



---

## **DevOps CI/CD Pipeline Project**

---

**CI/CD Pipeline with Jenkins using Maven, SonarQube,  
Trivy, Nexus, Docker, Kubernetes, Grafana and  
Prometheus**

PREPARED BY  
SIDNEY SMITH EBOT  
[ebotsmith@gmail.com](mailto:ebotsmith@gmail.com)

## Contents

<b>1 Introduction</b> .....	4
1.1 Real-Time Scenario:.....	4
1.2 Description: .....	4
1.3 Introduction.....	4
<b>2 Overview and Key Concepts</b> .....	5
2.1 Architecture.....	5
2.2 Git .....	6
2.3 Amazon EC2 (Elastic Compute Cloud).....	6
2.4 Jenkins .....	6
2.5 Kubernetes .....	6
2.6 Nexus Repository .....	6
2.7 SonarQube.....	6
2.8 SNS (Simple Notification Service).....	6
2.9 Prometheus.....	6
2.10 Grafana.....	6
<b>3 Phase 1: Set up Infrastructure</b> .....	7
3.1 Set up Network Environment.....	7
3.1.1 Set up Virtual Private Cloud .....	7
3.1.2 Set up Security Group .....	11
3.2 Create and Configure Virtual Machines Kubernetes.....	24
3.2.1 Create Virtual Machines for Kubernetes.....	24
3.2.2 SSH Connect to the Virtual Machines .....	33
3.3 Set up Kubernetes Cluster using kubeadm .....	59
3.3.1 Set up Kubernetes on Virtual Machine “Master” .....	59
3.3.2 Set up Kubernetes on Virtual Machine “Slave-1” .....	69
3.3.3 Set up Kubernetes on Virtual Machine “Slave-2” .....	77
3.3.4 Deploy Sample Nginx Microservice on Kubernetes .....	85
3.4 Installing Kube Audit for Trivy .....	89
3.5 Create and Configure Virtual Machine for SonarQube .....	94
3.5.1 Create Virtual Machine for SonarQube Server .....	94
3.5.2 SSH Connect to SonarQube Server.....	98
3.5.3 Configure SonarQube Server.....	106
3.5.4 Accessing SonarQube through Browser.....	112
3.5.5 Generate SonarQube Token.....	115

3.6 Create and configure Virtual Machine for Nexus.....	118
3.6.1 Create Virtual Machine for Nexus.....	118
3.6.2 SSH Connect to Nexus Server .....	122
3.6.3 Configure Nexus Server .....	130
3.6.4 Accessing Nexus through Browser .....	136
3.7 Create and Configure Virtual Machine for Jenkins .....	150
3.7.1 Create Virtual Machine for Jenkins .....	150
3.7.2 SSH Connect to Virtual Machine for Jenkins .....	154
3.7.3 Configure Jenkins .....	162
3.7.4 Accessing Jenkins through Browser .....	171
3.7.5 Install Trivy on Jenkins Server .....	176
3.7.6 Install kubectl on Jenkins Server .....	180
3.8 Create and Configure Virtual Machine for Monitoring.....	183
3.8.1 Create Virtual Machine for Monitor .....	183
3.8.2 SSH Connect to Virtual Machine for Jenkins .....	187
3.8.3 Install Prometheus .....	195
3.8.4 Access Prometheus on the browser.....	202
3.8.5 Install Grafana .....	203
3.8.6 Access Grafana on the browser .....	207
<b>4 Phase 2: Private Git Set up .....</b>	<b>211</b>
4.1 Create a Private Git Repository .....	211
4.2 Push the Source code to the repository.....	215
4.3 Create a GitHub Token .....	221
4.4 Make the Repository Visible .....	228
<b>5 Phase 3: CI/CD Pipeline .....</b>	<b>229</b>
5.1 Install the Plugins .....	229
5.1.1 Install JDK Plugins .....	229
5.1.2 Install Maven Plugins.....	234
5.1.3 Install Pipeline Stage View Plugins .....	237
5.1.4 Install SonarQube Plugins.....	239
5.1.5 Install Docker Plugins .....	242
5.1.6 Install Kubernetes Plugins .....	244
5.2 Configure the tools in Jenkins .....	247
5.2.1 Configure JDK Tool .....	247
5.2.2 Configure SonarQube Scanner Tool .....	253

5.2.3 Configure Maven Tool.....	256
5.2.4 Configure Docker Tool.....	260
5.3 Create Credentials.....	265
5.3.1 Create Credentials for Java JDK.....	265
5.3.2 Create Credentials for SonarQube .....	269
5.2.3 Configure SonarQube Server.....	273
5.3.4 Create Credentials for Nexus .....	279
5.3.5 Create Credentials for Docker.....	286
5.3.6 Create Credentials for Kubernetes.....	291
5.3.7 Create Credentials for Email Notifications.....	309
5.4 Create and configure the CI/CD Pipeline .....	318
5.4.1 Create Pipeline .....	318
5.4.2 Configure the Pipeline.....	322
<b>6 Phase 4: Monitoring .....</b>	<b>410</b>
6.1 Install Blackbox.....	410
6.1.1 Download Blackbox Exporter .....	411
6.1.2 Access Blackbox Exporter on browser .....	417
6.2 Configure Prometheus .....	418
6.3 Website Monitoring .....	430
6.3.1 Add Prometheus as data Source to Grafana .....	430
6.3.2 Import the Dashboard.....	434
<b>7 Phase 5: Automate the Process .....</b>	<b>441</b>
<b>7.1 Modification on Jenkins .....</b>	<b>441</b>
7.1.1 Add the URL of the GitHub Repository on Jenkins.....	441
7.1.2 Modify Trigger on Jenkins .....	444
7.1.3 Make Modification on GitHub.....	447
7.1.4 Verification .....	450

## 1 Introduction

### 1.1 Real-Time Scenario:

Imagine we have an application that belongs to a client. Then the client requests us to add some new features for example, the client wants us to change the background color of the application which is a new feature. The client is going to create a Jira task or Jira ticket.

That ticket will contain detail information about what changes need to be done or what features need to be added in the application. So, that ticket will be assigned to a developer.

### 1.2 Description:

The developer is going to write the source code and test the source code in his local machine. If the source code is running fine, the developer is going to push the new changes to the GitHub repository that contains the actual source code. Once those changes are pushed into the GitHub repository along with the source code. Then the DevOps engineer will start writing the Pipeline.

### 1.3 Introduction

In writing the pipeline, we are going to use Jenkins. In Jenkins, the first stage is going to be compiling of the source code. The compilation is done to find out if there are any syntax-based errors in the pipeline or not.

After that, we are going to run the unit test cases. We are going to run the test cases to test the functionality of the code.

Next, we will perform the SonarQube code quality check. Code Quality check is done to find out if there are bugs issues or code smells or vulnerability inside the source code. The lesser these things in our code, the better the code quality.

Then, we will perform the vulnerability scan on the source code to find out if there is any sensitive data using Trivy. The Trivy tool is also going to scan the dependencies of the application if they are vulnerable if they are outdated or any other issue and it is going to generate a report. We are going to generate the report in a specific format and export to a third-party file such as HTML, Text file or anything that is relevant.

Next, we are going to build or package the application to get the application artifact using Maven. We will then publish this application artifact to Nexus repository so that we can properly do release management with different versions of the artifact.

Once, we have the artifact ready, we are going to build the Docker image and tag it properly. Tagging is done basically to define different versions of the docker image.

After that, we are going to use Trivy again which is the most versatile tool in DevOps to scan the Docker image to find out vulnerability in the container.

Next, we are going to push the Docker image to the Docker Hub repository. The repository can be private or public depending on the project.

After that, we have to make sure that our Kubernetes cluster is secured. For that, we are going to use a tool known as cube audit that is going to scan the Kubernetes cluster and let us know if there are any

specific issues in the cluster. Then, we are going to deploy the application to Kubernetes cluster and then we are able to verify if the deployment is successful or not.

In the next stage, we are going to generate an email notification to inform if the pipeline is successful or not.

Finally, once the application is deployed, we are going to monitor the application. The monitoring will be done in two ways. First, we are going to do website level monitoring using Blackbox exporter to find out how is the traffic on the website if the website is up or not. And we can see the monitoring results on Grafana.

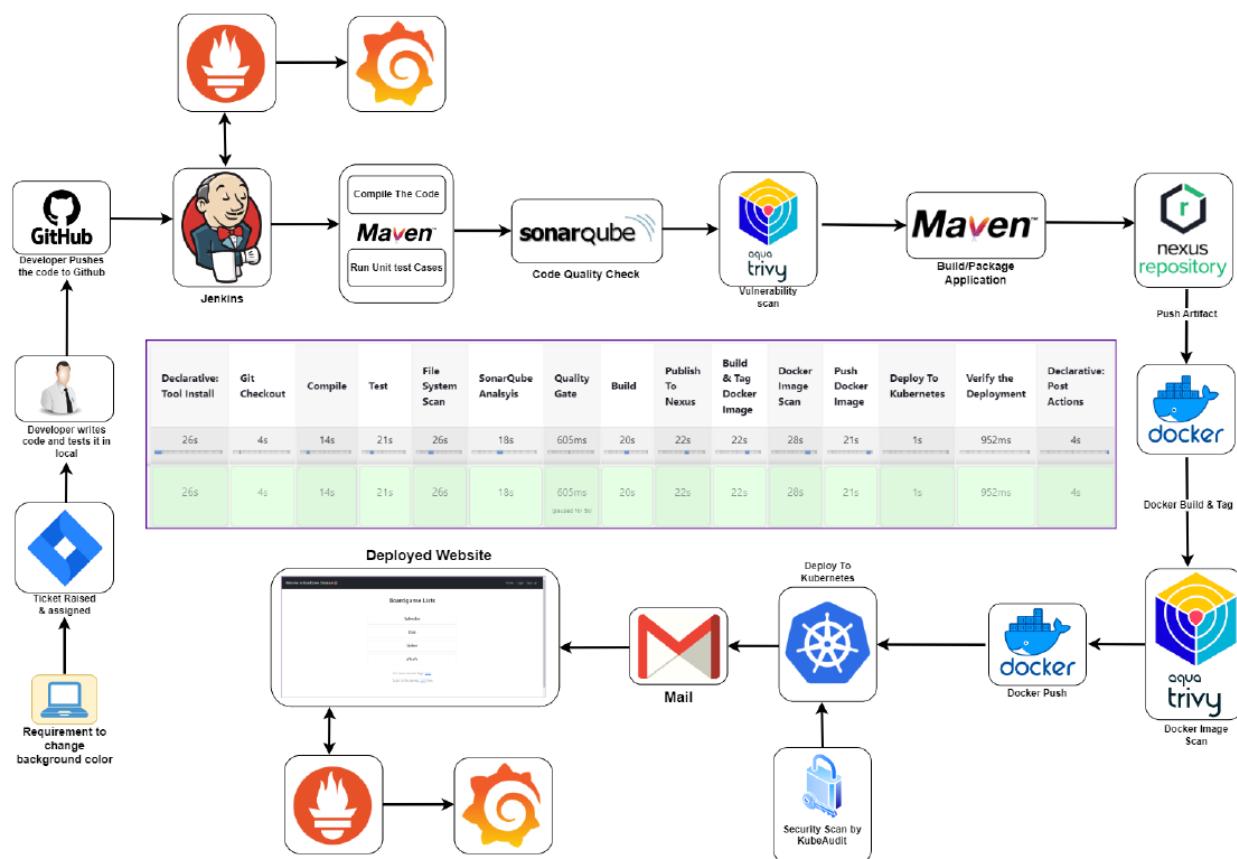
Also, in order to do the system level monitoring to find out how much CPU is being used, how much RAM is being used, we are going to monitor the Jenkins for that. For monitoring in this case, we are going to use Node Exporter.

## 2 Overview and Key Concepts

In this section, we are going to look at the architecture of this project and some important concepts.

### 2.1 Architecture

Below is the architecture for this project



## 2.2 Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

## 2.3 Amazon EC2 (Elastic Compute Cloud)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable computing capacity—literally, servers in Amazon's data centers—that you use to build and host your software systems.

## 2.4 Jenkins

Jenkins is an open-source continuous integration (CI) server. It manages and controls several stages of the software delivery process, including build, documentation, automated testing, packaging, and static code analysis. Jenkins is a highly popular DevOps tool used by thousands of development teams.

Jenkins automation is commonly triggered by code changes in repositories like GitHub, Bitbucket, and GitLab, and integrates with build tools like Maven and Gradle. Jenkins supports the use of container technologies like Docker and Kubernetes for testing and packaging of software releases, but it is neither a Kubernetes native solution nor a container-native CI solution.

## 2.5 Kubernetes

Kubernetes (often abbreviated as K8s) is an open-source platform for automating the deployment, scaling, and management of containerized applications. It acts as an orchestration system, managing the lifecycle of applications made up of many individual containers across clusters of physical or virtual machines.

## 2.6 Nexus Repository

Nexus Repository (Sonatype Nexus) is a universal artifact repository manager that centralizes the storage, management, and distribution of software components (like libraries, binaries, Docker images) for development teams, acting as a single source of truth for the entire software supply chain, integrating into DevOps pipelines for efficiency and control. It supports numerous formats (Maven, npm, Docker, etc.) and provides features for security, consistency, and faster builds by caching external dependencies and hosting internal artifacts.

## 2.7 SonarQube

SonarQube is a powerful open-source tool that helps you maintain code quality and security by analyzing your codebase for bugs and vulnerabilities. And it can play a major role when integrated into your CI/CD pipeline.

## 2.8 SNS (Simple Notification Service)

SNS is a fully managed messaging service that enables developers to send notifications to various endpoints or distributed systems

## 2.9 Prometheus

Prometheus is an open-source systems monitoring and alerting toolkit originally built at SoundCloud. Prometheus collects and stores its metrics as time series data, i.e. metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels.

## 2.10 Grafana

Grafana is an open-source analytics and visualization platform that lets you query, visualize, alert on, and understand metrics, logs, and traces from various data sources to monitor applications and infrastructure.

It creates customizable, interactive dashboards, acting like a universal control panel for your data, helping teams track performance, identify issues, and make data-driven decisions in real-time.

### 3 Phase 1: Set up Infrastructure

In this phase, we are going to set up the infrastructures which includes setting up the Networking environment and creating the virtual machines.

#### 3.1 Set up Network Environment

In this part, we are going to set up the networking environment. The reason for having separate networking environment is that all the resources we are going to work with and all the applications that we are going to deploy should be private. They should be in an isolated environment so that no outside entity should be able to access it. Thirdly, we are going to make sure that the deployments are secured.

##### 3.1.1 Set up Virtual Private Cloud

To do this, we will be creating a Virtual Private Cloud. Go to the AWS Management Console.

The screenshot shows the AWS Management Console home page. At the top, there's a search bar and a navigation menu icon. The main area has several sections: 'Recently visited' (EC2, Control Tower, AWS Organizations, IAM), 'Applications (0)', 'Welcome to AWS' (Getting started with AWS, Training and certification), 'AWS Health' (Open issues: 0, Past 7 days; Scheduled changes: 0, Upcoming and past 7 days; Other notifications: 0, Past 7 days), and 'Cost and usage' (Data unavailable). The top right corner shows account information (Account ID: 6423-9195-8541, Smith) and a region selector (United States (N. Virginia)).

Search for “VPC”

The screenshot shows the AWS VPC landing page. On the left, there's a sidebar with links like Services, Features, Resources, Documentation, and Tutorials. The main area has a "Services" section with cards for VPC (Isolated Cloud Resources), AWS Global View (provides a global dashboard and search functionality), and AWS Firewall Manager (central management of firewall rules). Below that is a "Features" section with cards for Dashboard (VPC feature), Route 53 VPCs (Route 53 feature), and VPC Reachability Analyzer (VPC feature). A modal window titled "Resources in us-east-1 / for a focused search" is open, containing a message about cross-Region search and a "Enable cross-Region search" button. The right side of the screen shows a "Create application" interface with a "Find applications" search bar and a "Create application" button.

Then, click on “VPC” under “Service”

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with sections for Virtual private cloud, Security, PrivateLink and Lattice, and Marketplace. The main area has a "Create VPC" and "Launch EC2 Instances" button. Below that is a "Resources by Region" section with a "See all regions" link. The right side features a "Service Health" section with a "View complete service health details" link, a "Settings" section with links for Block Public Access, Zones, and Console Experiments, an "Additional Information" section with links for VPC Documentation, All VPC Resources, Forums, and Report an Issue, and an "AWS Network Manager" section with a "Get started with Network Manager" link.

Click on “VPC”

Your VPCs

VPCs | VPC encryption controls

Your VPCs (1) Info

Name	VPC ID	State	Encryption c...	Encryption control...	Block Public...	IPv4 CIDR
-	vpc-0d74d5736a240e572	Available	-	-	Off	172.31.0.0/16

Select a VPC above

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

This is a default VPC. Let us create a VPC for this project. Click on “Create VPC”

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

**VPC settings**

Resources to create [Info](#)  
Creates only the VPC resource or the VPC and other networking resources.

VPC only  VPC and more

Name tag - optional [Info](#)  
Creates a tag with a key of 'Name' and a value that you specify.  
my-vpc-01

IPv4 CIDR block [Info](#)  
 IPv4 CIDR manual input  IPAM-allocated IPv4 CIDR block

IPv4 CIDR  
10.0.0.0/24  
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)  
 No IPv6 CIDR block  IPAM-allocated IPv6 CIDR block  Amazon-provided IPv6 CIDR block  IPv6 CIDR owned by me

Tenancy [Info](#)  
Default

VPC encryption control (\$) [Info](#)  
Monitor mode provides visibility into encryption status without blocking traffic. Enforce mode prevents unencrypted traffic. Additional charges apply [Learn more](#)

None  Monitor mode See which resources in your VPC are unencrypted but allow the creation of unencrypted resources.  Enforce mode Requires all resources, except exclusions, in your VPC to be encryption-capable and blocks creation of unencrypted resources.

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

We will use “VPC only” as shown above. Then we will call our VPC “**demo-vpc**”

A screenshot of the AWS VPC Create VPC wizard. The 'VPC settings' section is open, showing options for 'Resources to create' (set to 'VPC only'), a 'Name tag' ('demo-vpc'), and 'IPv4 CIDR block' ('10.0.0.0/24'). Under 'VPC encryption control', 'None' is selected. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App.

For the IPv4 CIDR, we will use “10.0.0.0/16”

A screenshot of the AWS VPC Create VPC wizard. The 'IPv4 CIDR block' is now set to '10.0.0.0/16'. In the 'Tags' section, a tag 'Name' is added with value 'demo-vpc'. An orange arrow points from the 'Create VPC' button at the bottom right towards the 'Create VPC' button in the main wizard area.

Leave all the other things as default and click on “Create VPC”

The screenshot shows the AWS VPC Details page for a VPC named 'demo-vpc'. The 'Details' tab is selected, displaying various configuration settings:

VPC ID	State	Block Public Access	DNS hostnames
vpc-03493749bc14b79b8	Available	Off	Disabled
DNS resolution	Tenancy	DHCP option set	Main route table
Enabled	default	dopt-0c4e69232fce628a4	rtb-0166757c9c0be3304
Main network ACL	Default VPC	IPv4 CIDR	IPv6 pool
acl-0643c497e4bc73c3e	No	10.0.0.0/16	-
IPv6 CIDR (Network border group)	Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID
-	Disabled	-	642391958541
Encryption control ID	Encryption control mode		
-	-		

The 'Resource map' section shows the components of the VPC:

- VPC: Your AWS virtual network (demo-vpc)
- Subnets (0): Subnets within this VPC
- Route tables (1): Route network traffic to resources (rtb-0166757c9c0be3304)
- Network Conne: Connections to other

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App.

Click on “Your VPCs”

The screenshot shows the AWS VPC dashboard under the 'Your VPCs' section. A VPC named 'demo-vpc' is listed in the table:

Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 CIDR
-	vpc-0d74d3736a240e572	Available	-	-	Off	172.31.0.0/16
demo-vpc	vpc-03493749bc14b79b8	Available	-	-	Off	10.0.0.0/16

An orange arrow points to the 'demo-vpc' row in the table.

You can see the VPC we just created. Let us now move to the next step to set up the security group.

### 3.1.2 Set up Security Group

We are going to set up the security groups that we are going to use in our virtual machines to define how much it will be accessible from outside. The security group is a kind of firewall for our resources.

## Go to AWS Management console

Console Home [Info](#)

Recently visited [Info](#)

- VPC
- EC2
- Control Tower
- AWS Organizations
- IAM

View all services

Welcome to AWS

Getting started with AWS [?](#)  
Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification [?](#)  
Learn from AWS experts and advance your skills and

AWS Health [Info](#)

Open issues 0 Past 7 days

Scheduled changes 0 Upcoming and past 7 days

Other notifications 0 Past 7 days

Cost and usage [Info](#)

No applications  
Get started by creating an application.  
[Create application](#)

Go to myApplications

CloudShell Feedback [Console Mobile App](#)

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Search for “EC2”

EC2 [Ask Amazon](#) [X](#)

Services

- EC2 Virtual Servers in the Cloud
- EC2 Image Builder A managed service to automate build, customize and deploy OS images
- Recycle Bin Protect resources from accidental deletion

Features

- Dashboard EC2 feature
- EC2 Instances CloudWatch feature
- AMIs EC2 feature

Resources in us-east-1 / for a focused search [Show more in Resource Explorer](#)

Were these results helpful?

Yes No

Looking for resources in other Regions?  
You can enable cross-Region search for resources across all Regions in your account by specifying an aggregator index.  
[Enable cross-Region search](#)

CloudShell Feedback [Console Mobile App](#)

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “EC2” under “Services”

The screenshot shows the AWS EC2 landing page. The left sidebar includes sections for Instances, Images, Elastic Block Store, and Network & Security (with 'Security Groups' highlighted). The main content area features a title 'Amazon Elastic Compute Cloud (EC2)' and a call-to-action button 'Launch a virtual server'. Below this are sections for 'Benefits and features' (highlighting scalability and control) and 'Use cases'. The bottom right contains 'Additional actions' and 'Pricing (US)' sections.

Click on “Security Groups”

The screenshot shows the 'Security Groups' page within the EC2 service. The left sidebar lists various EC2-related services. The main area displays a table of existing security groups with columns for Name, Security group ID, Security group name, VPC ID, and Description. At the top right of the table, there is a 'Create security group' button.

Name	Security group ID	Security group name	VPC ID	Description
-	sg-009ee77a4deac753e	launch-wizard-2	vpc-0d74d3736a240e572	launch-wizard-2 created 2025-12-22T
-	sg-04f95d06199a440ab	default	vpc-03495749bc14b79b8	default VPC security group
-	sg-087be0760a1dad7cd	launch-wizard-1	vpc-0d74d3736a240e572	launch-wizard-1 created 2025-12-22T
-	sg-0376da3c47f7ba87f	default	vpc-0d74d3736a240e572	default VPC security group

Click on “Create Security Group”

The screenshot shows the AWS EC2 'Create security group' page. In the 'Basic details' section, the security group name is set to 'MyWebServerGroup'. The description is 'Allows SSH access to developers'. The VPC is 'vpc-0d74d3736a240e572'. Under 'Inbound rules', there are no rules listed, with an 'Add rule' button. Under 'Outbound rules', there is a single rule for 'All traffic' to '0.0.0.0/0'. The 'Tags - optional' section is empty. The bottom navigation bar includes CloudShell, Feedback, Console Mobile App, and links for Account ID, United States (N. Virginia), Smith, Privacy, Terms, and Cookie preferences.

Let us give the Security Group a name. I will call it “Primary-SG”

The screenshot shows the AWS EC2 'Create security group' page with the security group name changed to 'Primary-SG'. The description remains 'Allows SSH access to developers'. The VPC is 'vpc-0d74d3736a240e572'. Under 'Inbound rules', there are no rules listed, with an 'Add rule' button. Under 'Outbound rules', there is a single rule for 'All traffic' to '0.0.0.0/0'. The 'Tags - optional' section is empty. The bottom navigation bar includes CloudShell, Feedback, Console Mobile App, and links for Account ID, United States (N. Virginia), Smith, Privacy, Terms, and Cookie preferences.

Then for the “Description”, I will enter “This is my primary Security Group”

The screenshot shows the AWS EC2 'Create security group' page. In the 'Basic details' section, 'Security group name' is set to 'Primary-SG'. The 'Description' field contains 'This is my primary Security Group'. Under 'VPC Info', the VPC is selected as 'vpc-0d74d3736a240e572'. The 'Inbound rules' section is empty. The 'Outbound rules' section shows a single rule for 'All traffic' with a destination of '0.0.0.0/0'. The 'Tags - optional' section is empty.

We are going to add the following inbound rules:

To do this, click on “Add Rule”

The screenshot shows the same EC2 'Create security group' page after adding an inbound rule. In the 'Inbound rules' section, a new rule has been added for 'Custom TCP' protocol on port '0' from 'Custom' source. The 'Outbound rules' section remains the same. The 'Tags - optional' section is still empty.

Here, enter the range “**30000-32767**” since we are going to use the Virtual machine as a Kubernetes cluster. We will be using this range for the deployment of our applications. Then, click on “Add Rule”

**Create security group** Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

### Basic details

Security group name Info  
Primary-SG  
Name cannot be edited after creation.

Description Info  
This is my primary Security Group

VPC Info  
vpc-0d74d3736a240e572

### Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
Custom TCP	TCP	30000 - 32767	Custom	
Custom TCP	TCP	0	Custom	

[Add rule](#) [Delete](#)

### Outbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Description - optional <small>Info</small>
All traffic	All	All	Custom	0.0.0.0/X

[Add rule](#) [Delete](#)

Now, we are going to add port 465, that is “**SMTPS**”. This is going to be used when we want to send mail notification from the Jenkins pipeline to our gmail email account.

To do this, click on the drop down on “**Type**”

**Create security group** Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Type Info: **SMTPS**

Protocol Info: TCP

Port range Info: 30000 - 32767

Source Info: Custom

Description - optional Info

[Add rule](#) [Delete](#)

### Outbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Description - optional <small>Info</small>
All traffic	All	All	Custom	0.0.0.0/X

[Add rule](#) [Delete](#)

Select “**SMTPS**”

A screenshot of the AWS EC2 'Create security group' interface. The 'Basic details' section shows a security group name 'Primary-SG' and a description 'This is my primary Security Group'. Under 'VPC Info', the VPC is set to 'vpc-0d74d3736a240e572'. The 'Inbound rules' section has two existing rules: one for Custom TCP port range 30000 - 32767 and another for SMTPS port 465. An orange arrow points to the 'Add rule' button. The 'Outbound rules' section shows a single rule for All traffic to 0.0.0.0/0.

