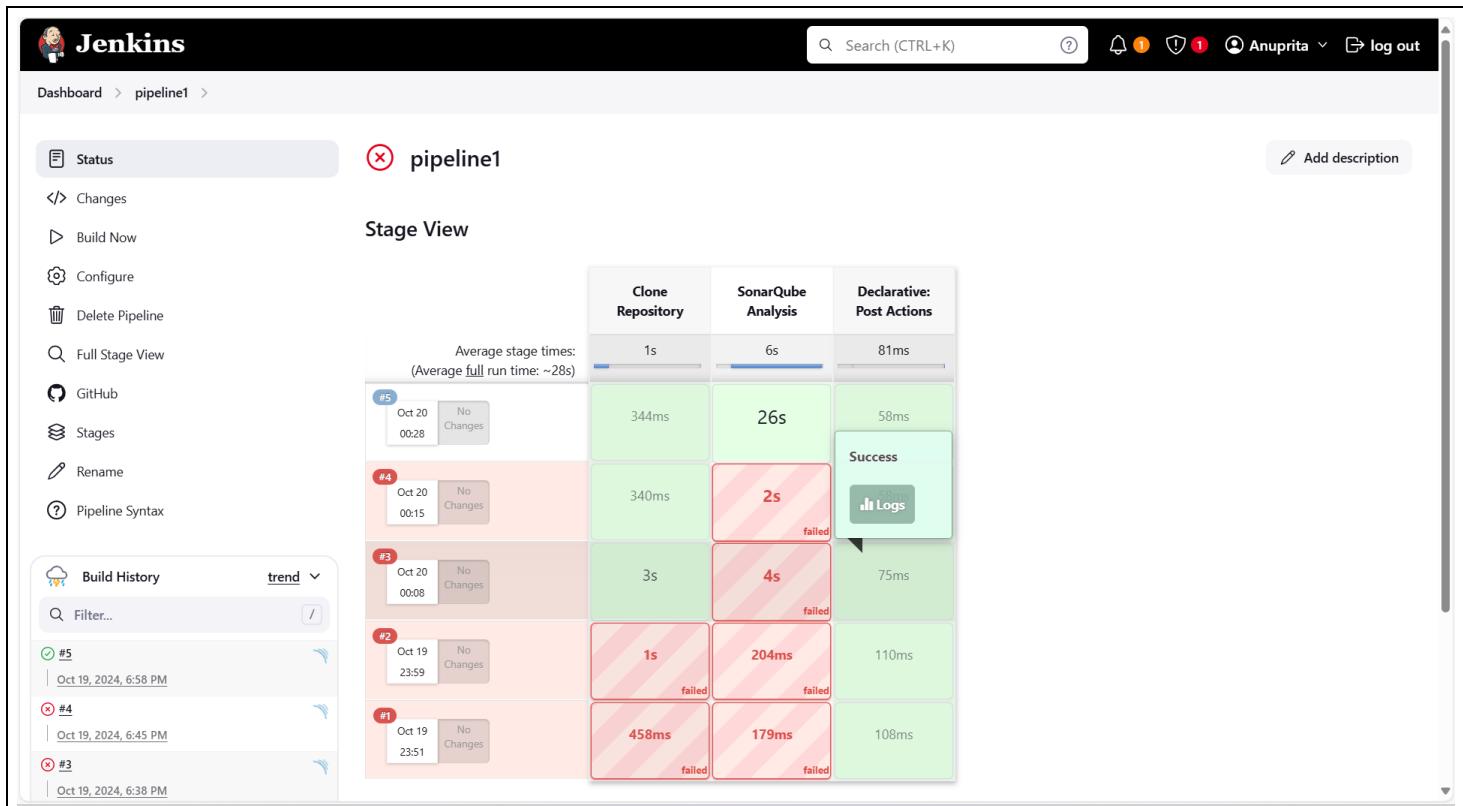
1 pipeline {
2     agent any
3     stages {
4         stage('Clone Repository') {
5             steps {
6                 git branch: 'main', url: 'https://github.com/Anuprita2022-26/helloworld_python.git'
7             }
8         }
9         stage('SonarCube Analysis') {
10            environment {
11                scannerHome = tool 'SonarCubeScanner' // Ensure SonarCube Scanner is installed
12            }
13            steps {
14                withSonarQubeEnv('SonarCube') { // Name of SonarCube server configured in Jenkins
15                    withCredentials([string(credentialsId: '6e0ad648-6931-48d0-a2eb-938a55db6234', variable: 'SONAR_TOKEN')]) {
16                        sh "${scannerHome}/bin/sonar-scanner -Dsonar.projectKey=Hello-World -Dsonar.sources=. -Dsonar.login=$SONAR_TOKEN"
17                    }
18                }
19            }
20        }
21    }
22}
```

The 'Use Groovy Sandbox' checkbox is checked.

At the bottom of the screen, there are 'Save' and 'Apply' buttons.

Step 7:

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



Not secure | 98.80.223.40:8080/job/pipeline1/5/console

Jenkins

Dashboard > pipeline1 > #5

Status **Changes** **Console Output** **Edit Build Information** **Delete build '#5'** **Timings** **Git Build Data** **Pipeline Overview** **Pipeline Console** **Restart from Stage** **Replay** **Pipeline Steps** **Workspaces** **Previous Build**

Console Output

Started by user Anuprita
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/pipeline1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone Repository)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/pipeline1/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Anuprita2022-26/helloworld_python.git # timeout=10
Fetching upstream changes from https://github.com/Anuprita2022-26/helloworld_python.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/Anuprita2022-26/helloworld_python.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 28b08a8b749d48fe74988ca608ac5cde40bd23bd (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 28b08a8b749d48fe74988ca608ac5cde40bd23bd # timeout=10
> git branch -v --no-abbrev # timeout=10
> git branch -D main # timeout=10
> git checkout -b main 28b08a8b749d48fe74988ca608ac5cde40bd23bd # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk 28b08a8b749d48fe74988ca608ac5cde40bd23bd # timeout=10
[Pipeline] }
[Pipeline] // stage

Not secure | 98.80.223.40:8080/job/pipeline1/5/console

Dashboard > pipeline1 > #5