Then, the next port we have to add is port 6443. This is required when you set up your Kubernetes cluster. So, click on “Add Rule”

A screenshot of the AWS EC2 'Create security group' interface. The 'Basic details' section shows a security group name 'Primary-SG' and a description 'This is my primary Security Group'. Under 'VPC Info', the VPC is set to 'vpc-0d74d3736a240e572'. The 'Inbound rules' section has three existing rules: one for Custom TCP port range 30000 - 32767, one for SMTPS port 465, and one for Custom TCP port 0. An orange arrow points to the 'Port range' field where '6443' is entered. The 'Outbound rules' section shows a single rule for All traffic to 0.0.0.0/0.

On “Port Range”, enter “6443”

The screenshot shows the 'Create security group' page in the AWS Management Console. In the 'Basic details' section, the security group name is 'Primary-SG' and the description is 'This is my primary Security Group'. The VPC is set to 'vpc-0d74d3736a240e572'. Under 'Inbound rules', there are three rules: a Custom TCP rule from port 30000 to 32767, an SMTPS rule from port 465, and a Custom TCP rule from port 6443. An 'Add rule' button is visible. Under 'Outbound rules', there is a single rule for 'All traffic' to '0.0.0.0/0'. The bottom of the screen shows standard AWS navigation links like CloudShell, Feedback, and Console Mobile App.

Now, let us add port 22, that is SSH. This will be used to access our virtual machine.

This screenshot is identical to the one above, showing the 'Create security group' page. However, an orange arrow points to the 'Type' dropdown menu in the 'Inbound rules' section, specifically highlighting the 'Custom TCP' option. This indicates where the user should click to add a new rule.

Click on the drop down on “Type”

The screenshot shows the 'Create security group' page in the AWS EC2 console. In the left sidebar, under 'Inbound rules', 'Custom TCP' is selected. The main area displays four inbound rules:

- Protocol: TCP, Port range: 30000 - 32767, Source: Custom, Description: optional
- Protocol: TCP, Port range: 465, Source: Custom, Description: optional
- Protocol: TCP, Port range: 6443, Source: Custom, Description: optional
- Protocol: TCP, Port range: 0, Source: Custom, Description: optional

At the bottom of the 'Inbound rules' section is a blue 'Add rule' button.

Below this is the 'Outbound rules' section, which is currently empty.

At the bottom of the page are standard AWS navigation links: CloudShell, Feedback, Console Mobile App, and links for Privacy, Terms, and Cookie preferences.

Select “SSH”

The screenshot shows the 'Create security group' page in the AWS EC2 console. Under 'Basic details', the 'Security group name' is set to 'Primary-SG'. The 'Description' field contains 'This is my primary Security Group'. The 'VPC Info' dropdown is set to 'vpc-0d74d3736a240e572'.

In the 'Inbound rules' section, 'SSH' is selected from the 'Protocol' dropdown. The resulting rule is:

- Protocol: TCP, Port range: 22, Source: Custom, Description: optional

At the bottom of the 'Inbound rules' section is a blue 'Add rule' button.

Below this is the 'Outbound rules' section, which is currently empty.

At the bottom of the page are standard AWS navigation links: CloudShell, Feedback, Console Mobile App, and links for Privacy, Terms, and Cookie preferences.

Then, we have to enable port 443 and 80 for HTTPS and HTTP respectively.

Screenshot of the AWS EC2 Security Groups 'Create security group' page.

**Basic details**

- Security group name**: Primary-SG
- Description**: This is my primary Security Group
- VPC Info**: vpc-0d74d3736a240e572

**Inbound rules**

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	30000 - 32767	Custom	
SMTPS	TCP	465	Custom	
Custom TCP	TCP	6443	Custom	
SSH	TCP	22	Custom	
HTTP	TCP	80	Custom	
HTTPS	TCP	443	Custom	

**Add rule** button highlighted with an orange arrow.

**Outbound rules**

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	

Page footer: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

We will now open the port range “**3000-10000**” because most of the applications can easily be deployed within this range. To do this, click on “**Add Rule**”

Screenshot of the AWS EC2 Security Groups 'Create security group' page.

**Inbound rules**

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	30000 - 32767	Custom	
SMTPS	TCP	465	Custom	
Custom TCP	TCP	6443	Custom	
SSH	TCP	22	Custom	
HTTP	TCP	80	Custom	
HTTPS	TCP	443	Custom	
Custom TCP	TCP	0	Custom	

**Add rule** button highlighted with an orange arrow.

**Outbound rules**

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

**Tags - optional**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Page footer: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Enter the range “**3000-10000**” on the “**Port Range**”

This is my primary Security Group

VPC Info  
vpc-0d74d3736a240e572

**Inbound rules**

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	30000 - 32767	Custom	
SMTPS	TCP	465	Custom	
Custom TCP	TCP	6443	Custom	
SSH	TCP	22	Custom	
HTTP	TCP	80	Custom	
HTTPS	TCP	443	Custom	
Custom TCP	TCP	3000 - 10000	Custom	

Add rule

**Outbound rules**

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

Add rule

**Tags - optional**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Then, we add the last rule which is port 25, that is “SMTP”. So, click on “Add Rule”

This is my primary Security Group

VPC Info  
vpc-0d74d3736a240e572

**Inbound rules**

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	30000 - 32767	Custom	
SMTPS	TCP	465	Custom	
Custom TCP	TCP	6443	Custom	
SSH	TCP	22	Custom	
HTTP	TCP	80	Custom	
HTTPS	TCP	443	Custom	
Custom TCP	TCP	3000 - 10000	Custom	
Custom TCP	TCP	0	Custom	

Add rule

**Outbound rules**

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

Add rule

**Tags - optional**

Click on the drop down on “Type”

This is my primary Security Group

**Custom TCP** ✓

Protocol	Port range	Info	Source	Info	Description - optional	Info
TCP	30000 - 32767		Custom	Q		Delete
TCP	465		Custom	Q		Delete
TCP	6443		Custom	Q		Delete
TCP	22		Custom	Q		Delete
TCP	80		Custom	Q		Delete
TCP	443		Custom	Q		Delete
TCP	3000 - 10000		Custom	Q		Delete
TCP	0		Custom	Q		Delete

Add rule

**Outbound rules** Info

Type	Info	Protocol	Info	Port range	Info	Destination	Info	Description - optional	Info
All traffic		All		All		Custom	Q	0.0.0.0/X	

Add rule

**Tags - optional**

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Select “**SMTP**”, this is another “**SMTP**” server commonly used by companies when we want to send email notifications over gmail. In all the rules the “**Source**” will be “**Anywhere**”.

HTTP

HTTPS

Custom TCP

SMTP

Add rule

**Outbound rules** Info

Type	Info	Protocol	Info	Port range	Info	Destination	Info	Description - optional	Info
All traffic		All		All		Custom	Q	0.0.0.0/X	

Add rule

**Tags - optional**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags

Create security group

Cancel Create security group

CloudShell Feedback Console Mobile App

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on “Create Security Group”

Screenshot of the AWS EC2 Security Groups page showing the creation of a new security group named "Primary-SG".

**Details:**

- Security group name: Primary-SG
- Owner: 642391958541
- Description: This is my primary Security Group
- VPC ID: vpc-0d74d3736a240e572
- Inbound rules count: 8 Permission entries
- Outbound rules count: 1 Permission entry

**Inbound rules (8):**

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0dcacfb5fe6e5261b	IPv4	SSH	TCP	22	0.0.0.0/0
-	sgr-0e169e89732002c60	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-0f5b7819ebf35eb8f	IPv4	SMTP	TCP	25	0.0.0.0/0
-	sgr-091e8aa0e685afce4	IPv4	SMTPS	TCP	465	0.0.0.0/0
-	sgr-0e95122aeef809de74	IPv4	Custom TCP	TCP	3000 - 10000	0.0.0.0/0
-	sgr-0e858983448a0c926	IPv4	HTTPS	TCP	443	0.0.0.0/0
-	sgr-08547104fed2f034f	IPv4	Custom TCP	TCP	30000 - 32767	0.0.0.0/0
-	sgr-0d12710872aba7787	IPv4	Custom TCP	TCP	6443	0.0.0.0/0

We are done with this part.

## 3.2 Create and Configure Virtual Machines Kubernetes

We are going to create multiple virtual machines in a secured network. We are going to create virtual machines for Kubernetes Master node and two slave nodes.

### 3.2.1 Create Virtual Machines for Kubernetes

We will start by creating three virtual machines for our Kubernetes cluster. Go to the EC2 dashboard on AWS Management Console.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a navigation sidebar with sections like Dashboard, Instances, Images, Elastic Block Store, Network & Security, and more. The main area is titled 'Resources' and shows various EC2 metrics: Instances (running) 0, Auto Scaling Groups 0, Capacity Reservations 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 1, Load balancers 0, Placement groups 0, Security groups 5, Snapshots 0, and Volumes 0. Below this, there are three main sections: 'Launch instance', 'Service health', and 'Zones'. The 'Launch instance' section has a prominent orange 'Launch instance' button. A red arrow points from the text 'Click on "Launch Instance"' to this button. The 'Service health' section shows the status as 'OK'. The 'Zones' section lists five availability zones: us-east-1a, us-east-1b, us-east-1c, us-east-1d, and us-east-1e, each associated with a Zone ID. To the right of the main content area, there are several cards: 'EC2 cost' (Date range: Past 6 months, Region: Global), 'Account attributes' (Default VPC vpc-0d74d3736a240e572), and 'Explore AWS' (Get Up to 40% Better Price Performance). The bottom of the page includes standard AWS footer links like CloudShell, Feedback, and Console Mobile App.

Click on “Launch Instance”

**Launch an instance** [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** [Info](#)

Name  
e.g. My Web Server  Add additional tags

**Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recent OS Images: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Amazon Linux 2023 kernel-6.1 AMI  
ami-068c0051b15cdb816 (64-bit (x86), uefi-preferred) / ami-0720c0a2e1e125edd (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**  
Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

CloudShell Feedback [Console Mobile App](#) © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Let us give the virtual machines the name “VM” for now, we will change the names after creating it.

**Launch an instance** [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** [Info](#)

Name  
VM  Add additional tags

**Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recent OS Images: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Amazon Linux 2023 kernel-6.1 AMI  
ami-068c0051b15cdb816 (64-bit (x86), uefi-preferred) / ami-0720c0a2e1e125edd (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**  
Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

CloudShell Feedback [Console Mobile App](#) © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Then on “Application and OS Images (Amazon Machine Image)”, we will choose “Ubuntu”

Then scroll down to “Instance Type” and select “t2.medium”

### ▼ Instance type [Info](#) | [Get advice](#)

#### Instance type

t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true  
On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour On-Demand Linux base pricing: 0.0464 USD per Hour  
On-Demand RHEL base pricing: 0.0752 USD per Hour On-Demand Windows base pricing: 0.0644 USD per Hour  
On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

Scroll down to “Key Pair”

### ▼ Key pair (login) [Info](#)

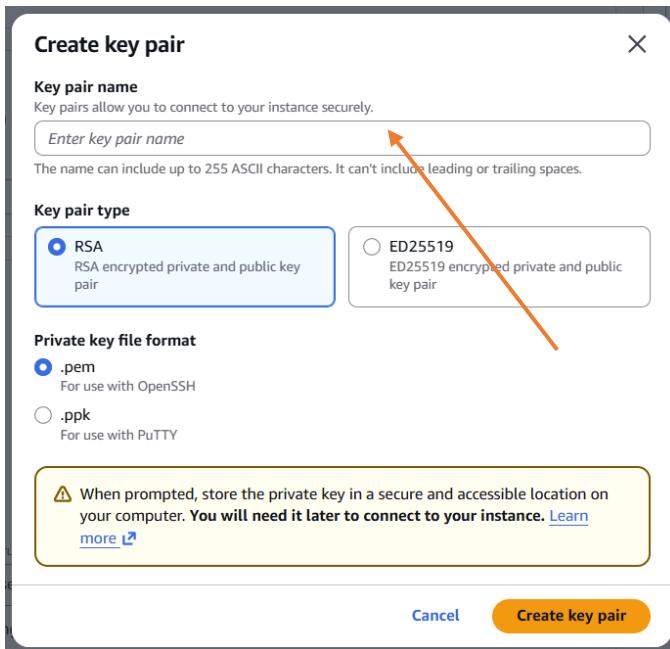
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name - required**

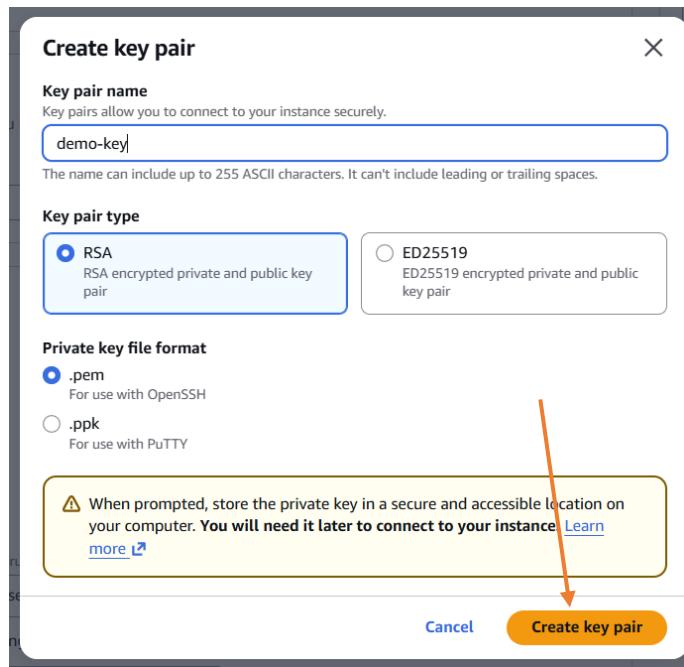
Select

Create new key pair

Click on “Create new key pair”



Let us give the key a name, we will call it “**demo-key**”



Then, click on “**Create key pair**”

**▼ Key pair (login) Info**

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name - required**

demo-key

▼ **Create new key pair**

The key pair has been created and downloaded to the “**Downloads**” folder. Then scroll down to “**Network Settings**”.

▼ Network settings [Info](#)

[Edit](#)

**Network** | [Info](#)  
vpc-0d74d3736a240e572

**Subnet** | [Info](#)  
No preference (Default subnet in any availability zone)

**Auto-assign public IP** | [Info](#)  
Enable

**Firewall (security groups)** | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group       Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

Allow SSH traffic from Anywhere  
Helps you connect to your instance

Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

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

Click on “Select Existing Security Group”

▼ Network settings [Info](#)

[Edit](#)

**Network** | [Info](#)  
vpc-0d74d3736a240e572

**Subnet** | [Info](#)  
No preference (Default subnet in any availability zone)

**Auto-assign public IP** | [Info](#)  
Enable

**Firewall (security groups)** | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group       Select existing security group

**Common security groups** | [Info](#)

Select security groups ▼  Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Click on the drop down on “Common Security Groups”

**▼ Network settings** [Info](#)

**Network** | [Info](#)  
vpc-0d74d3736a240e572

**Subnet** | [Info](#)  
No preference (Default subnet in any availability zone)

**Auto-assign public IP** | [Info](#)  
Enable

**Firewall (security groups)** | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group       Select existing security group

**Common security groups** | [Info](#)

Select security groups ▾

Compare security group rules

Advanced

Security Group	Description
<input type="checkbox"/> Primary-SG VPC: vpc-0d74d3736a240e572	sg-002d4edfb66259799
<input type="checkbox"/> launch-wizard-2 VPC: vpc-0d74d3736a240e572	sg-009ee77a4deac753e
<input type="checkbox"/> default VPC: vpc-0d74d3736a240e572	sg-0376da3c47f7ba87f
<input type="checkbox"/> launch-wizard-1 VPC: vpc-0d74d3736a240e572	sg-087be0760a1dad7cd

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

[Edit](#)

Then, select “**Primary-SG**”

**▼ Network settings** [Info](#)

**Network** | [Info](#)  
vpc-0d74d3736a240e572

**Subnet** | [Info](#)  
No preference (Default subnet in any availability zone)

**Auto-assign public IP** | [Info](#)  
Enable

**Firewall (security groups)** | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group       Select existing security group

**Common security groups** | [Info](#)

Select security groups ▾

Compare security group rules

Primary-SG sg-002d4edfb66259799 [X](#)  
VPC: vpc-0d74d3736a240e572

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Then, scroll down to “**Configure Storage**” and make the value “**20GiB**”

**▼ Configure storage** [Info](#)

Advanced

1x  GiB  [▼](#) Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

---

(?) Click refresh to view backup information [↻](#)  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

---

0 x File systems [Edit](#)

**► Advanced details** [Info](#)

Then, go to the summary and make the “Number of Instances” as “3”

**▼ Summary**

Number of instances | [Info](#)

3

When launching more than 1 instance, [consider EC2 Auto Scaling](#)

---

**Software Image (AMI)**  
Canonical, Ubuntu, 24.04, amd6... [read more](#)  
ami-0ecb62995f68bb549

---

**Virtual server type (instance type)**  
t2.medium

---

**Firewall (security group)**  
Primary-SG

---

**Storage (volumes)**  
1 volume(s) - 20 GiB

---

[Cancel](#) [Launch instance](#)  [Preview code](#)

Then click on “Launch Instance”

Success  
Successfully initiated launch of instances (i-0618808c6e1829c35, i-0b47272282d6961f5, i-0d0eabfc020a5de7a)

[Launch log](#)

**Next Steps**

Q. What would you like to do next with these instances, for example "create alarm" or "create backup"

<b>Create billing usage alerts</b> To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds. <a href="#">Create billing alerts</a>	<b>Connect to your instance</b> Once your instance is running, log into it from your local computer. <a href="#">Learn more</a>	<b>Connect an RDS database</b> Configure the connection between an EC2 instance and a database to allow traffic flow between them. <a href="#">Connect an RDS database</a> <a href="#">Create a new RDS database</a> <a href="#">Learn more</a>	<b>Create EBS snapshot policy</b> Create a policy that automates the creation, retention, and deletion of EBS snapshots. <a href="#">Create EBS snapshot policy</a>
<b>Manage detailed monitoring</b> Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period. <a href="#">Manage detailed monitoring</a>	<b>Create Load Balancer</b> Create an application, network gateway or classic Elastic Load Balancer. <a href="#">Create Load Balancer</a>	<b>Create AWS budget</b> AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. <a href="#">Create AWS budget</a>	<b>Manage CloudWatch alarms</b> Create or update Amazon CloudWatch alarms for the instance. <a href="#">Manage CloudWatch alarms</a>

CloudShell Feedback [Console Mobile App](#) © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The instance is being created. Click on “Instances”

EC2 > Instances

**Instances (3) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
VM	i-0b47272282d6961f5	Running	t2.medium	Initializing	<a href="#">View alarms</a>	us-east-1c
VM	i-0618808c6e1829c35	Running	t2.medium	Initializing	<a href="#">View alarms</a>	us-east-1c
VM	i-0d0eabfc020a5de7a	Running	t2.medium	Initializing	<a href="#">View alarms</a>	us-east-1c

Select an instance

CloudShell Feedback [Console Mobile App](#) © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

You can see that three instances have been created and they are initializing. Wait for them to pass the **“2/2 check”**.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table titled "Instances (3) Info" with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The table lists three VM instances, all of which are running and have passed 2/2 checks. The "Actions" and "Launch instances" buttons are visible at the top right of the table.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
VM	i-0b47272282d6961f5	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
VM	i-0618808c6e1829c35	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
VM	i-0d0eabfc020a5de7a	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c

The instances have passed the “**2/2 Check**”. Now, let us rename the instances. We will name them as “**Master**, **Slave-1** and **Slave-2**”.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table titled "Instances (1/3) Info" with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The table lists three VM instances, all of which are running and have passed 2/2 checks. The instance "Slave-2" is selected, indicated by a blue border around its row. The "Actions" and "Launch instances" buttons are visible at the top right of the table.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0b47272282d6961f5	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Slave-1	i-0618808c6e1829c35	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
<b>Slave-2</b>	i-0d0eabfc020a5de7a	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c

Below the table, a detailed view for the selected instance "i-0d0eabfc020a5de7a (VM)" is shown. The "Details" tab is selected. The instance summary section shows the instance ID (i-0d0eabfc020a5de7a), Public IPv4 address (100.31.153.217), Instance state (Running), and Private IP DNS name (IPv4 only) (ip-172-31-16-191.ec2.internal). The "Private IPv4 addresses" section shows the private IP address (172.31.16.191) and Public DNS (ec2-100-31-153-217.compute-1.amazonaws.com).

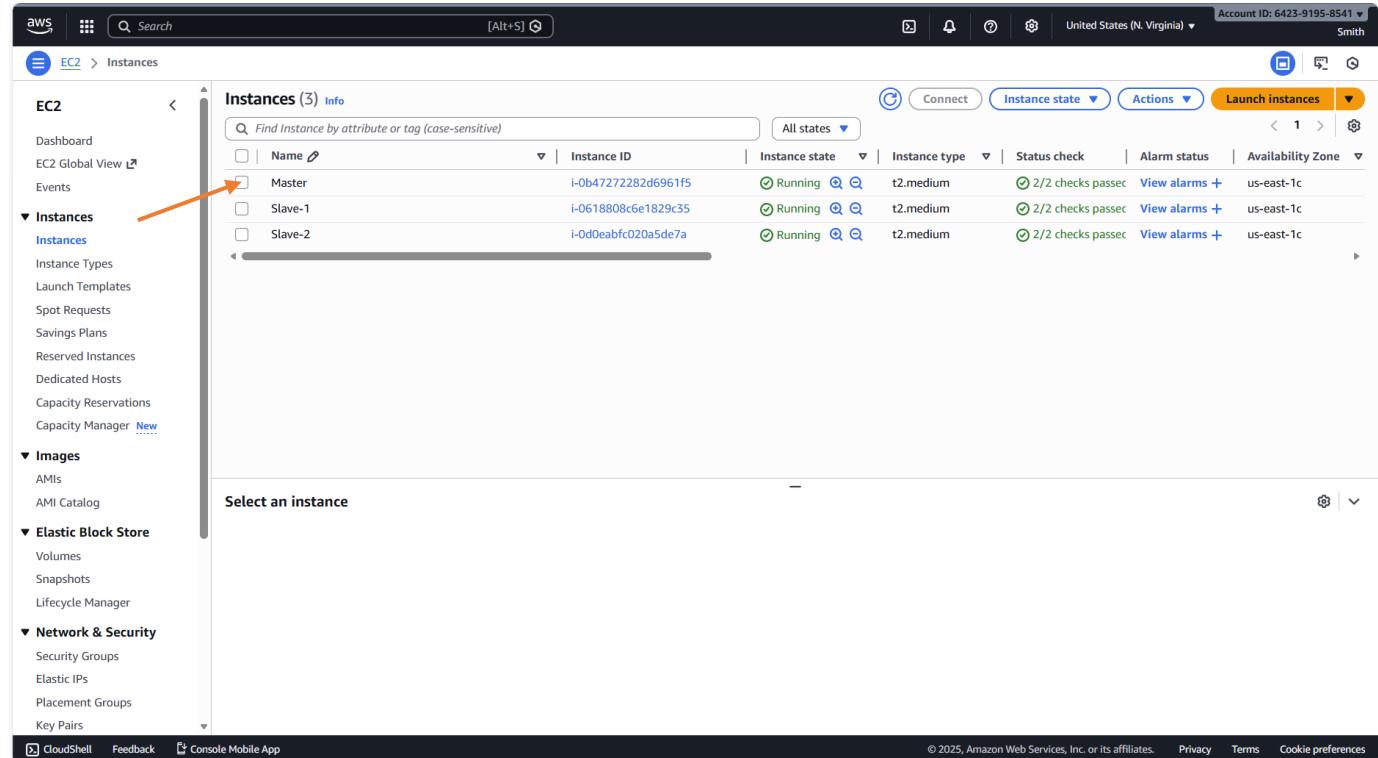
We have created the virtual machines, the next thing we have to do is to configure them.

### 3.2.2 SSH Connect to the Virtual Machines

We have to SSH connect to the virtual machine before we can configure the.

#### 3.2.2.1 SSH Connect to Virtual Machine “Master”

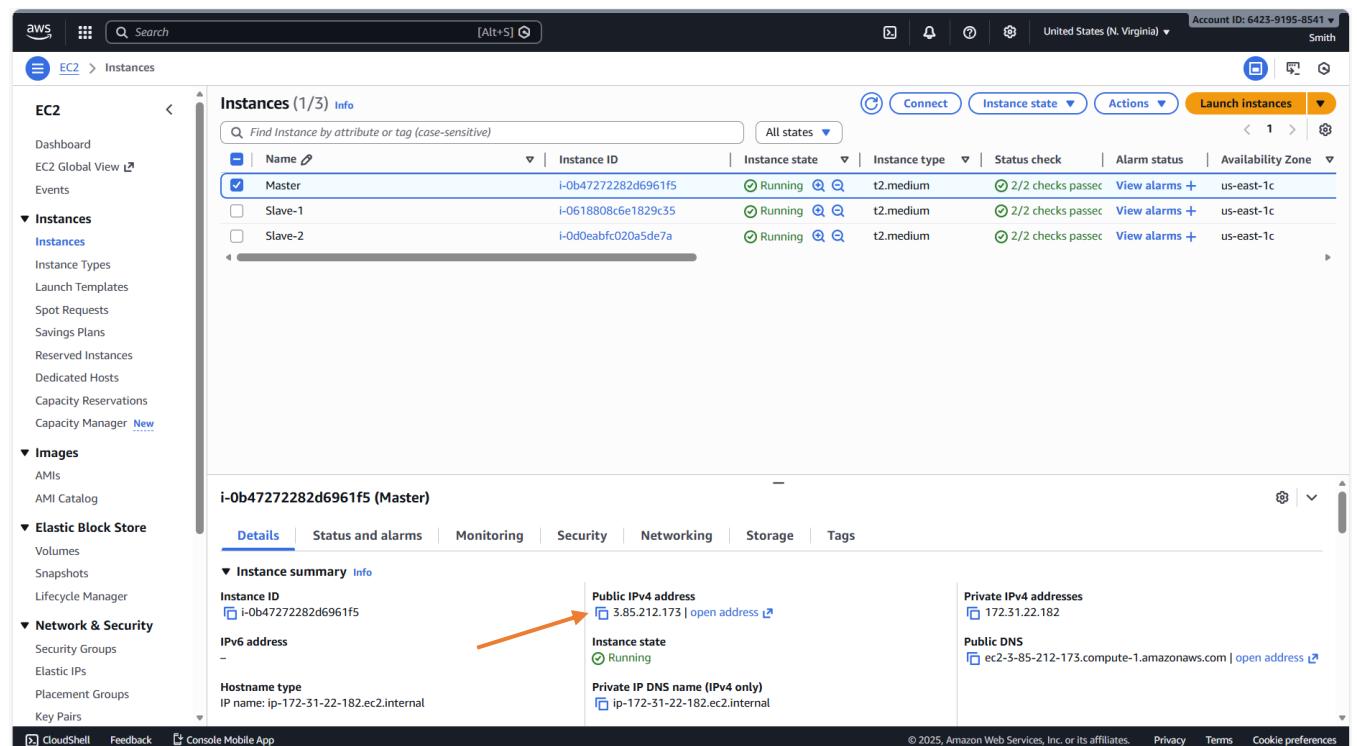
We will start by SSH connecting to the first virtual machine called “Master”.



The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like EC2, Dashboard, EC2 Global View, Events, Instances (selected), Images, Elastic Block Store, Network & Security, and CloudShell. The main area displays a table titled "Instances (3) Info" with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The "Master" instance is selected, indicated by a checked checkbox in the first column. The table shows the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0b47272282d6961f5	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Slave-1	i-0618808c6e1829c35	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Slave-2	i-0d0eabfc020a5de7a	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c

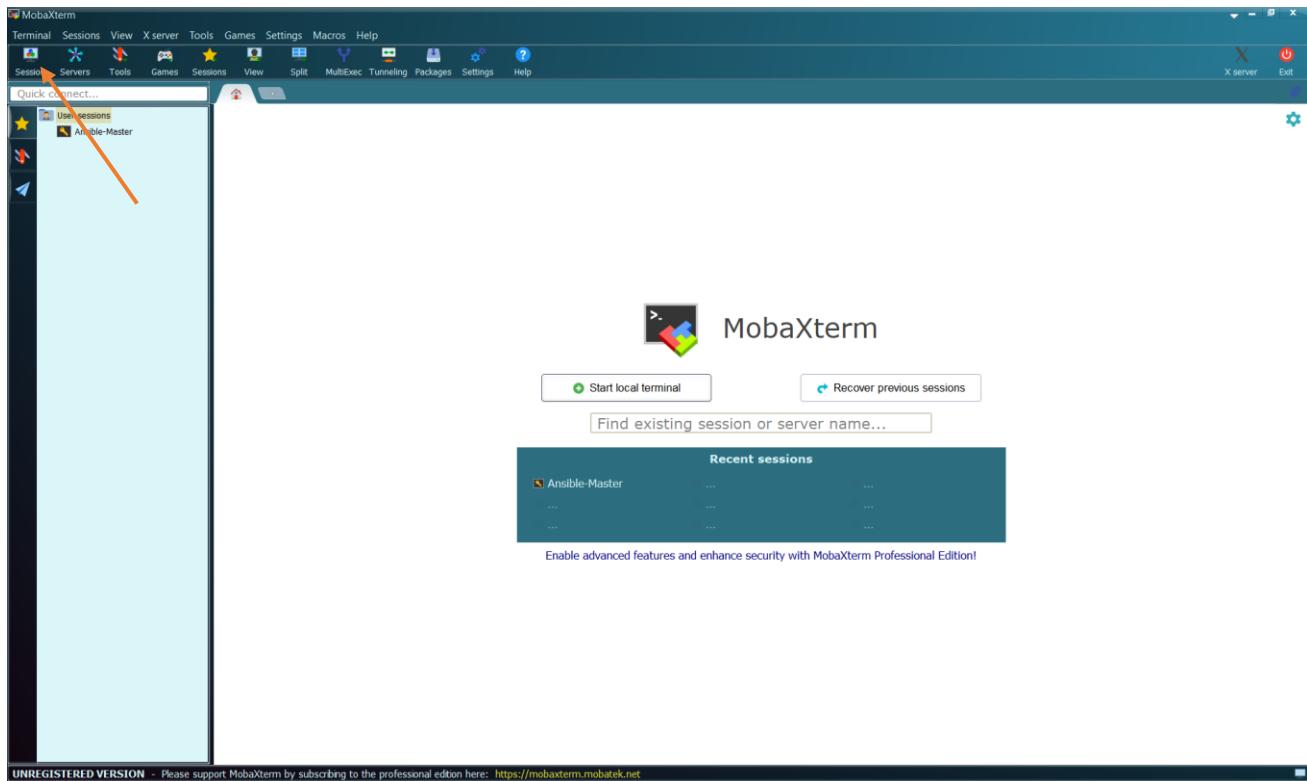
Select the virtual machine “Master”



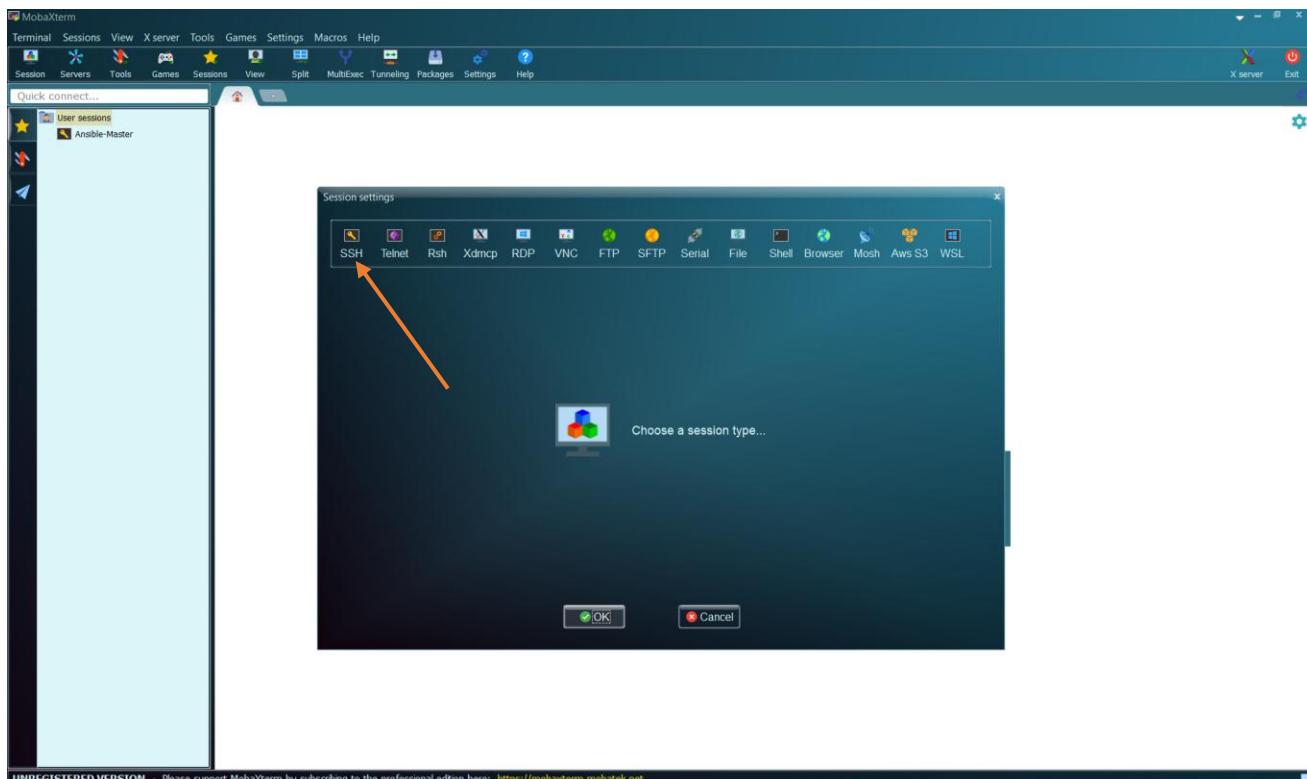
The screenshot shows the AWS EC2 Instances page with the "Master" instance selected. The navigation sidebar and table structure are identical to the previous screenshot. Below the table, a detailed view for the "Master" instance is shown under the heading "i-0b47272282d6961f5 (Master)". The "Details" tab is selected, displaying various instance details. An orange arrow points to the "Public IPv4 address" field, which contains the value "3.85.212.173". Other visible fields include "Instance ID" (i-0b47272282d6961f5), "IPv6 address" (empty), "Hostname type" (IP name: ip-172-31-22-182.ec2.internal), "Instance state" (Running), "Public IP DNS name (IPv4 only)" (ip-172-31-22-182.ec2.internal), and "Private IP addresses" (172.31.22.182). The "Status and alarms", "Monitoring", "Security", "Networking", "Storage", and "Tags" tabs are also present in the navigation bar below the details.

Copy the Public IP address: 3.85.212.173

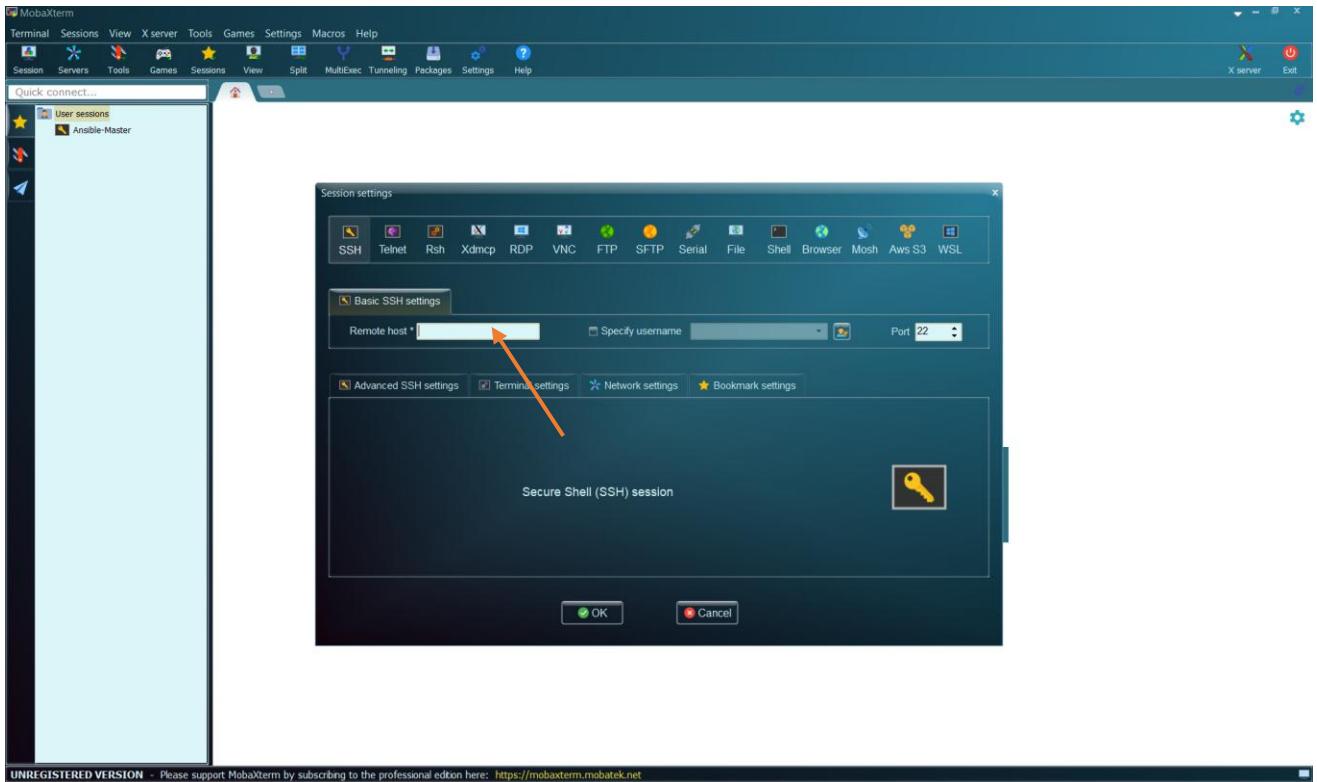
Open MobaXterm



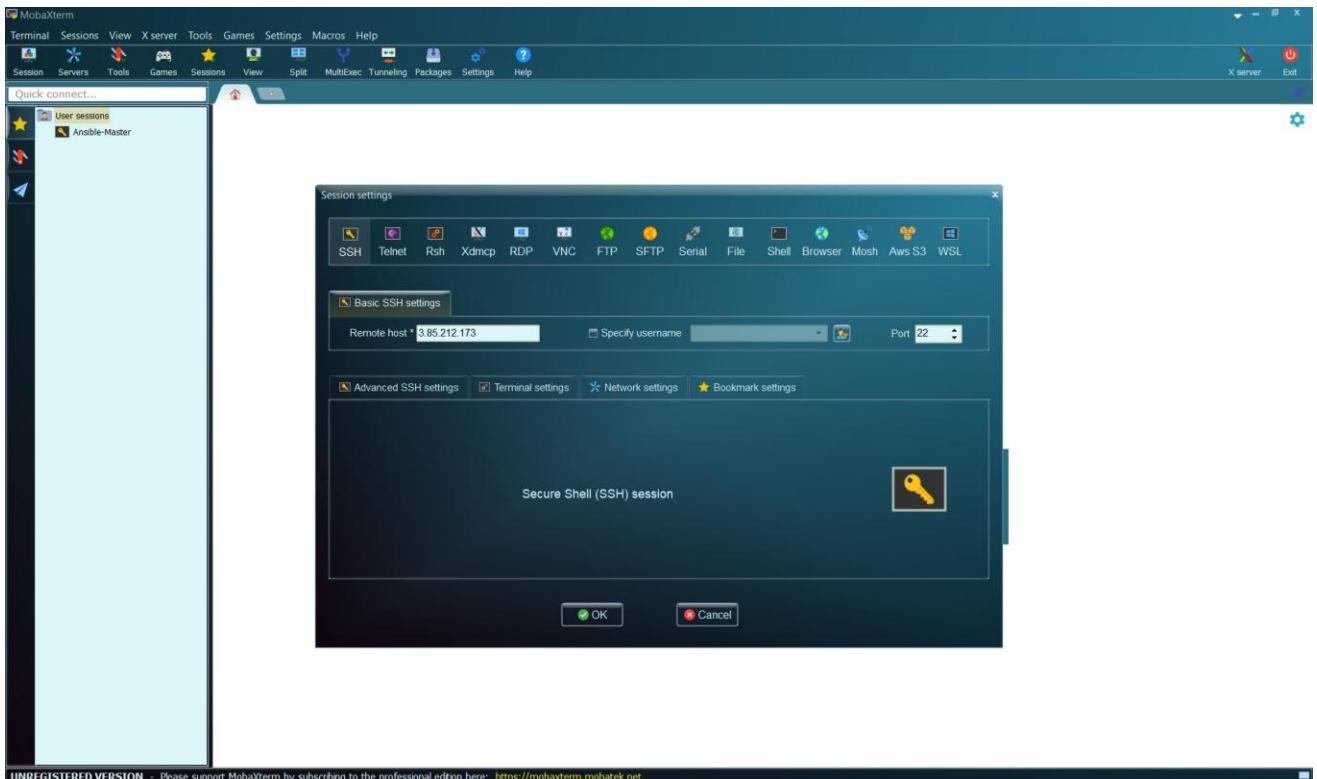
Click on “Session”



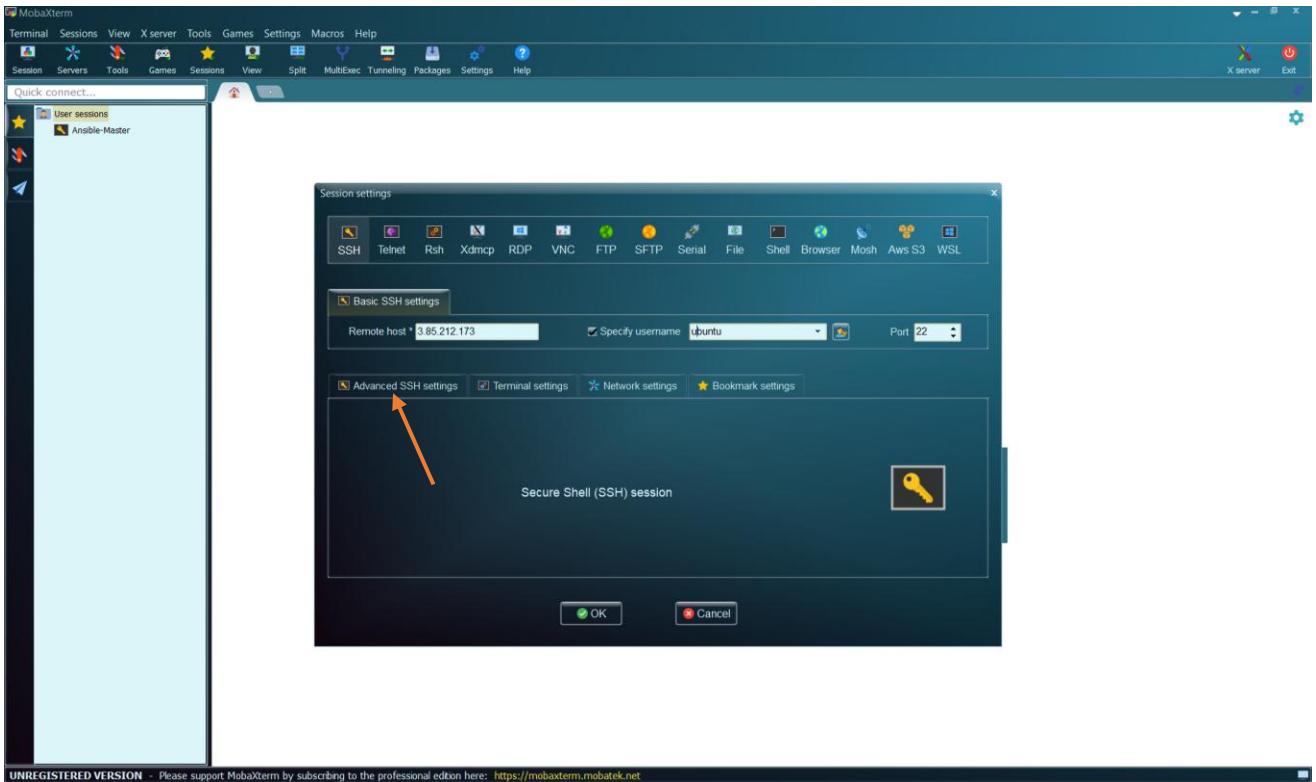
Click on “SSH”



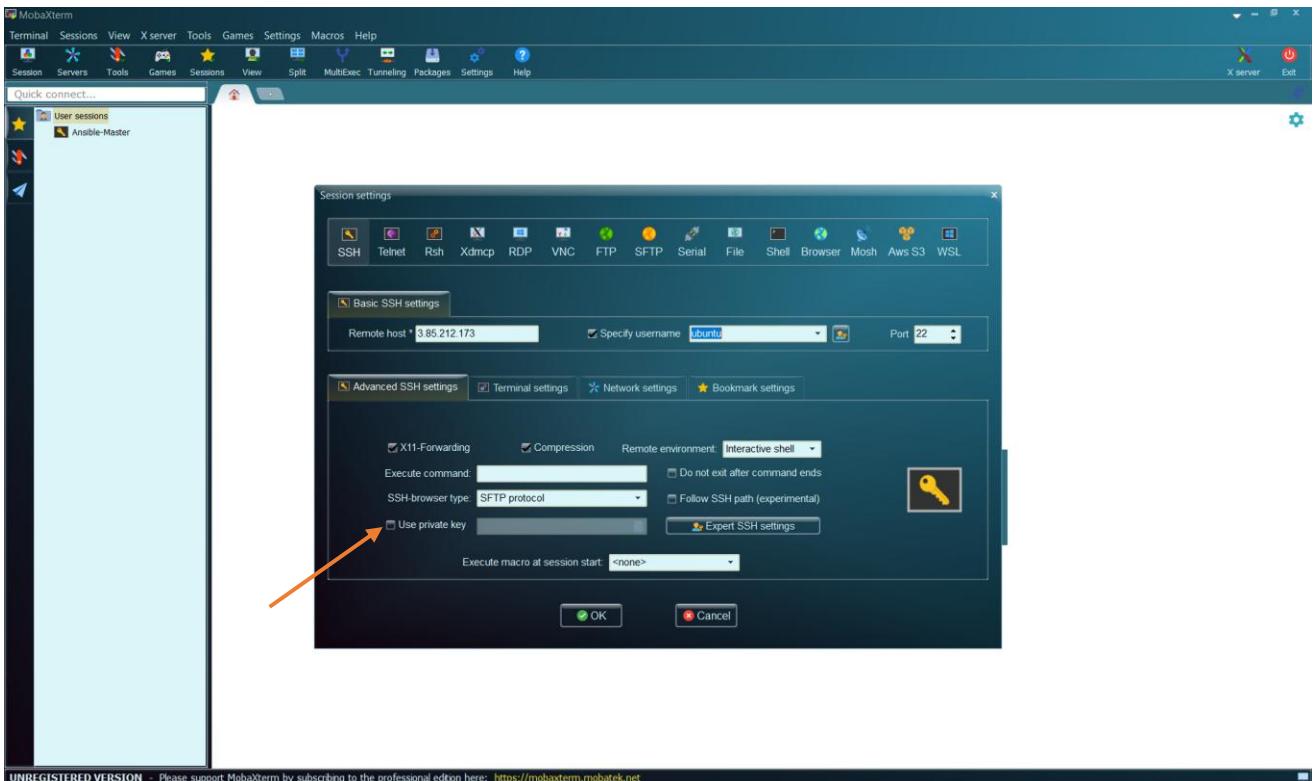
Then, enter the copied Public IP address here.



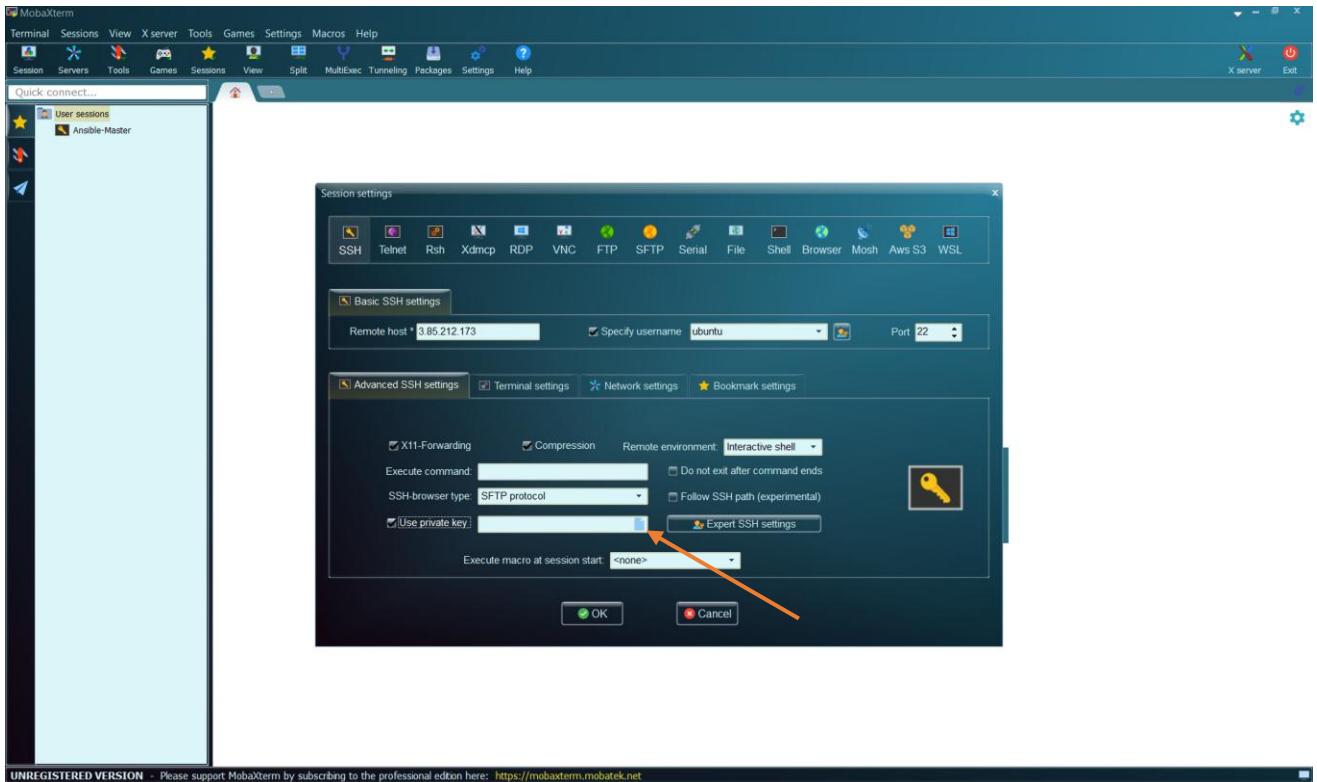
Then. Check the box “Specify username” and enter the name “ubuntu” since we are using ubuntu virtual machine.