```

18:59:20.46 INFO CPU Executor CPU calculation finished (done) | time=0ms
18:59:20.483 INFO SCM revision ID '28b08a8b749d48fe74988ca608ac5cde40bd23bd'
18:59:20.715 INFO Analysis report generated in 223ms, dir size=221.6 kB
18:59:20.766 INFO Analysis report compressed in 47ms, zip size=22.5 kB
18:59:20.823 INFO Analysis report uploaded in 53ms
18:59:20.825 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://54.210.13.192:9000/dashboard?id=Hello-World
18:59:20.826 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
18:59:20.826 INFO More about the report processing at http://54.210.13.192:9000/api/ce/task?id=6d8de4d6-21ba-4b74-922f-410cd630e8
18:59:20.852 INFO Analysis total time: 13.793 s
18:59:20.853 INFO SonarScanner Engine completed successfully
18:59:20.925 INFO EXECUTION SUCCESS
18:59:20.930 INFO Total time: 24.932s
[Pipeline]
[Pipeline] // withCredentials
[Pipeline]
[Pipeline] // withSonarQubeEnv
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline completed
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

REST API Jenkins 2.462.3

SonarQube Analysis and Results

Step 1:

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

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

Not secure | 54.210.13.192:9000/projects

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

[Create Project](#)

My Favorites All

Filters

Quality Gate

- Passed 1
- Failed 0

Security

- A ≥ 0 info issues 1
- B ≥ 1 minor issue 0
- C ≥ 1 major issue 0
- D ≥ 1 critical issue 0
- E ≥ 1 blocker issue 0

Reliability

- A ≥ 0 info issues 1
- B ≥ 1 minor issue 0
- C ≥ 1 major issue 0
- D ≥ 1 critical issue 0

Star Hello World PUBLIC

Last analysis: 4 minutes ago · 1 Lines of Code · Python

Passed

Security	A 0	Reliability	A 0	Maintainability	A 0	Hotspots Reviewed	A —	Coverage	0.0%	Duplications	0.0%
----------	-----	-------------	-----	-----------------	-----	-------------------	-----	----------	------	--------------	------

1 of 1 shown

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.

Guidelines for User:

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