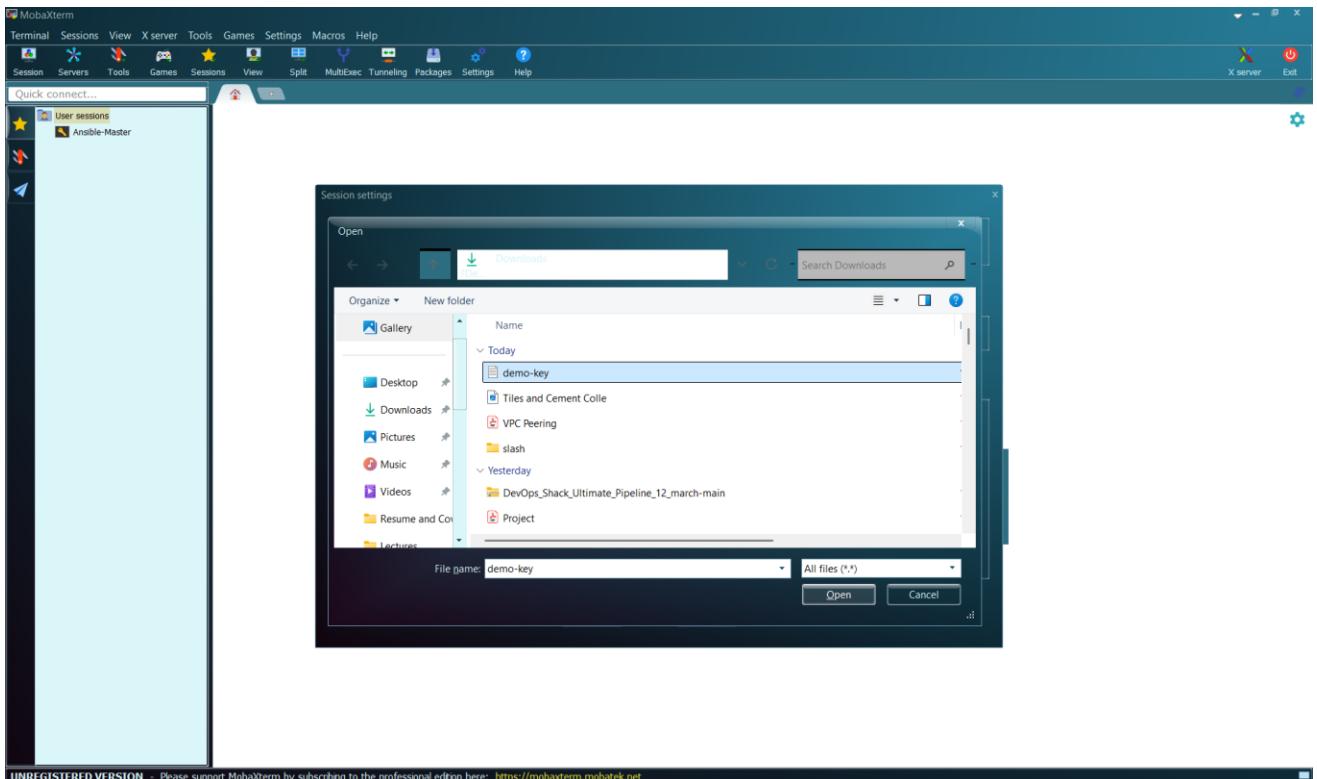
Then, click on “Advanced Settings”



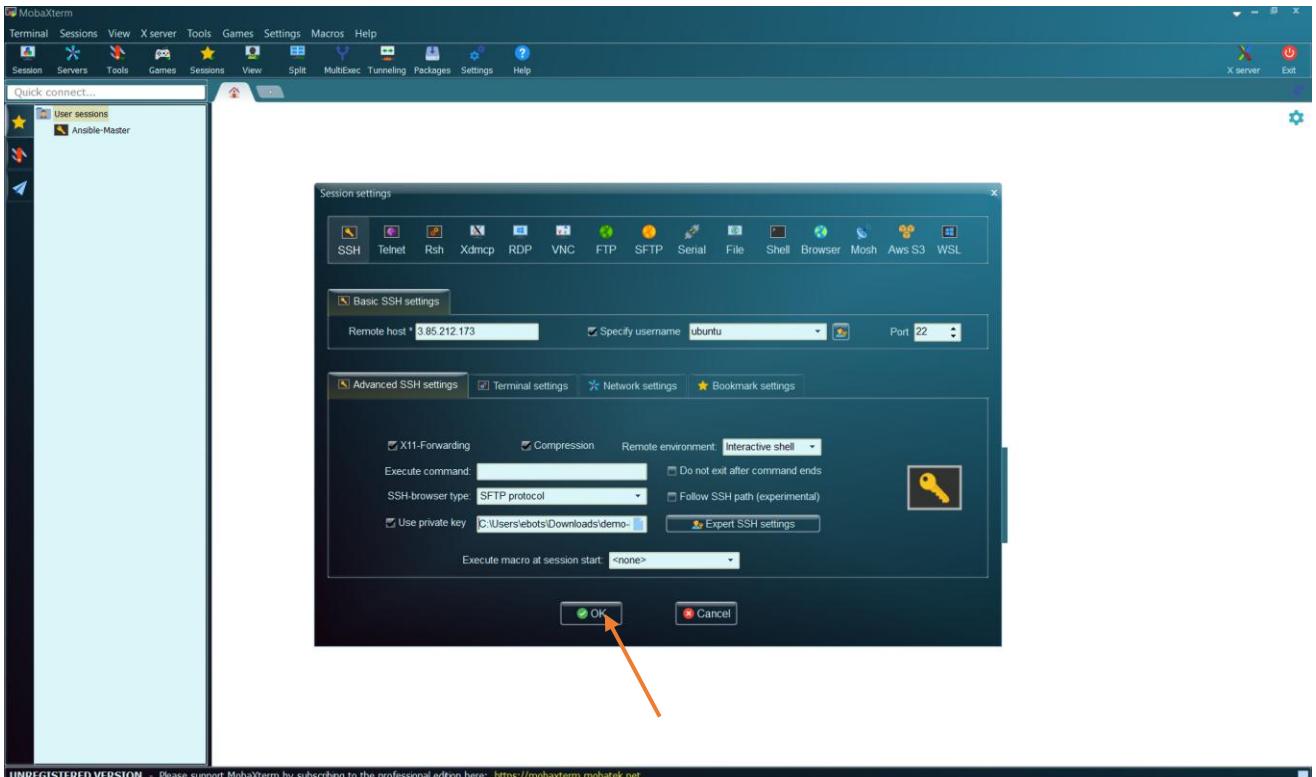
Since we are going to connect using the private key, check the box “Use Private Key”



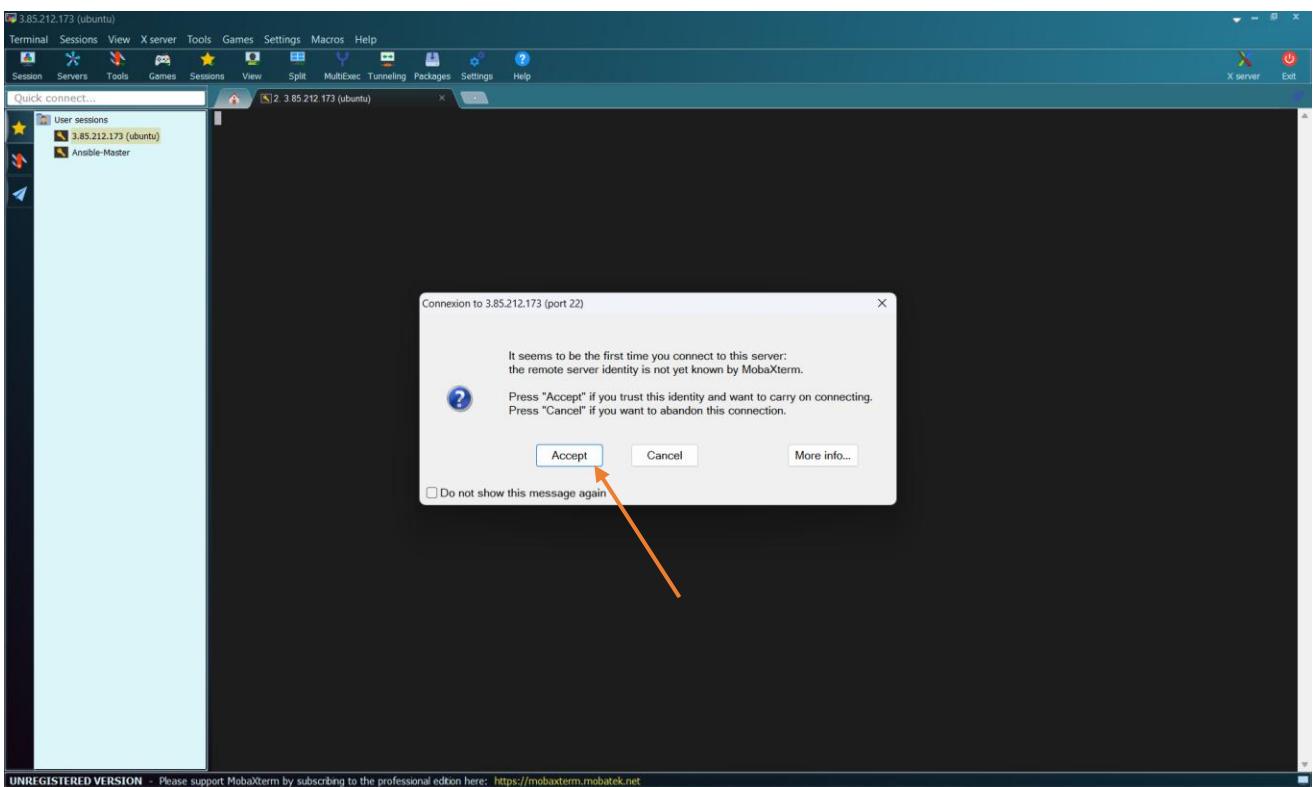
Click there to browse to where the private key is saved.



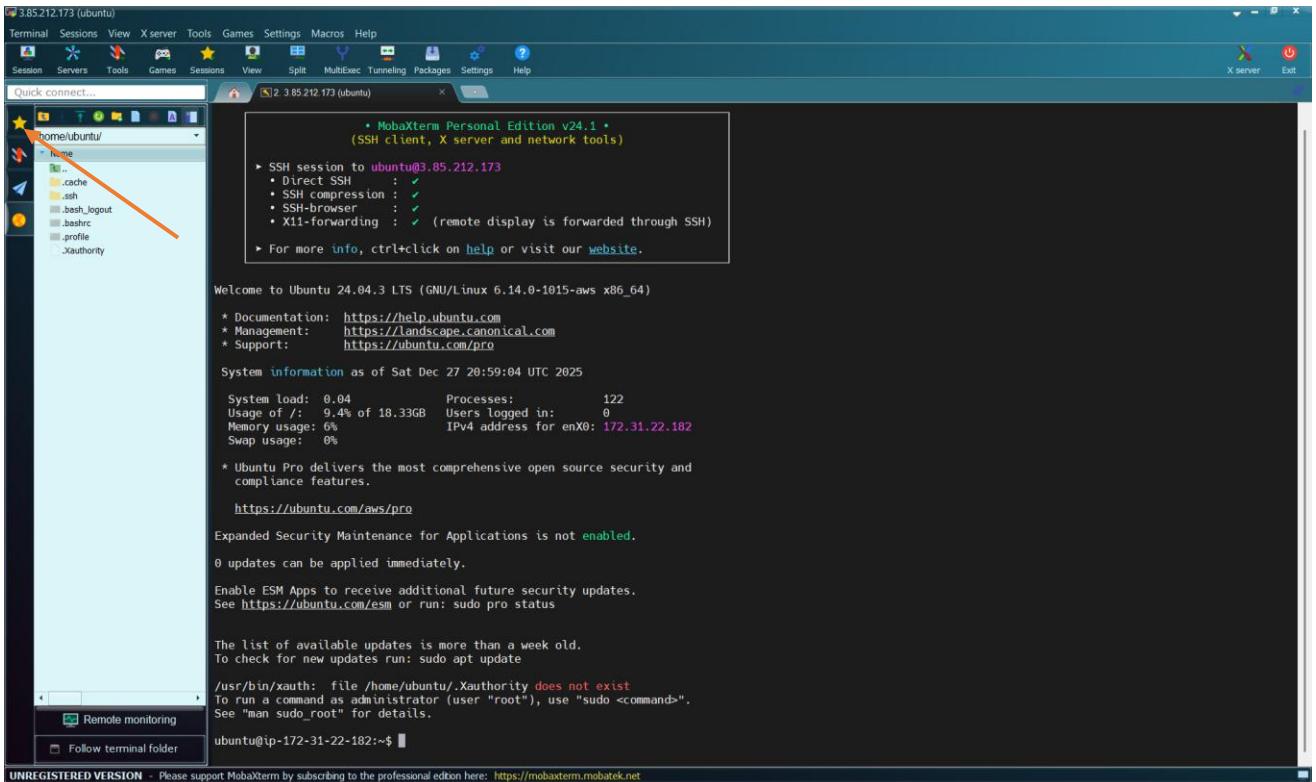
Select the private key and click on “Open”



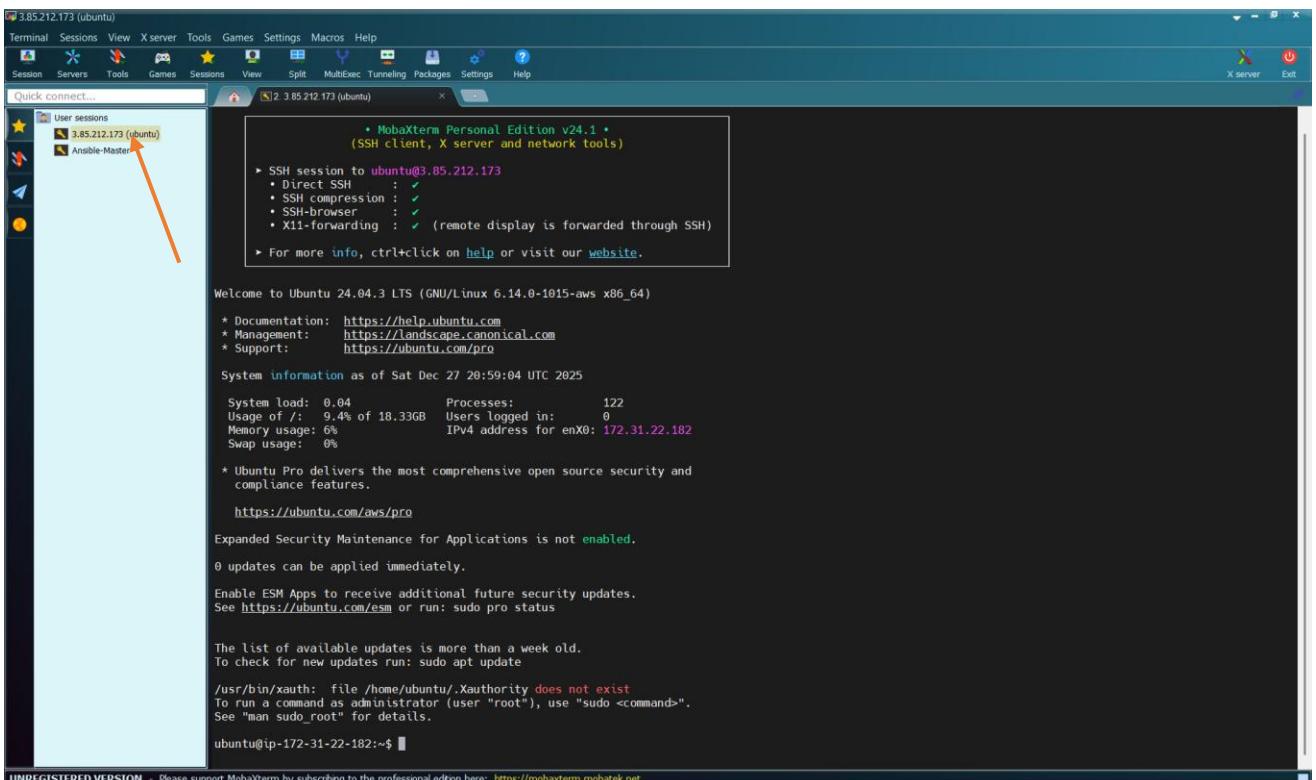
Then, click on “OK”



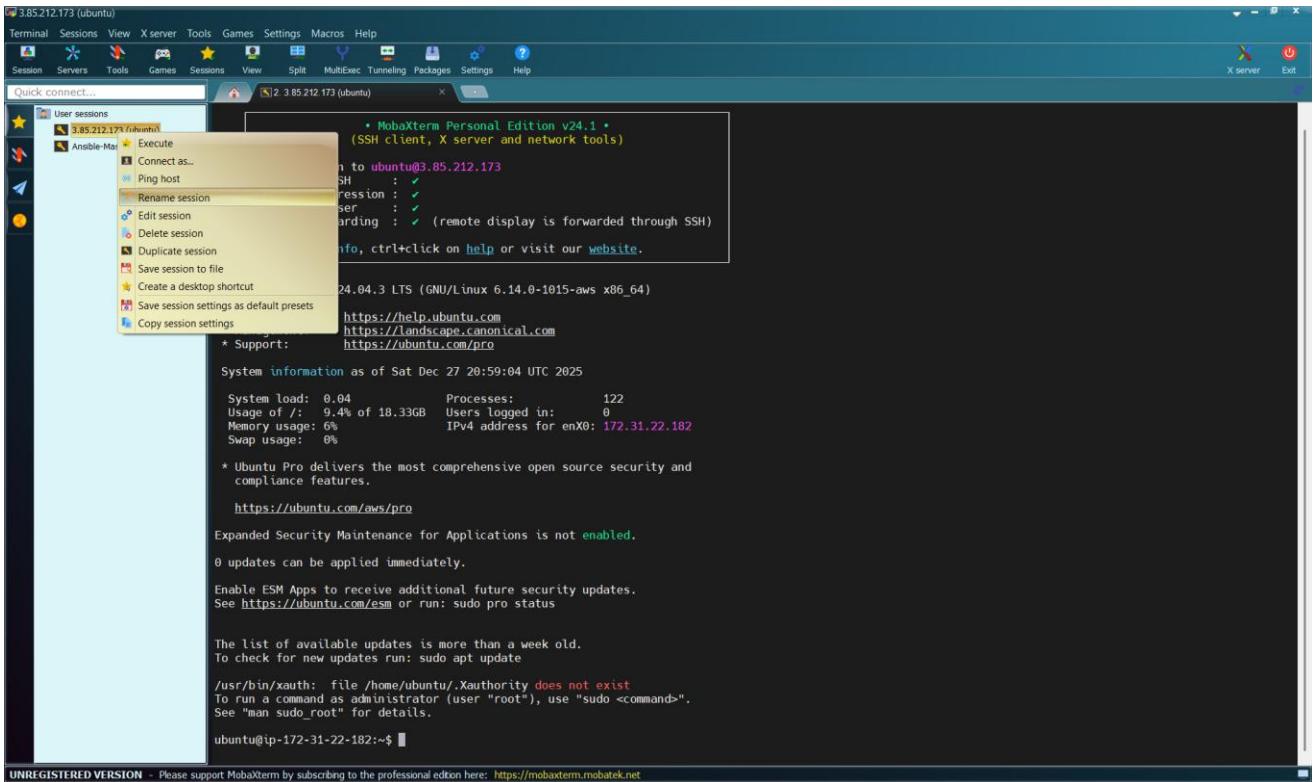
And click on “Accept”



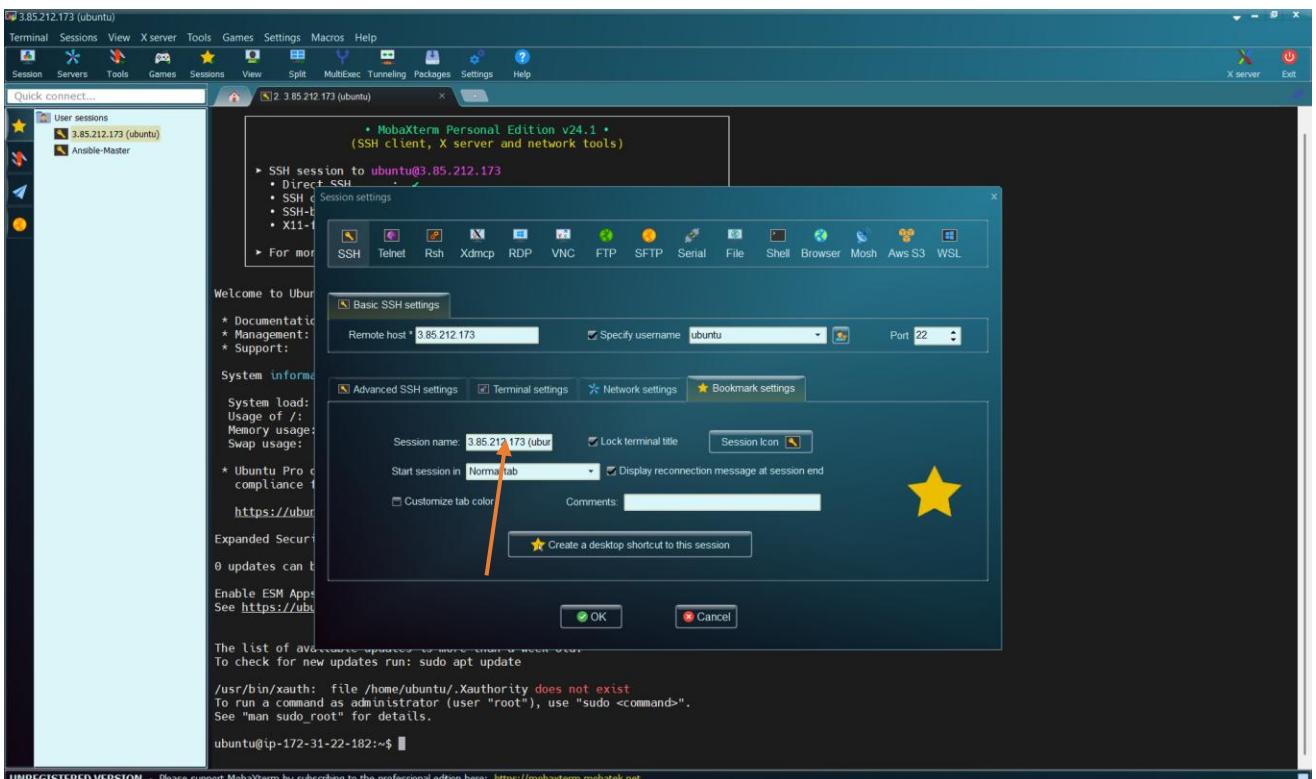
We are now connected to the virtual machine. Then, let us rename it as “Master”. To do this, click on the star.



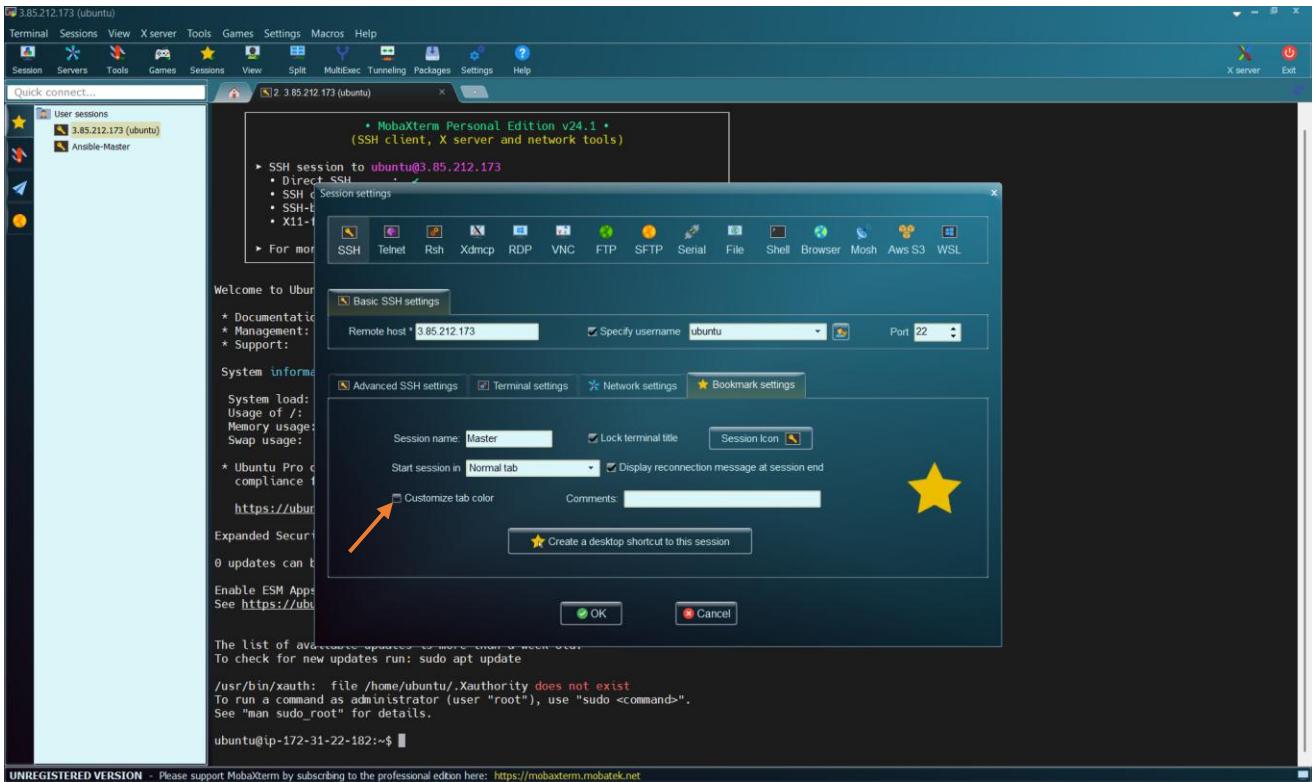
Then, right-click on the name of the session



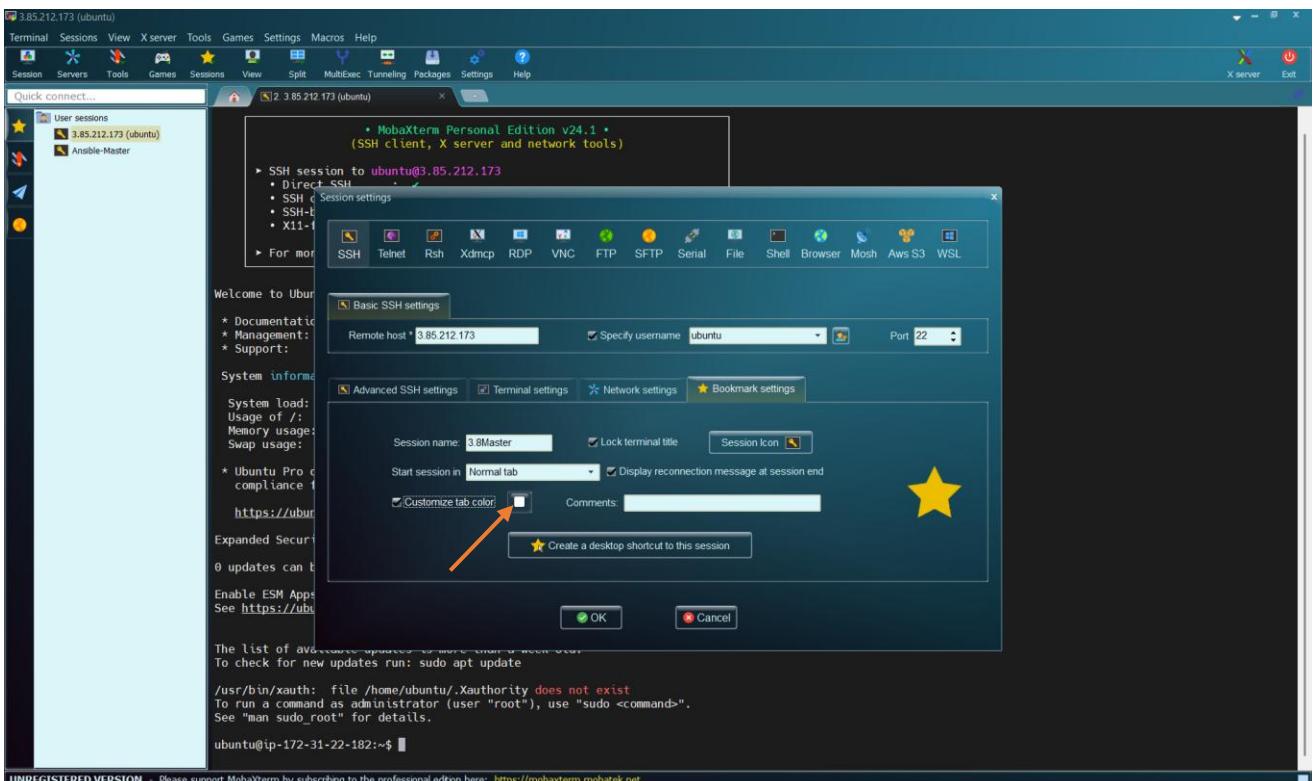
## Select “Rename Session”



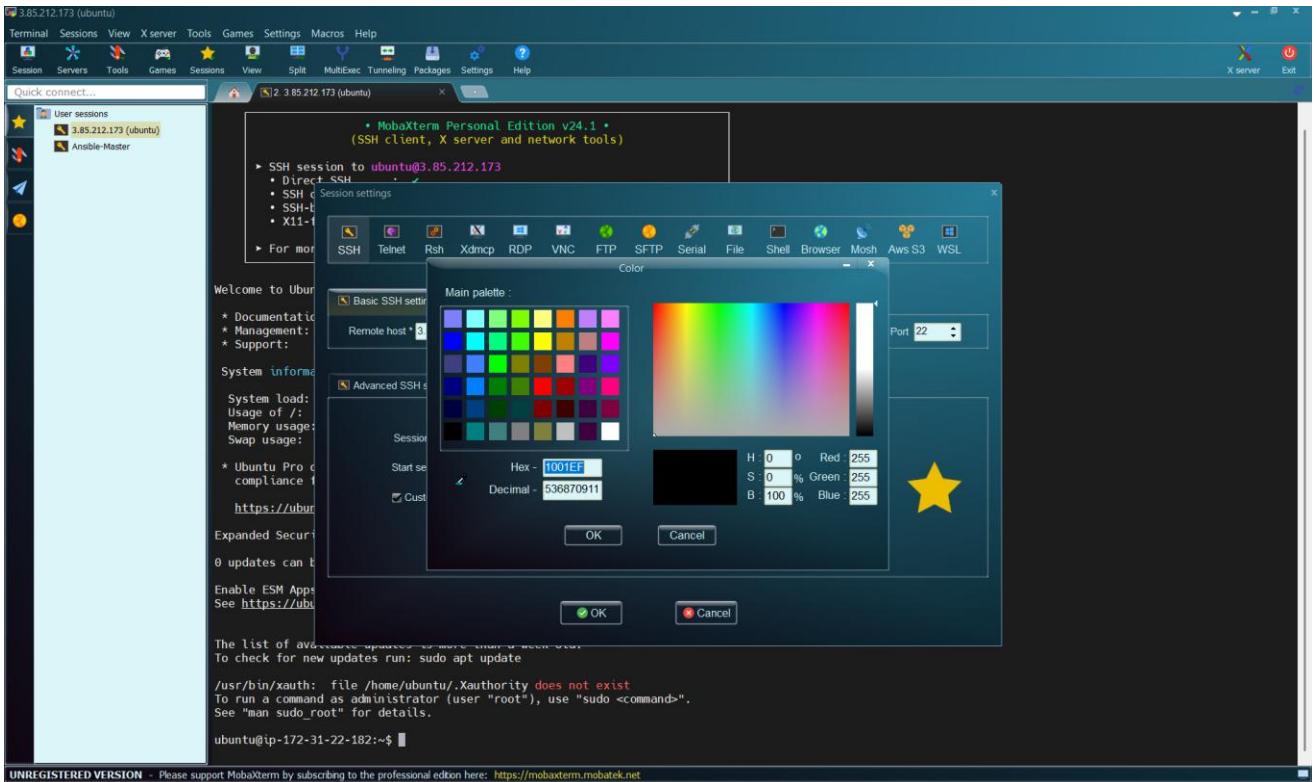
Then enter the name “Master”



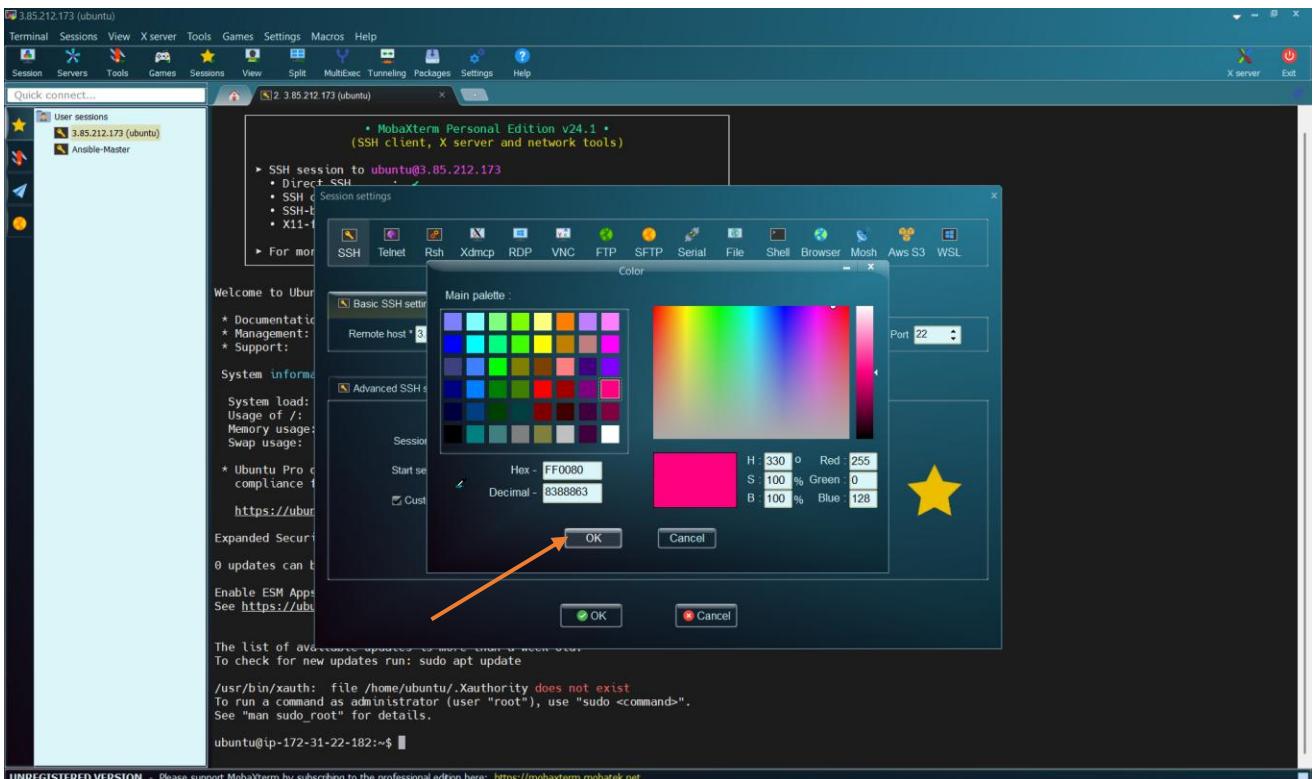
And we can change the colour by checking the box “Customize tab color”



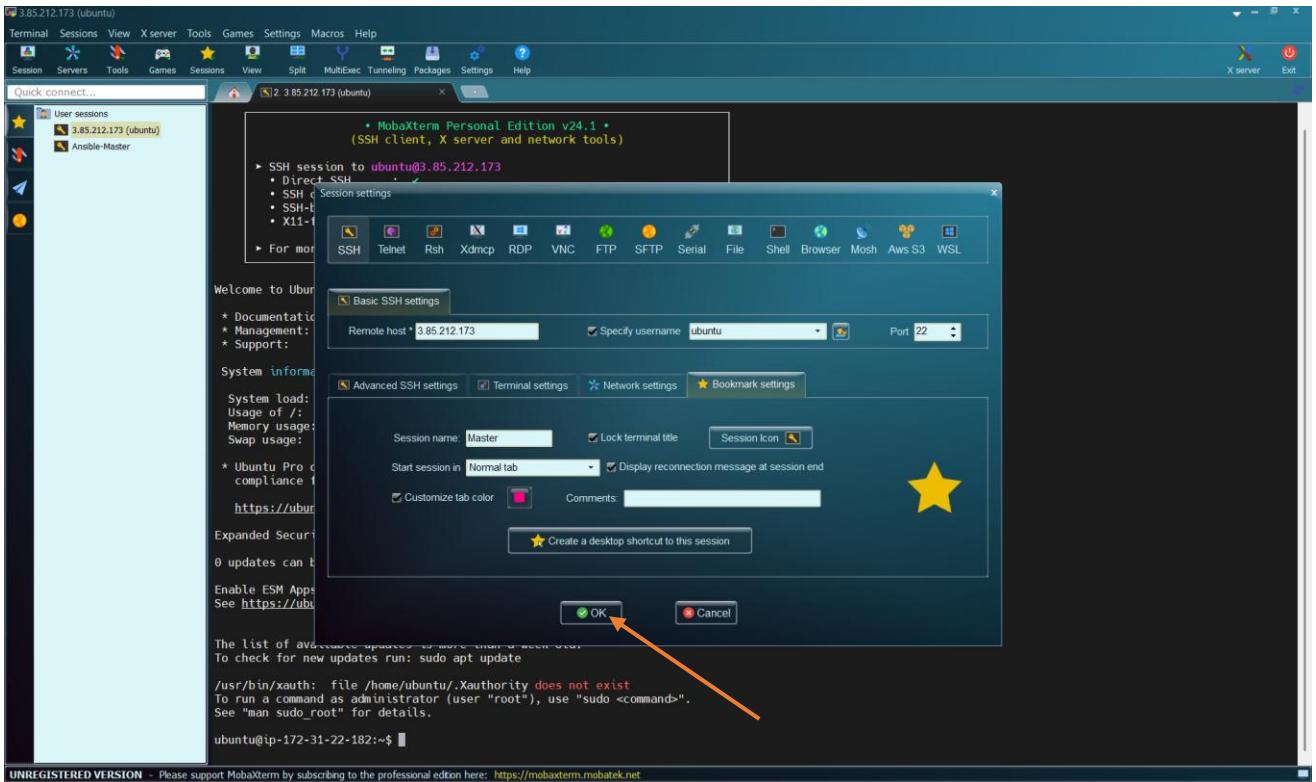
Then, click there



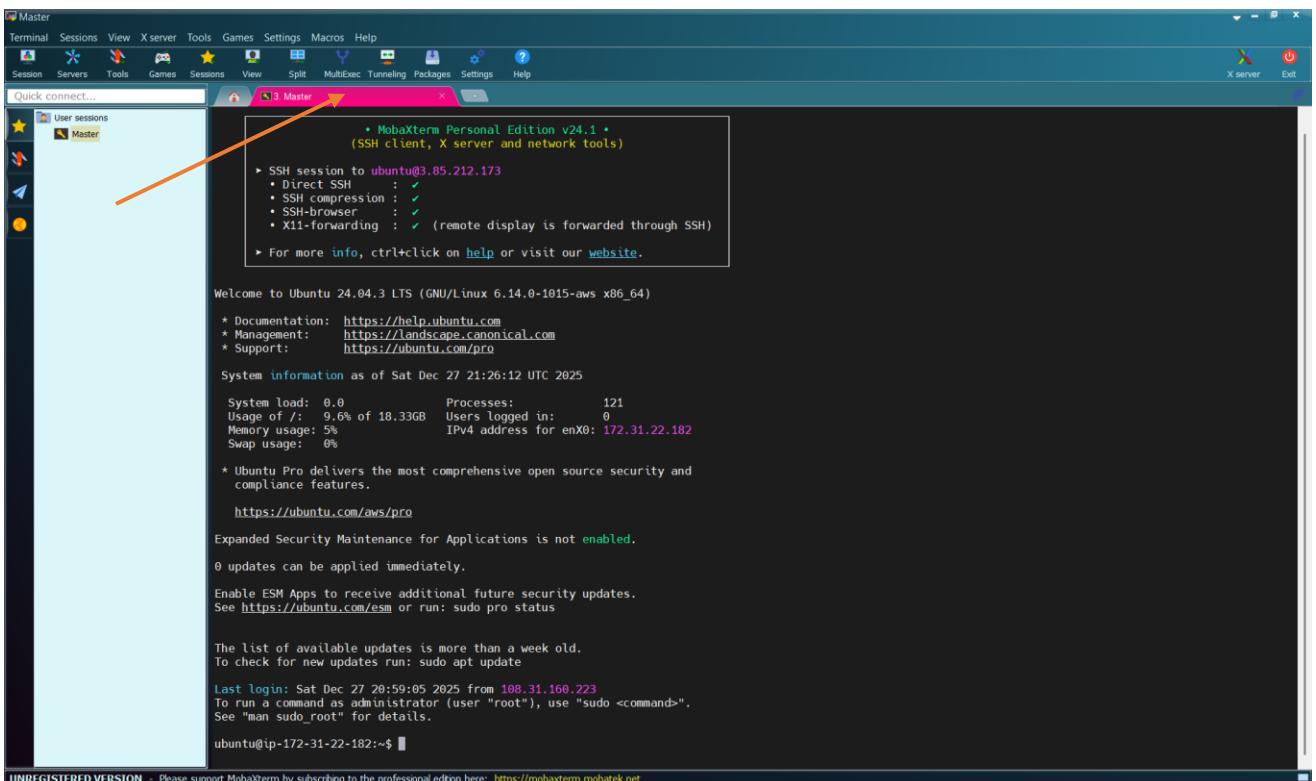
Select “Red”



And click on “OK”



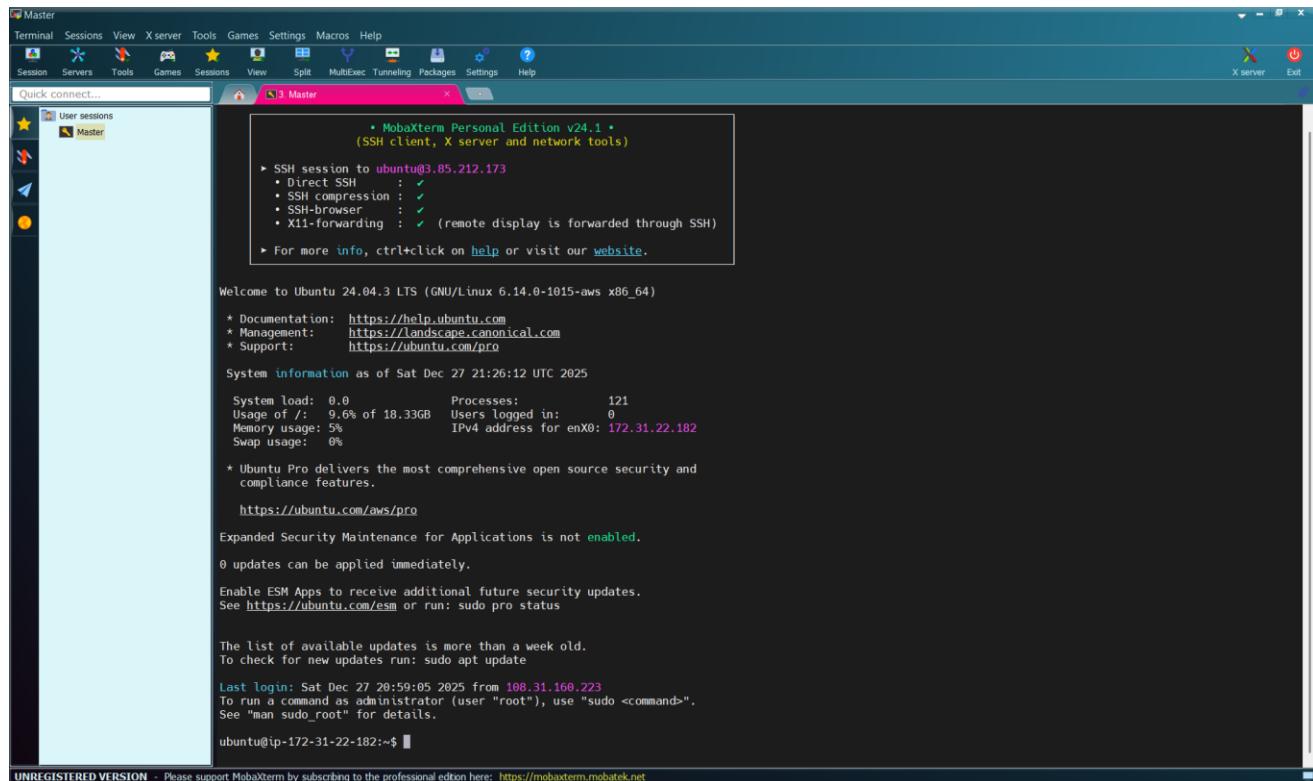
And click on “OK” again



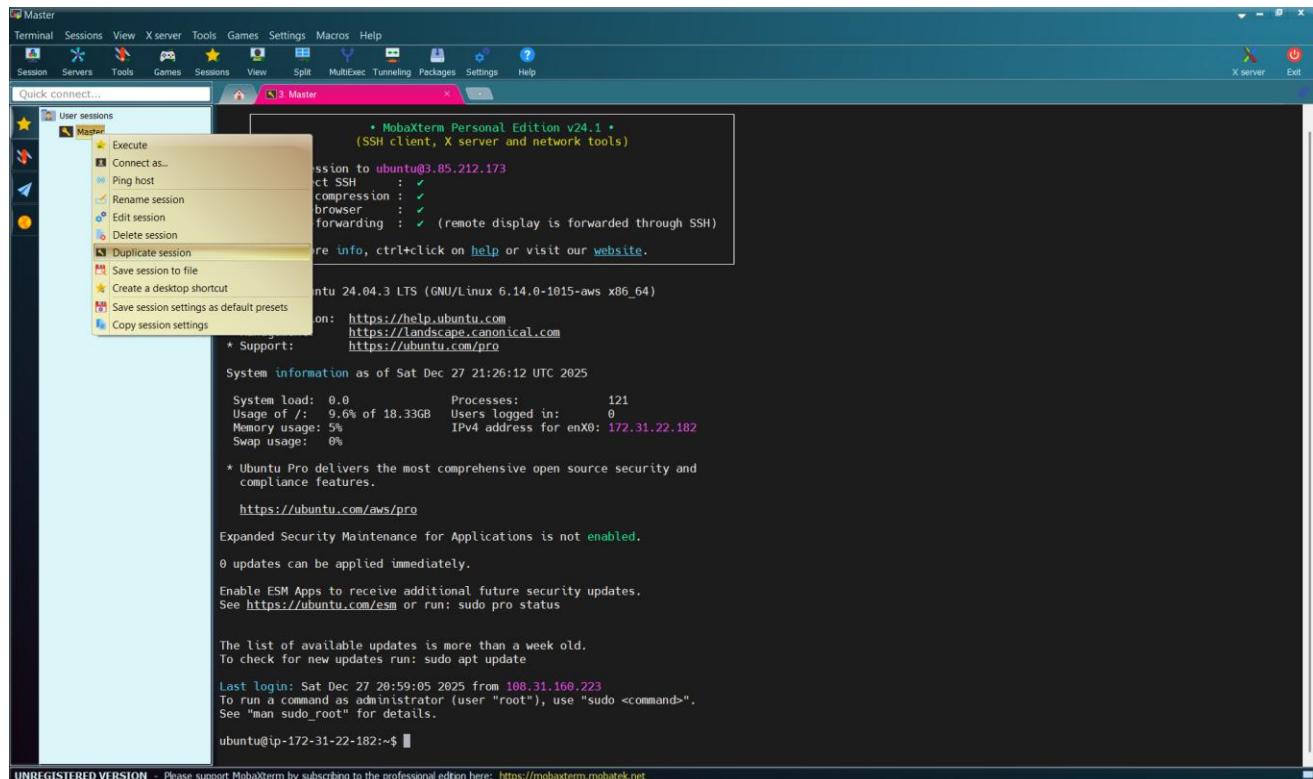
You can see the color of the tab has changed.

### 3.2.2.2 SSH Connect to Virtual Machine “Slave-1”

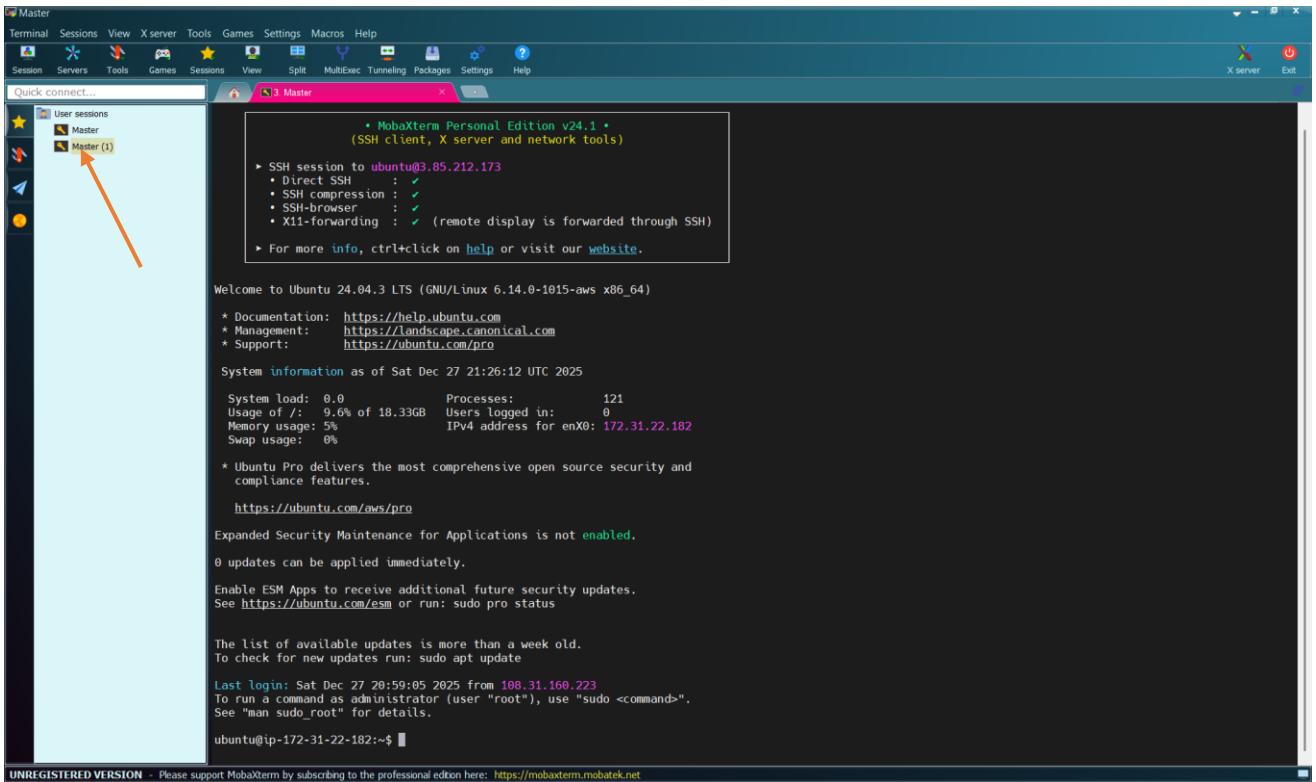
Let us create a duplicate of the session “Master”.



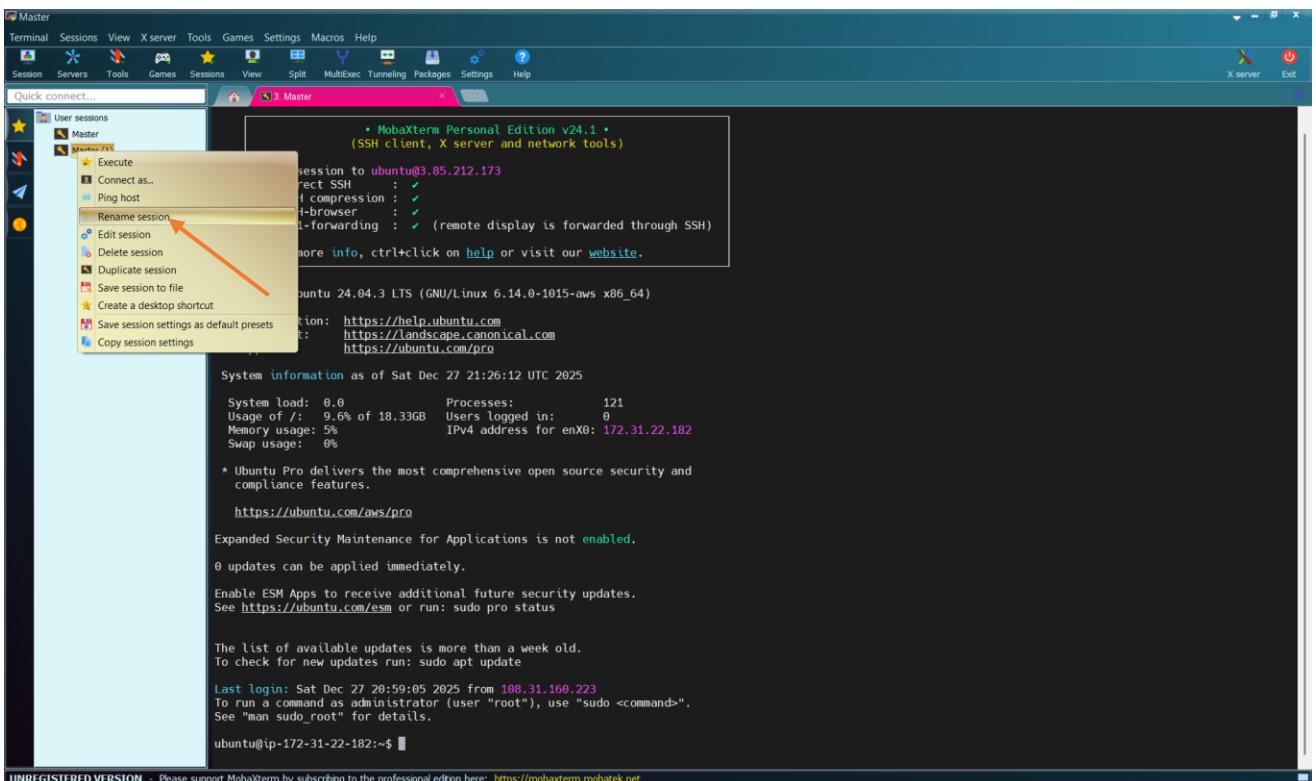
Right-click on the session name “Master”



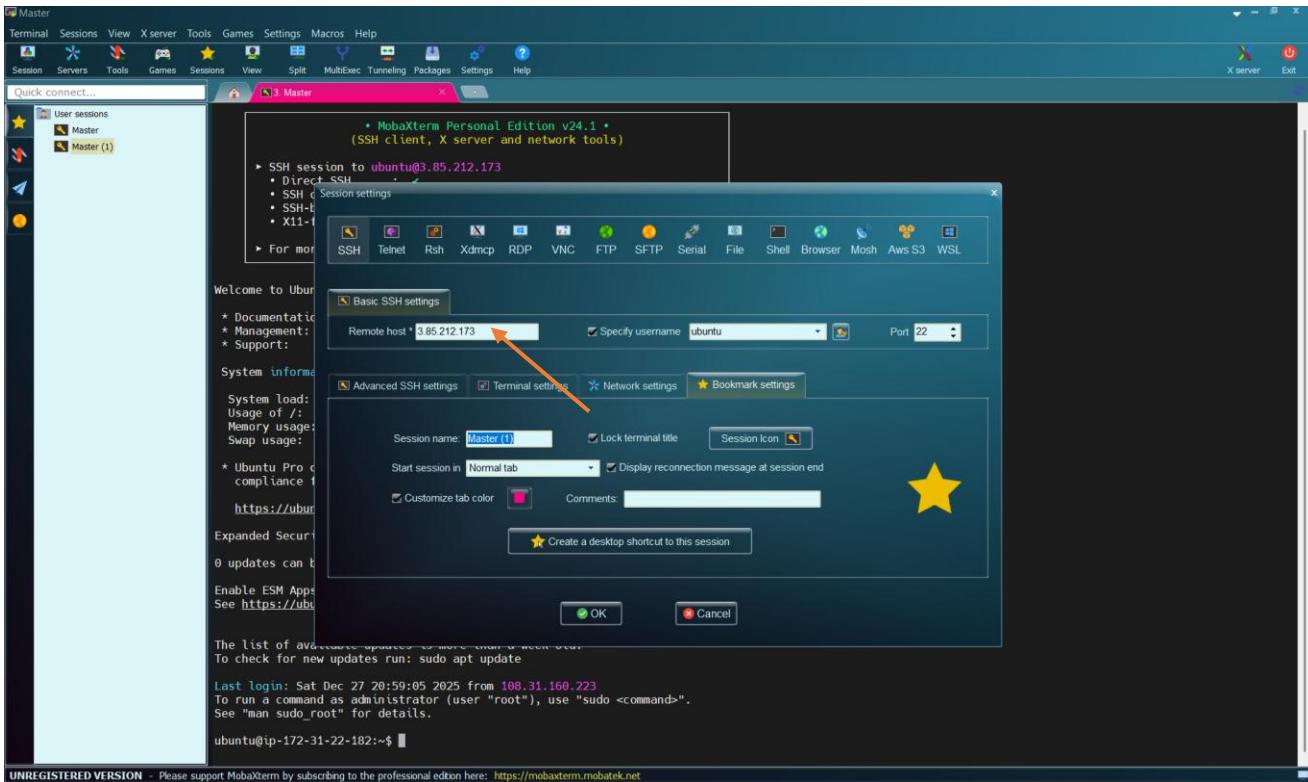
Select “Duplicate Session”



We want to rename “Master(1)” to “Slave-1”. Right-click on “Master(1)”



Select “Rename Session”



Copy the Public IP address of our “Slave-1” virtual machine and paste here.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0b47272282d6961f5	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
<b>Slave-1</b>	i-0618808c6e1829c35	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Slave-2	i-0d0eabfc020a5de7a	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c

**i-0618808c6e1829c35 (Slave-1)**

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

**Instance summary**

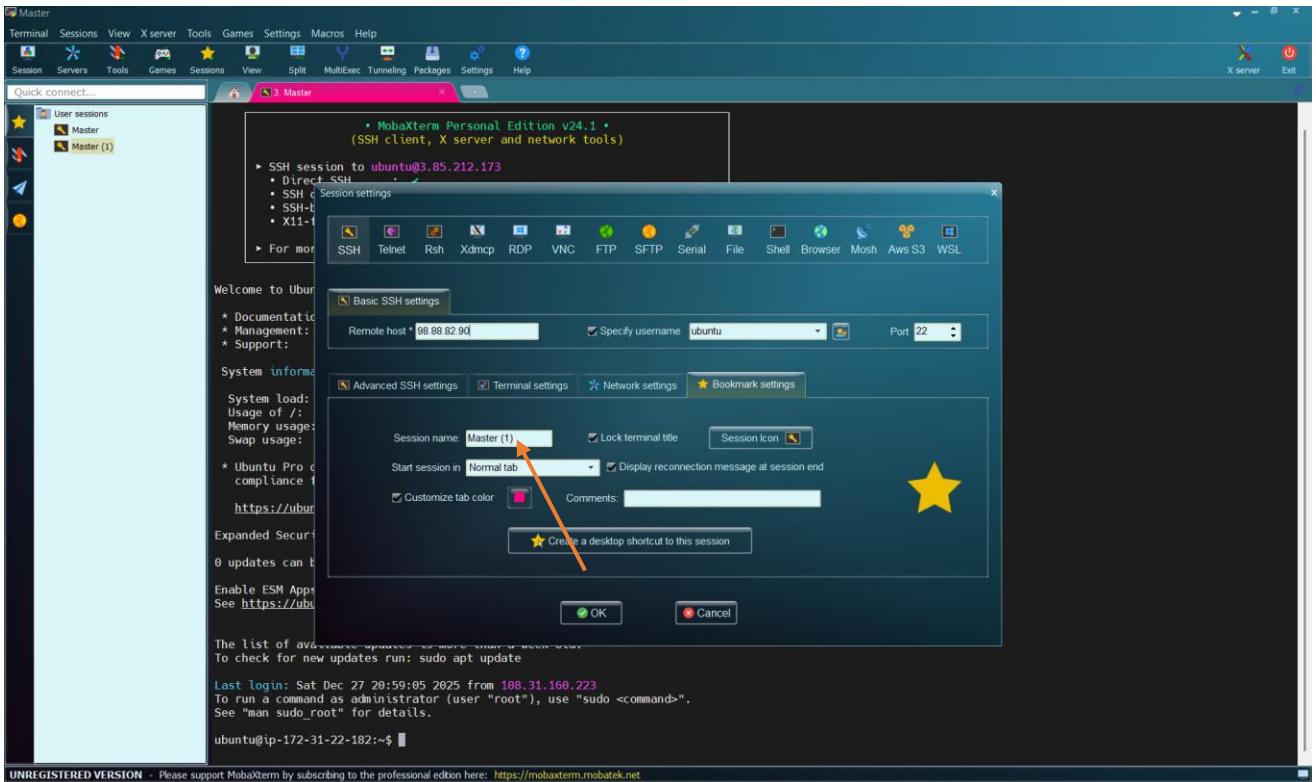
Instance ID: i-0618808c6e1829c35  
 IPV6 address: -  
 Hostname type: IP name: ip-172-31-25-98.ec2.internal

Public IPv4 address: [98.88.82.90 | open address](#)

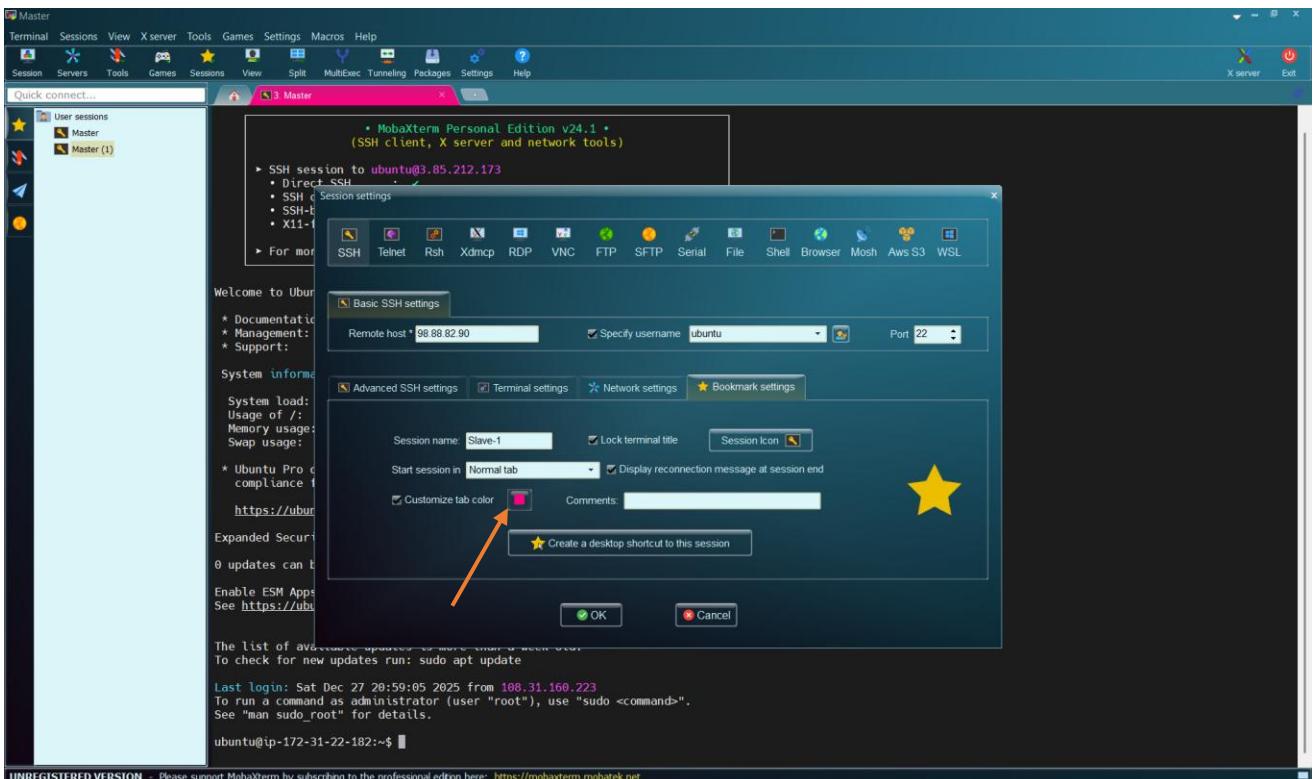
Private IPv4 addresses: 172.31.25.98

Public DNS: ec2-98-88-82-90.compute-1.amazonaws.com | [open address](#)

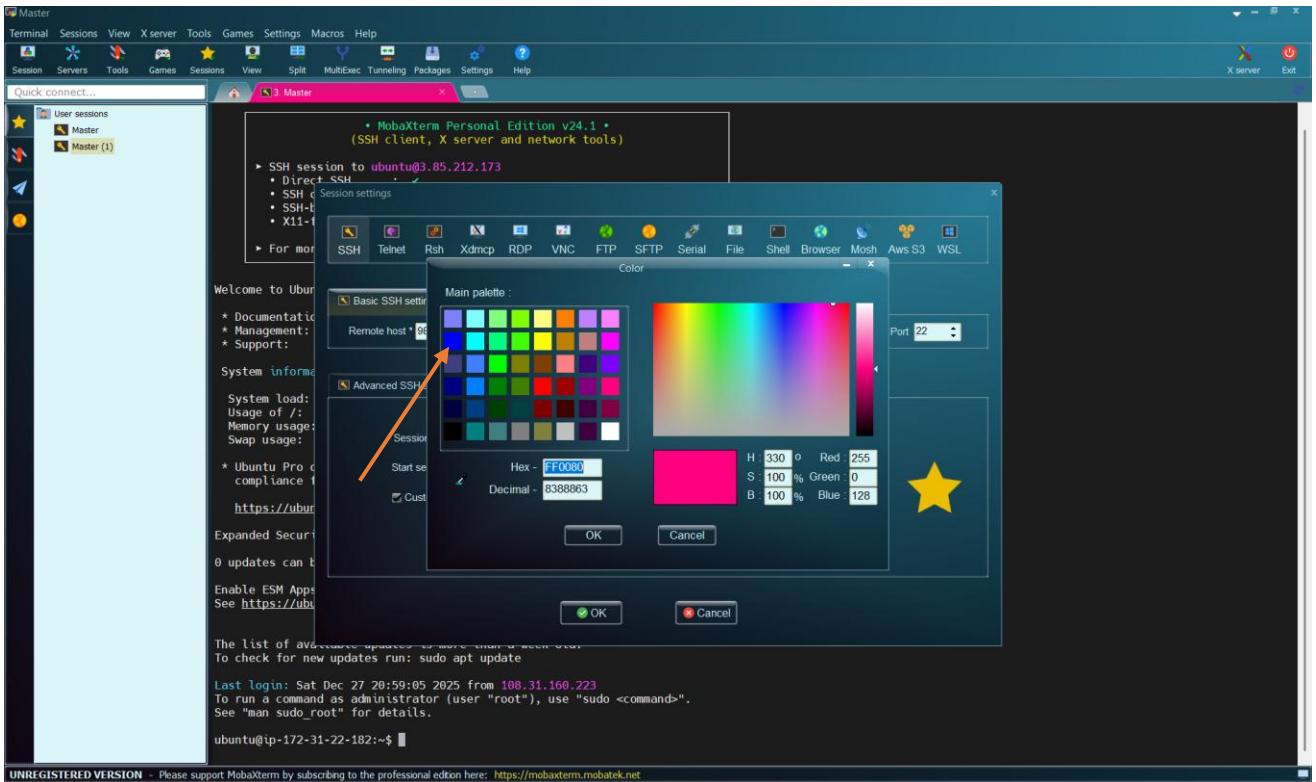
Copy the Public IP: 98.88.82.90 and paste the MobaXterm



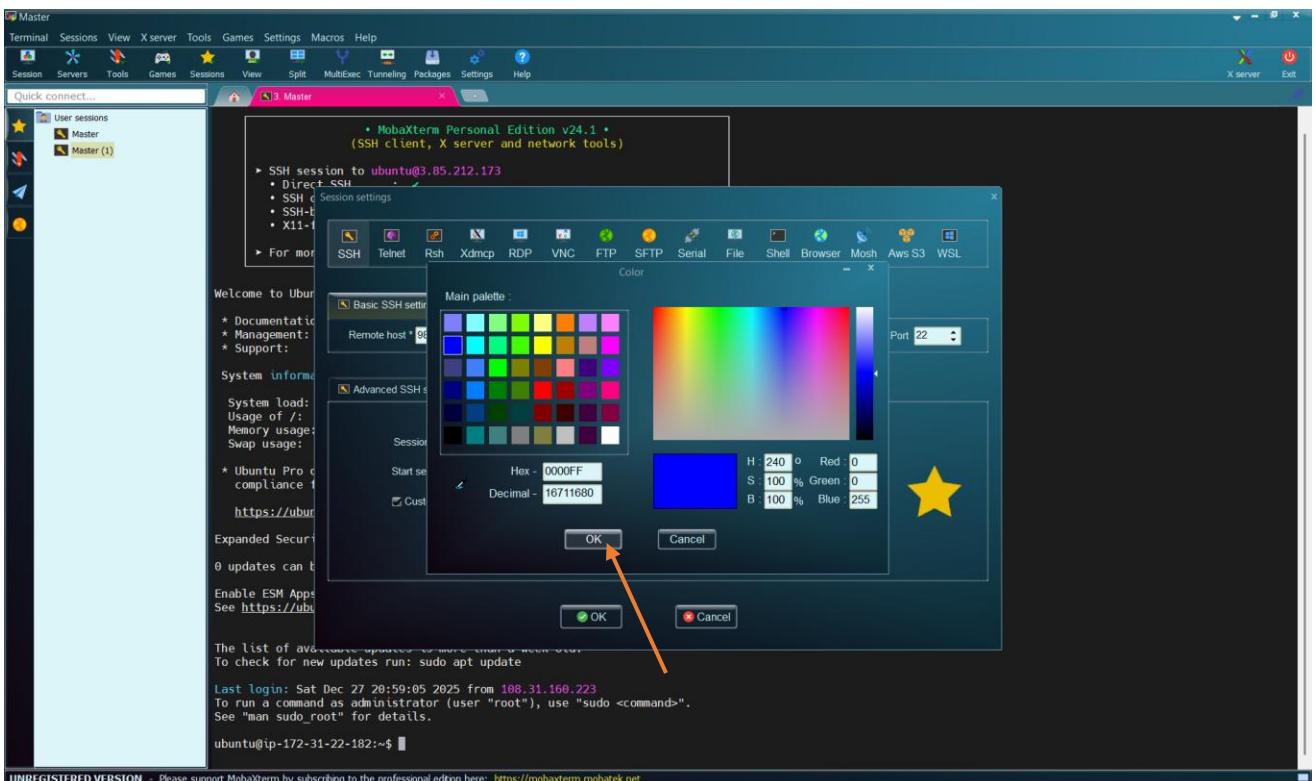
Then, change the name to “Slave-1”



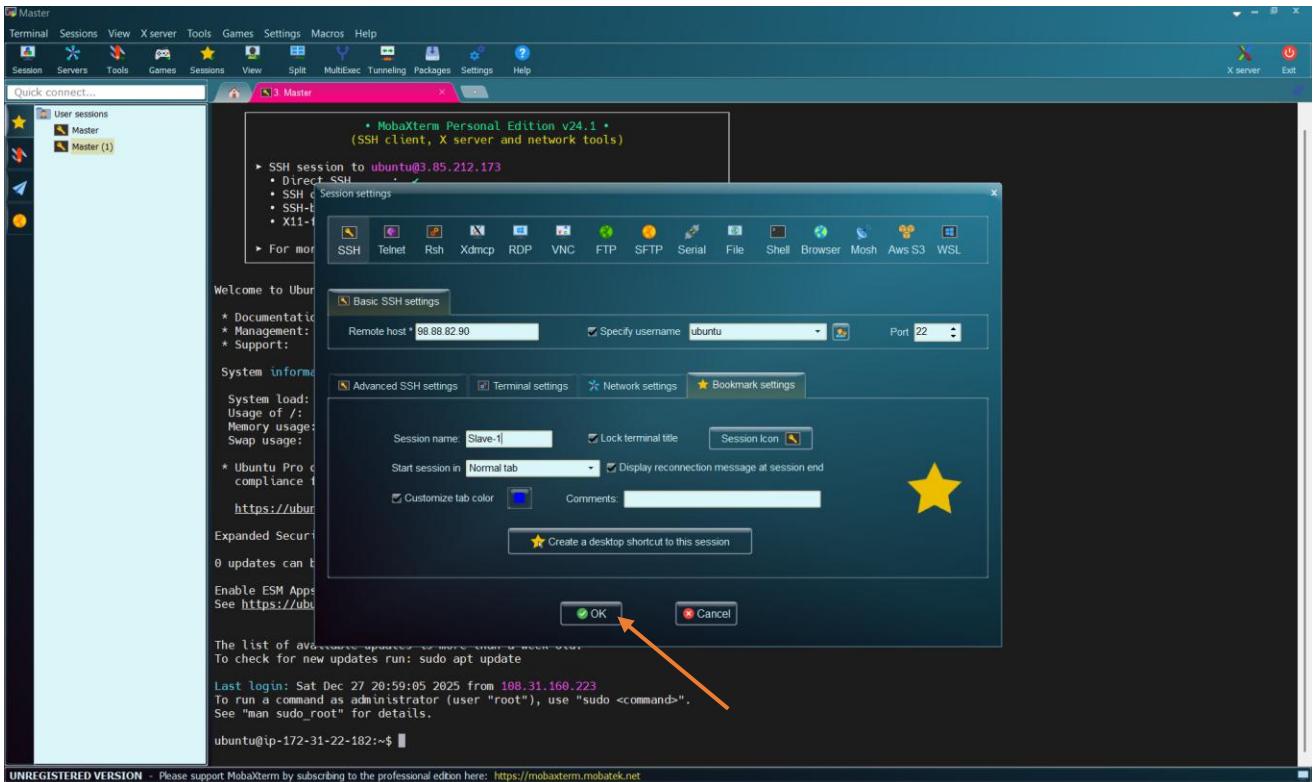
Then, let us change the color. Click there



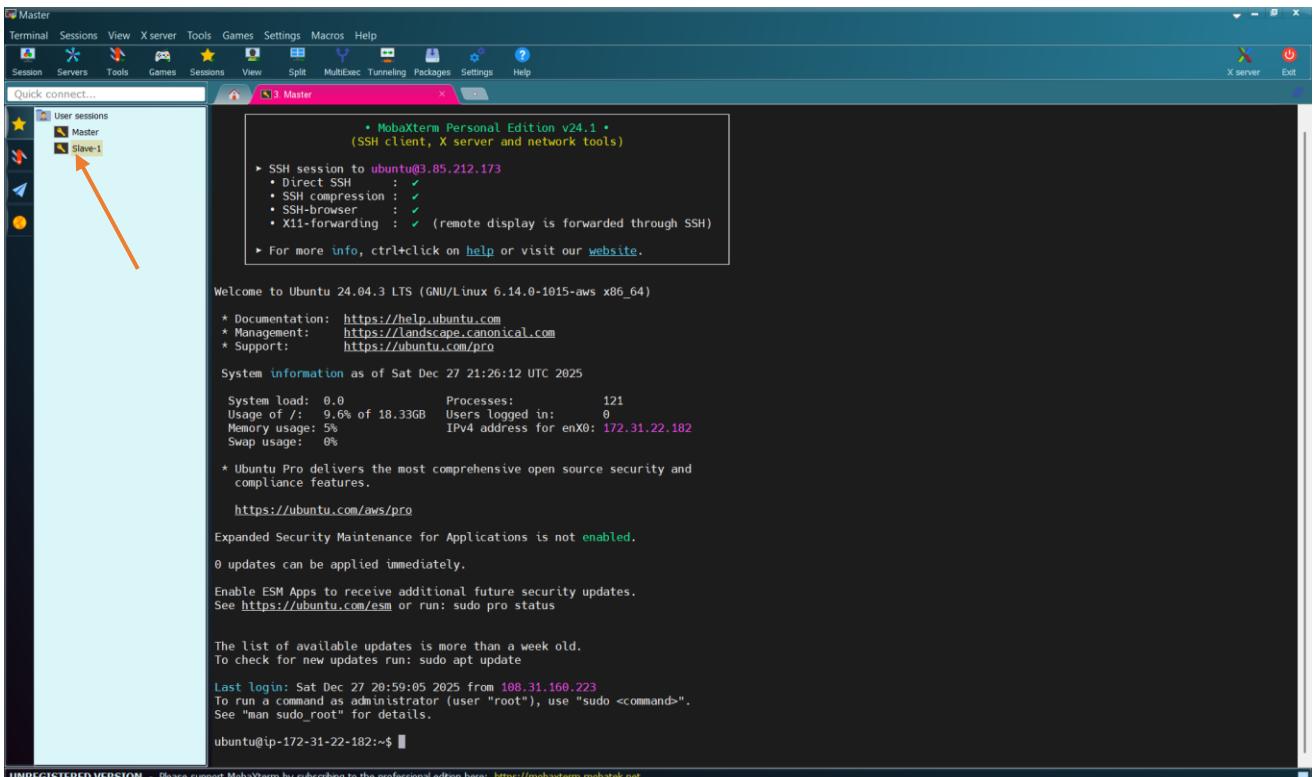
Let us select blue



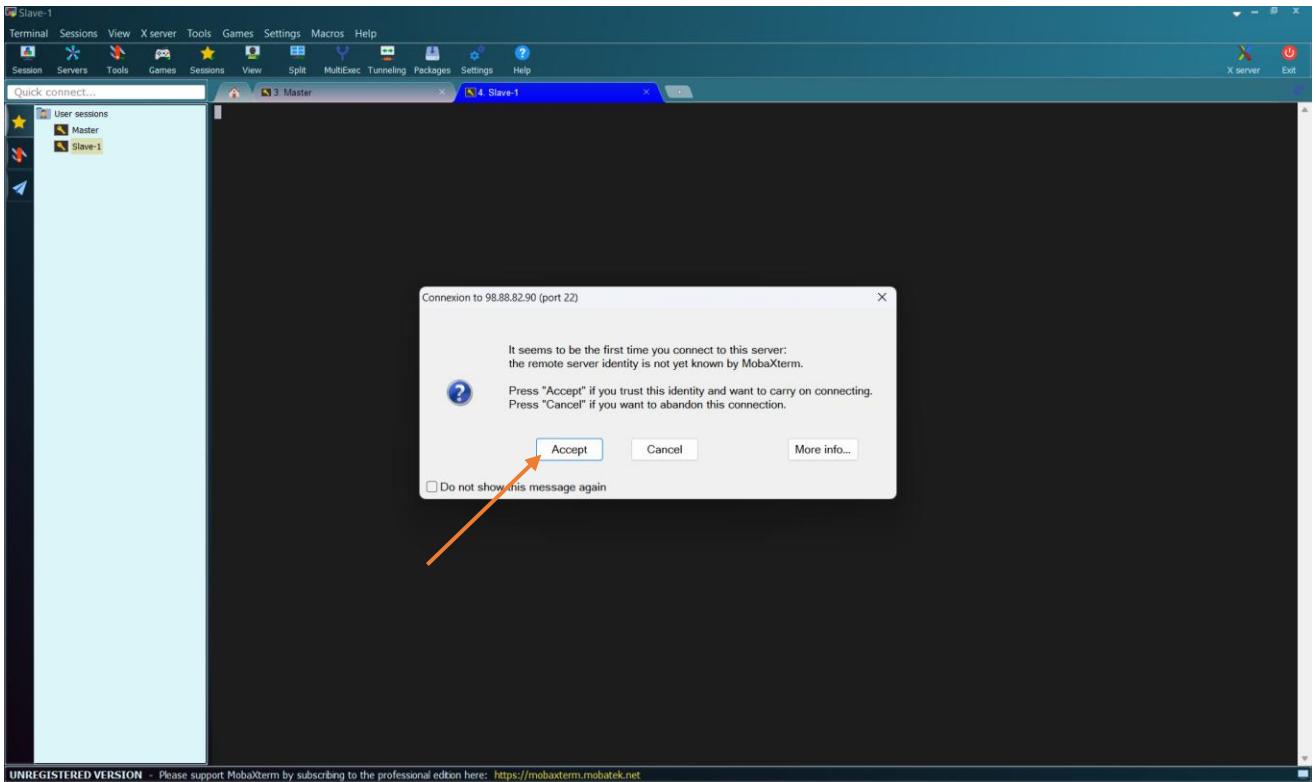
Click on "OK"



And click on “OK” again



Then, double-click on the session “Slave-1”



Click on “Accept”

```

MobaXterm Personal Edition v24.1
(SSH client, X server and network tools)

SSH session to ubuntu@98.88.82.90
  • Direct SSH : ✓
  • SSH compression : ✓
  • SSH-browser : ✓
  • X11-forwarding : ✓ (remote display is forwarded through SSH)
  • For more info, ctrl+click on help or visit our website.

Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Sat Dec 27 21:43:18 UTC 2025

System load: 0.0 Processes: 124
Usage of /: 9.5% of 18.33GB Users logged in: 0
Memory usage: 6% IPv4 address for enx0: 172.31.25.98
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

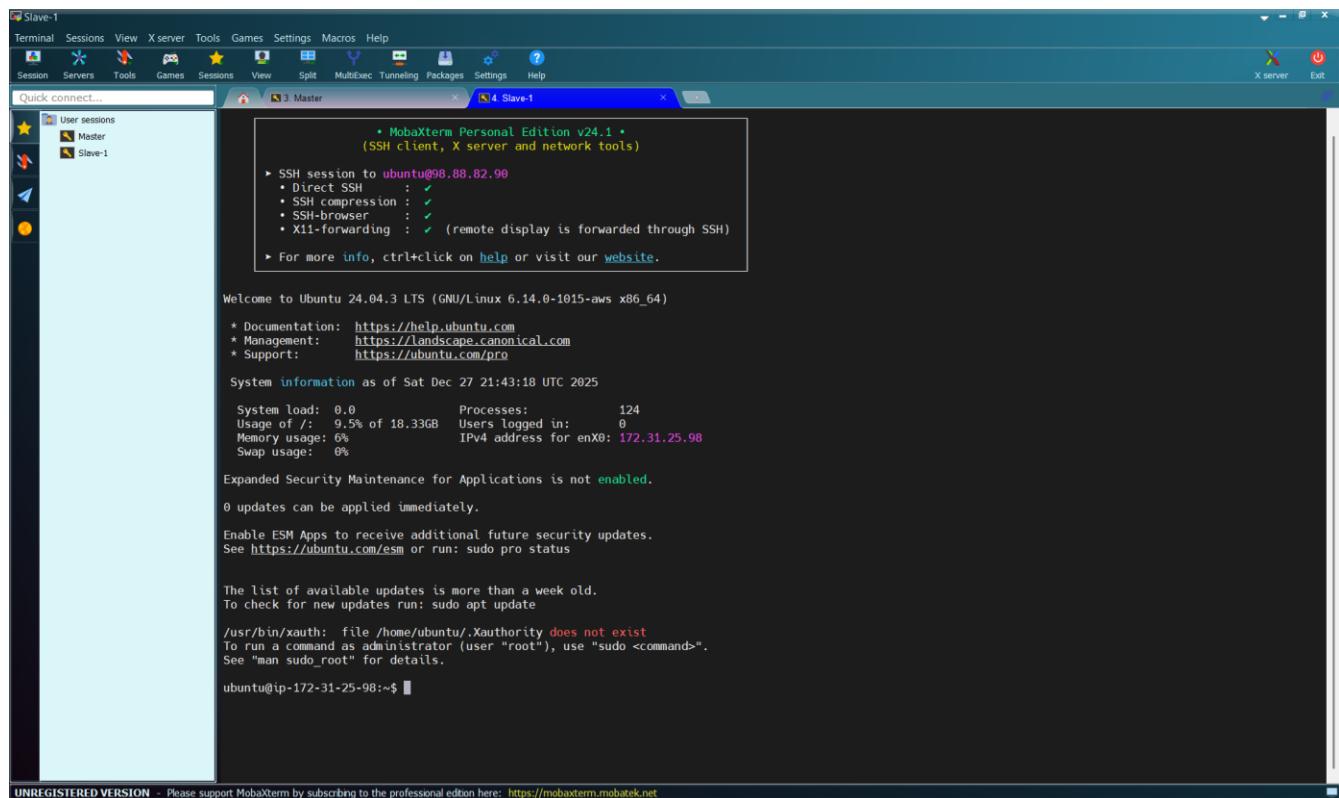
ubuntu@ip-172-31-25-98:~$ 

```

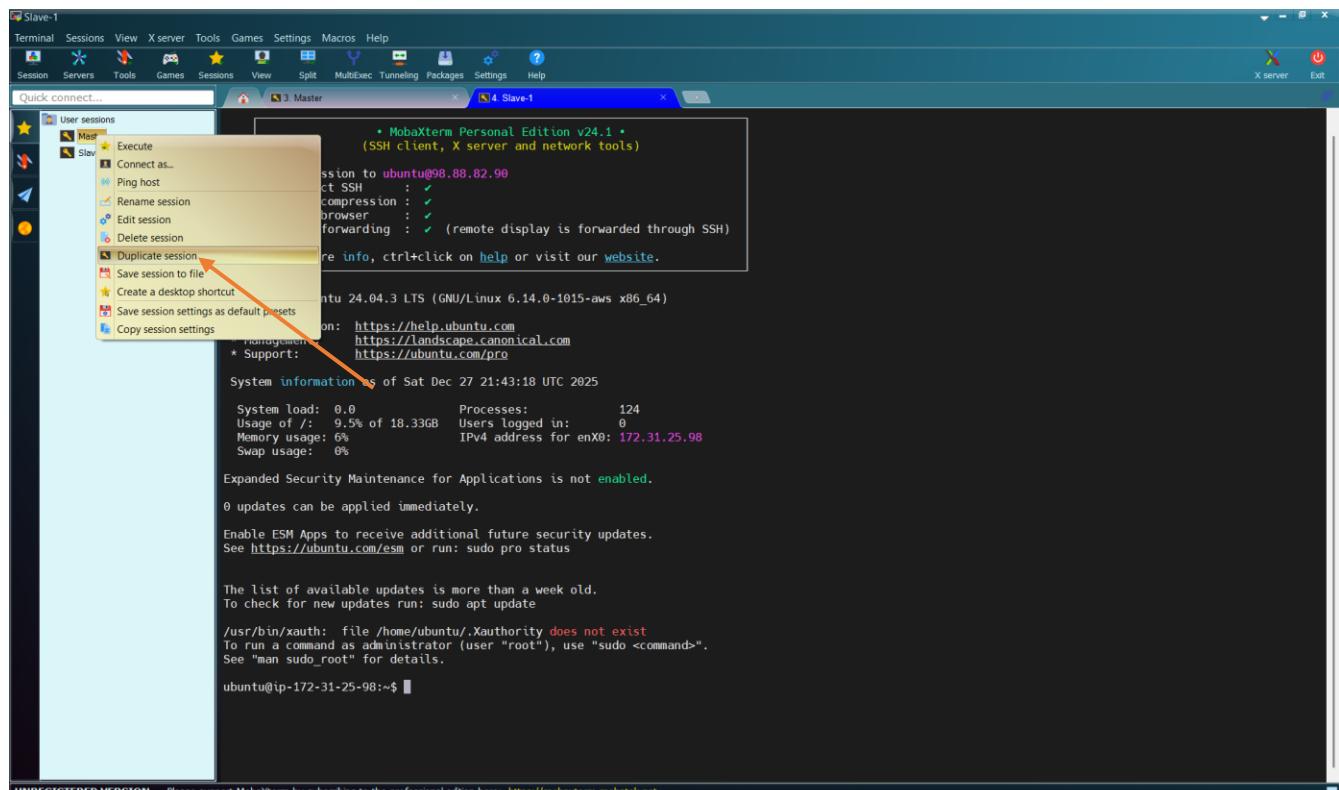
You can see that the tab of “Slave-1” is blue.

### 3.2.2.3 SSH Connect to Virtual Machine “Slave-2”

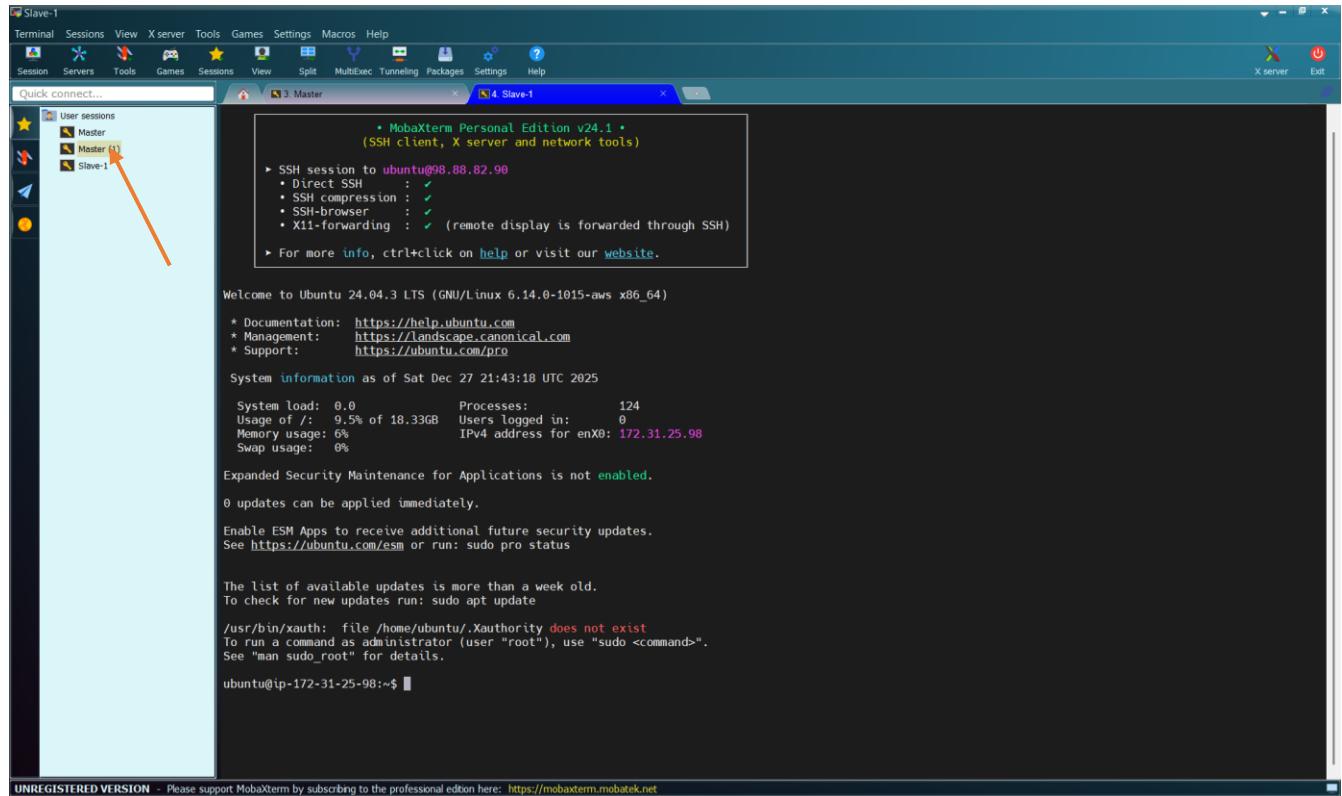
Click on the star



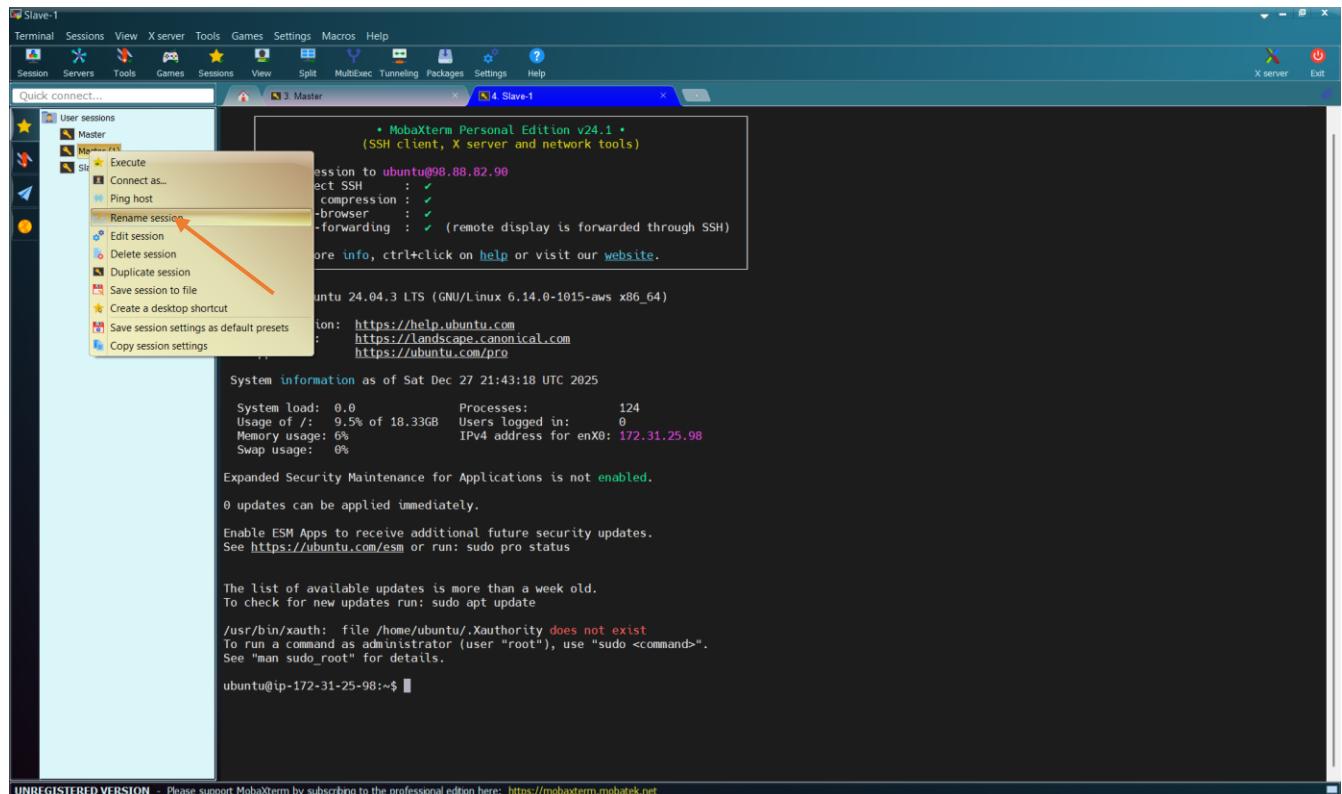
Let us create a duplicate of session “Master”. Right-click on the session name “Master”



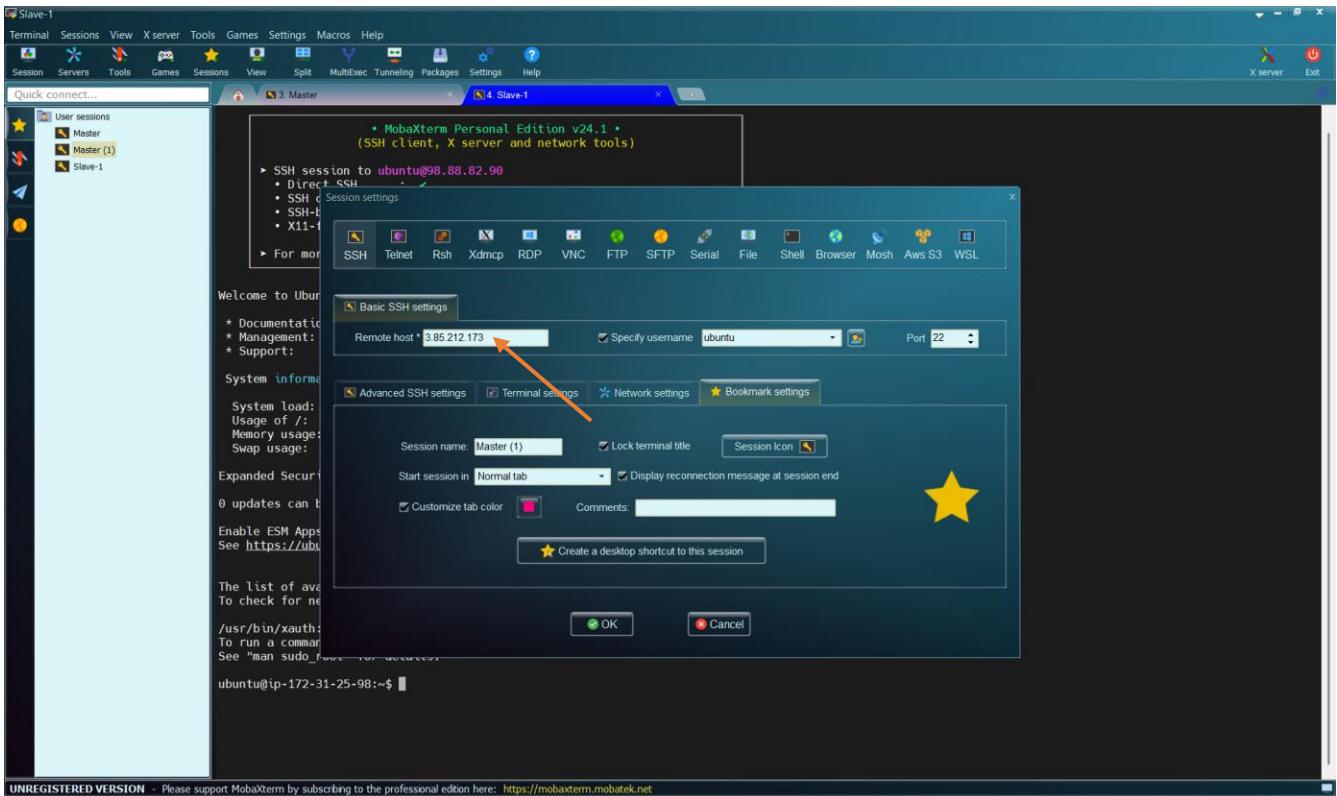
## Select “Duplicate Session”



We want to rename “Master(1)” to “Slave-2”. Right-click on “Master(1)”



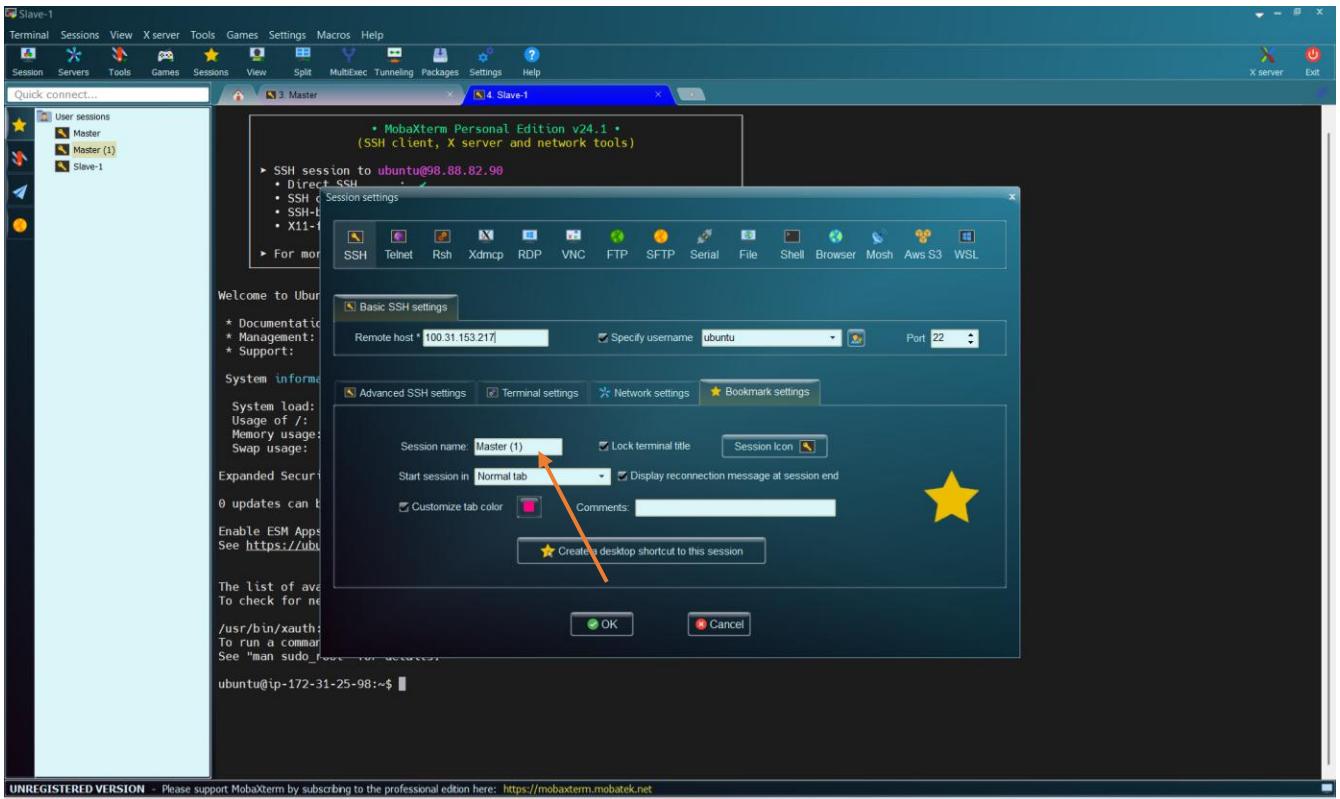
Select “Rename Session”



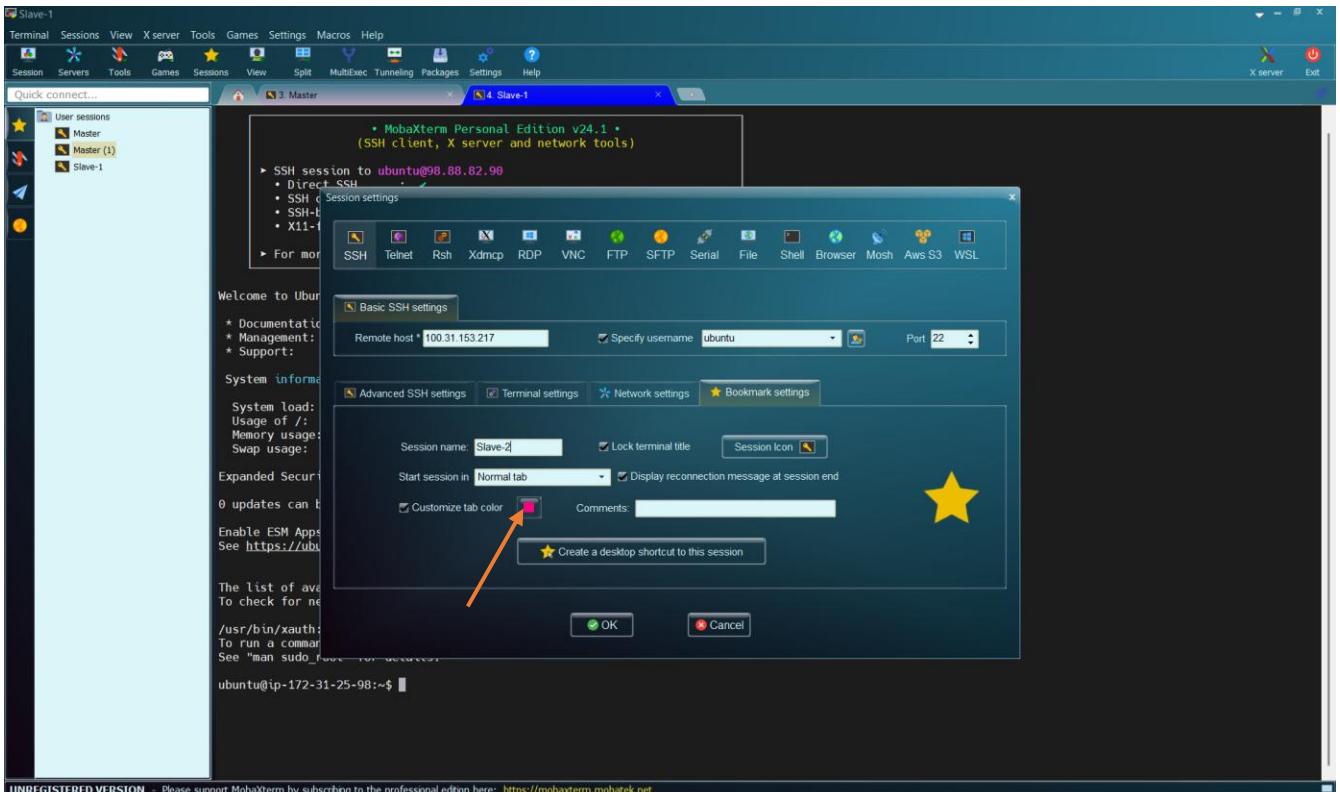
Copy the Public IP address of our “Slave-2” virtual machine and paste here.

The screenshot shows the AWS EC2 Instances page. The left sidebar shows "EC2" selected under "Instances". The main table lists three instances: "Master" (running), "Slave-1" (running), and "Slave-2" (running). A red arrow points to the "Instance state" column for Slave-2, which is highlighted in green with the word "Running". The instance ID for Slave-2 is shown as "i-0d0eabfc020a5de7a". The "Details" tab is selected for the Slave-2 instance. The "Instance summary" section shows the "Public IPv4 address" as "100.31.153.217" and the "Instance state" as "Running".

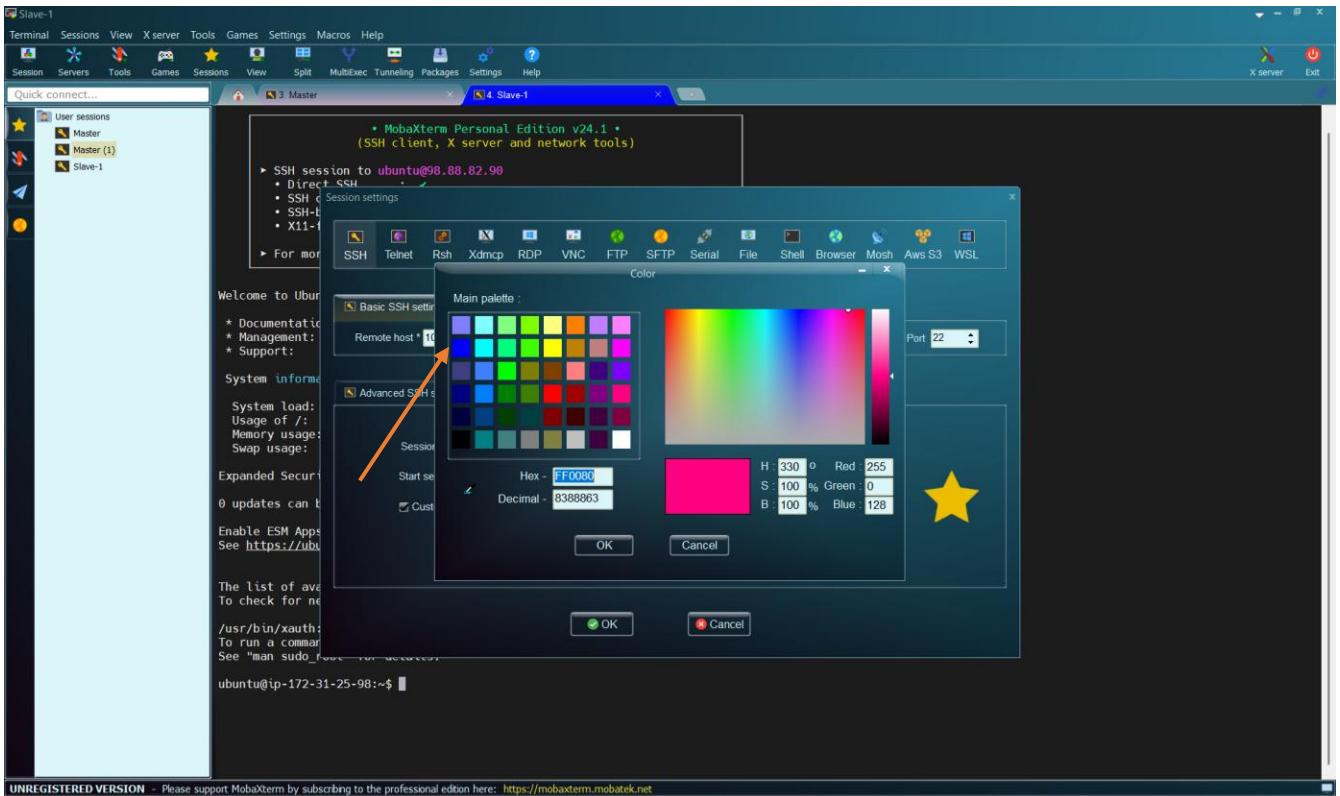
Copy the Public IP: **100.31.153.217** and paste the MobaXterm



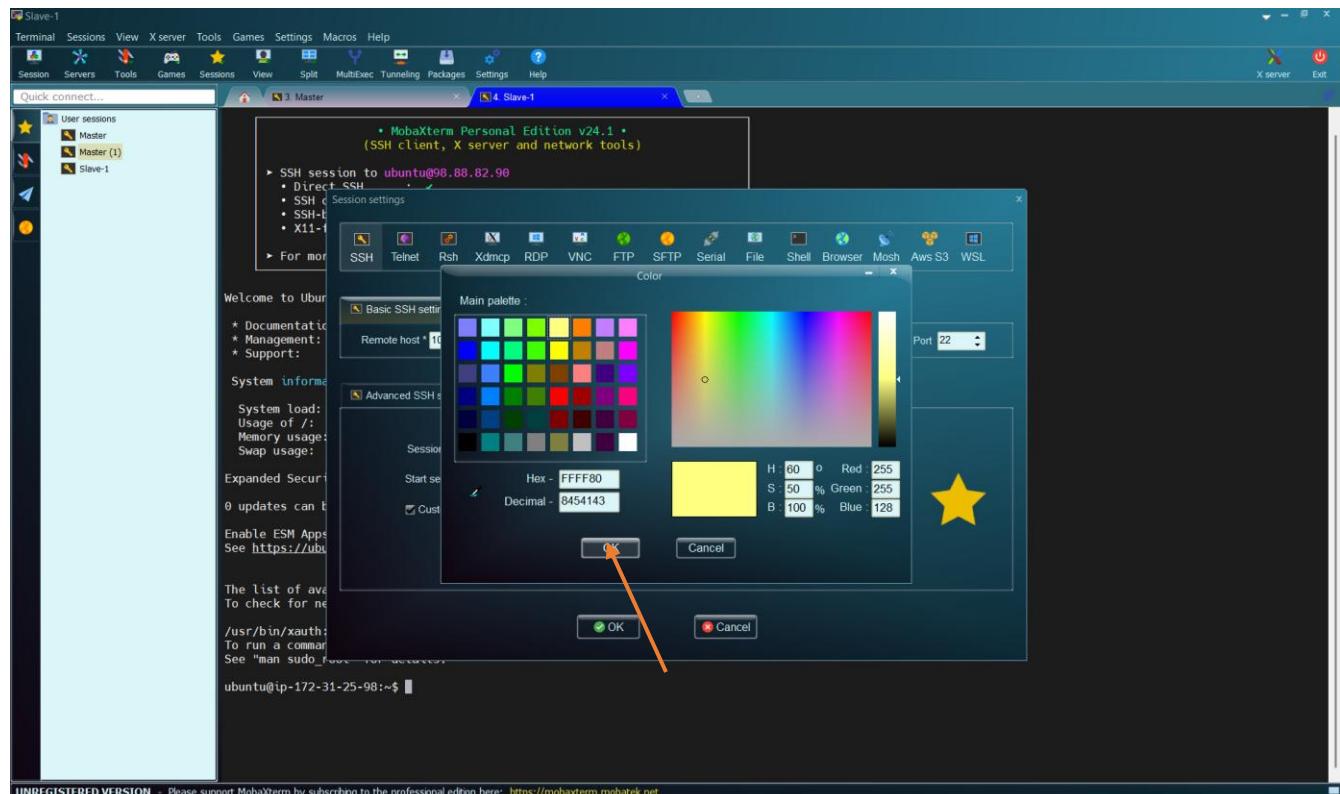
Then, change the name to “Slave-2”



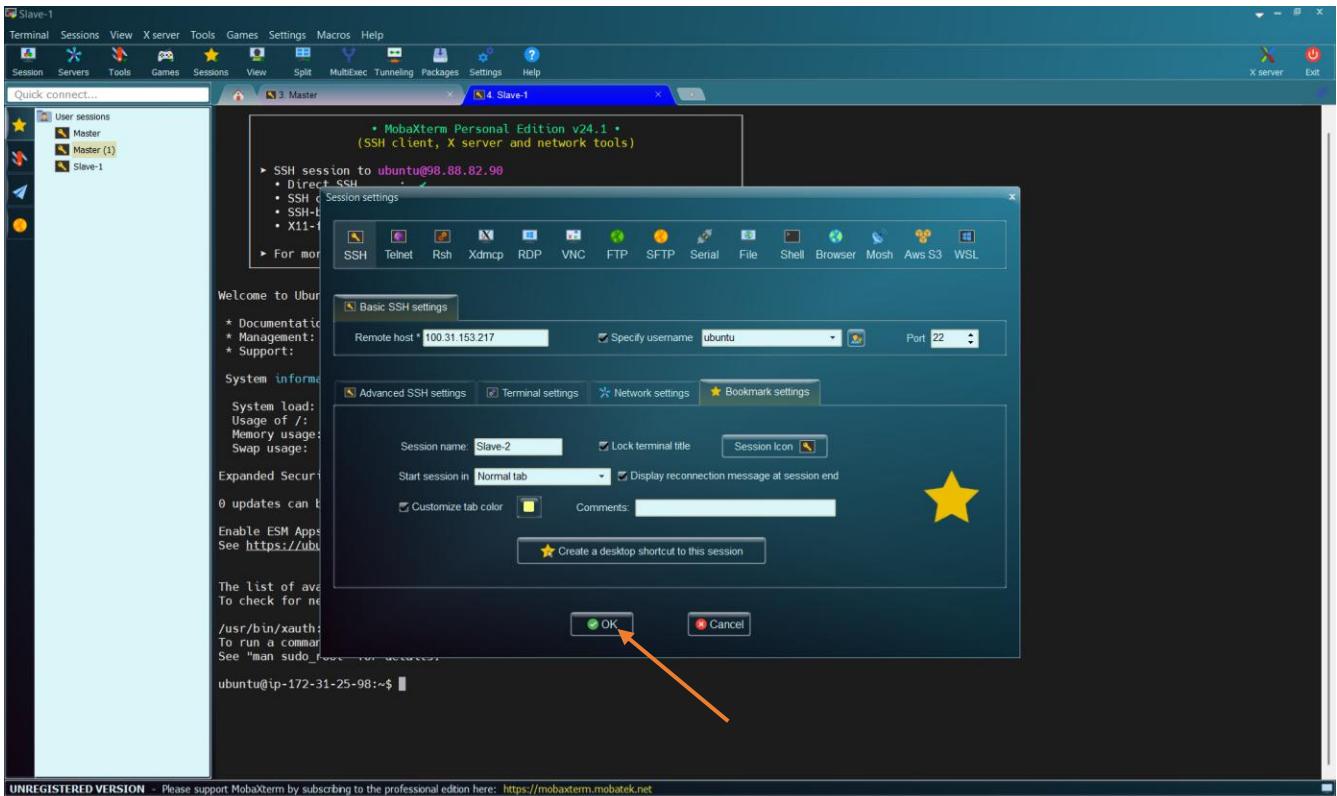
Then, let us change the color. Click there



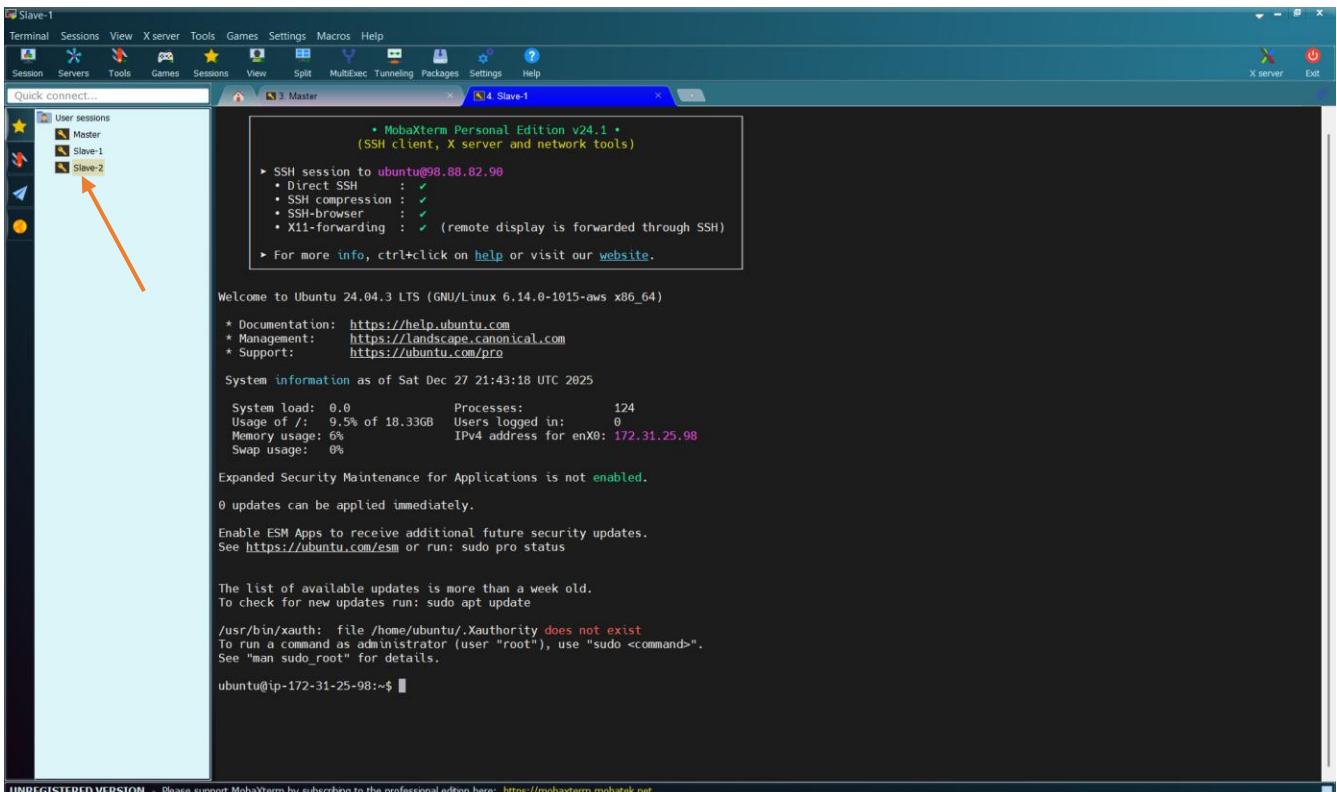
Let us select "yellow"



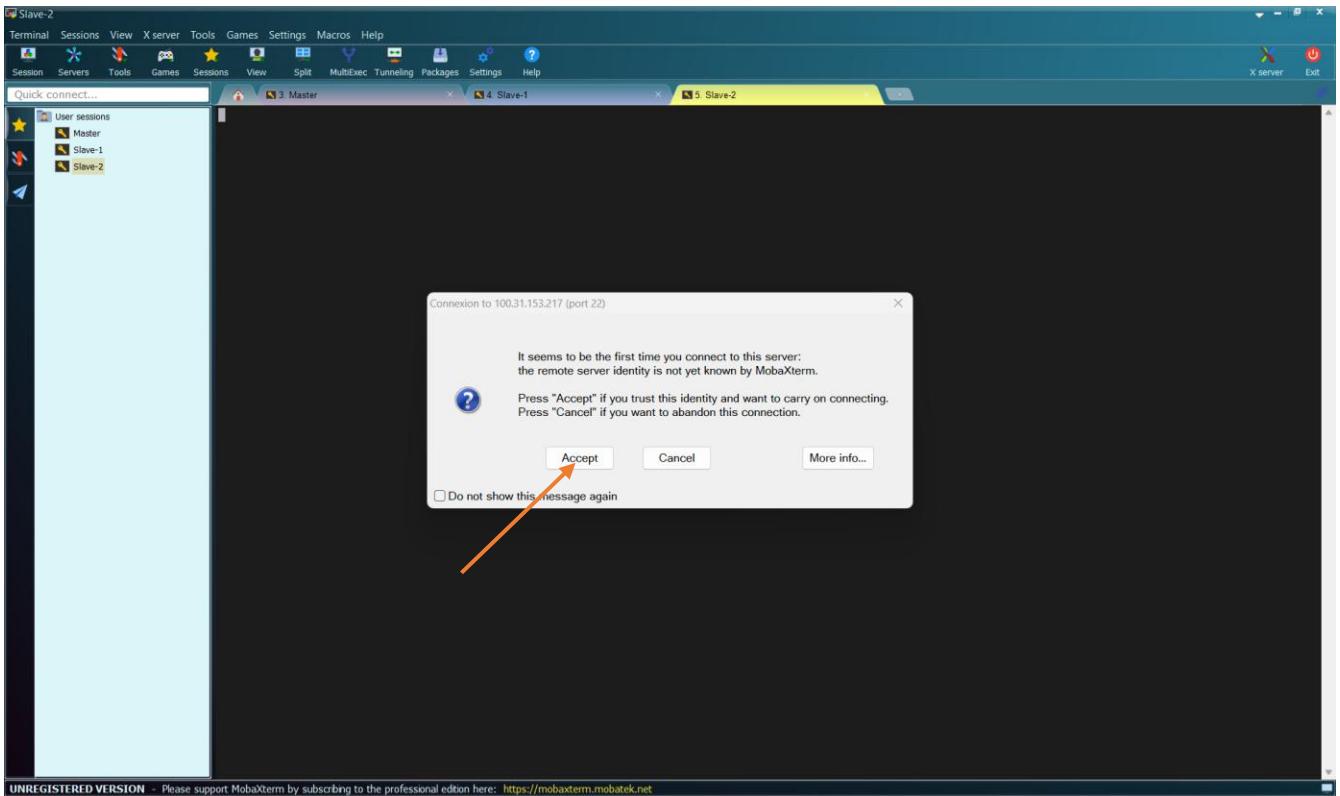
Click on "OK"



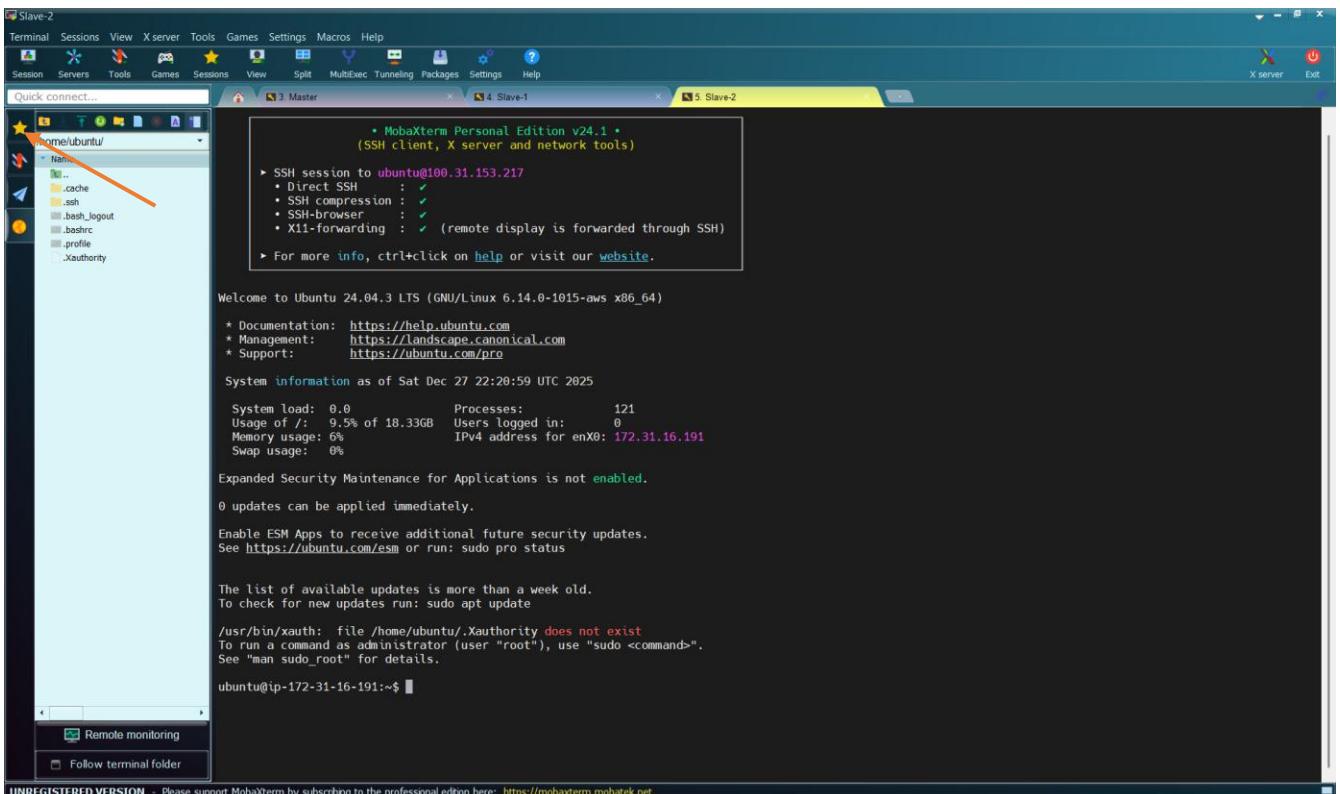
And click on “OK” again



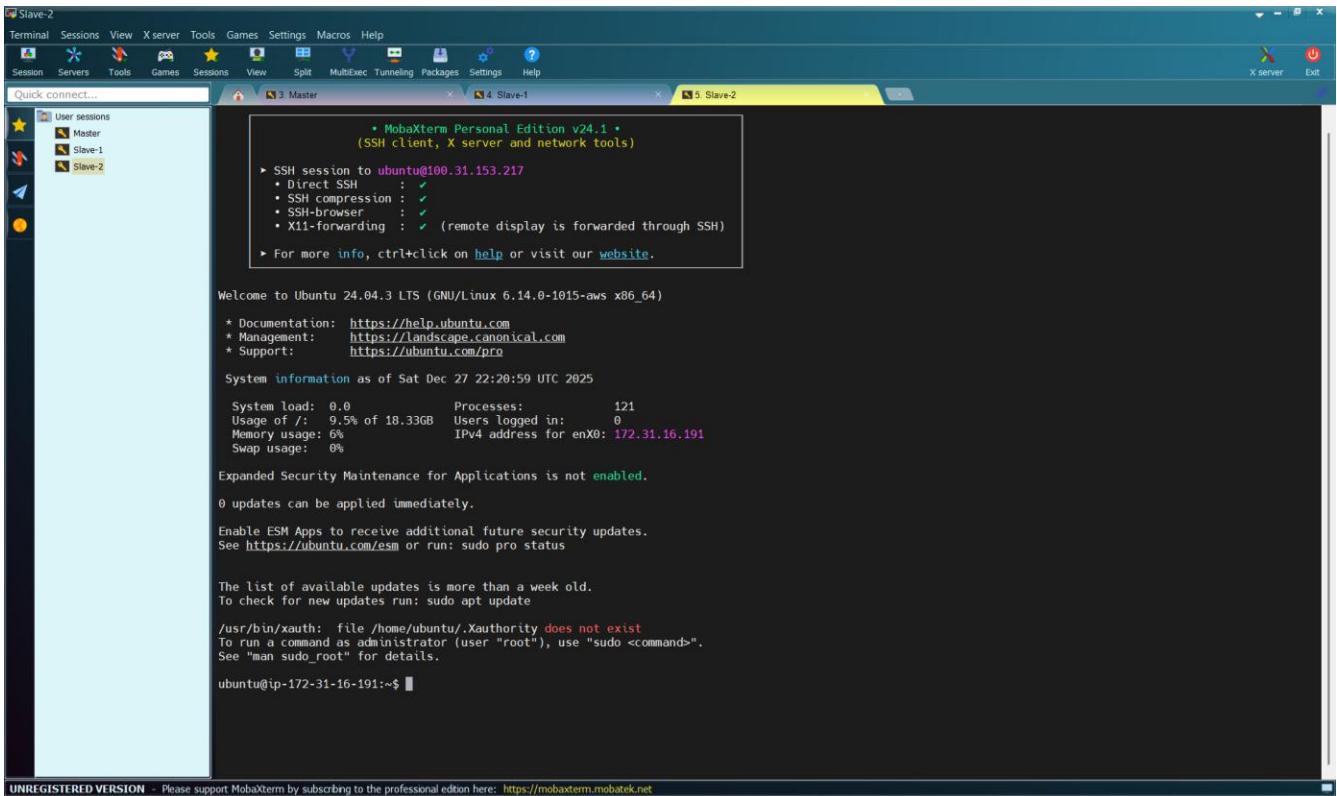
Then, double-click on the session “Slave-2”



Click on “Accept”



You can see that the tab of “Slave-2” is yellow. Click on the star.



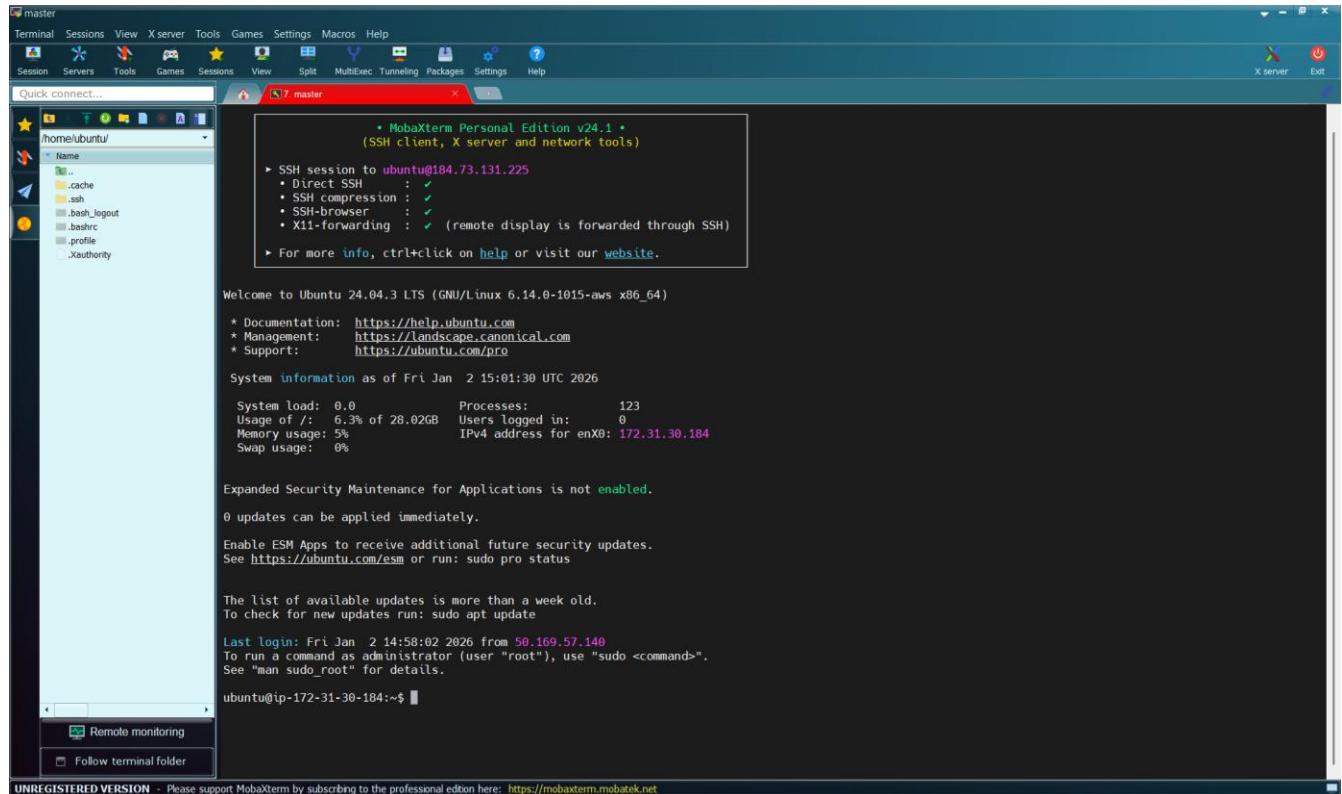
We have created the session “Slave-2”

### 3.3 Set up Kubernetes Cluster using kubeadm

Here, we are going to set up the Kubernetes cluster or any other deployment cluster where we are going to deploy our application and scan for vulnerability.

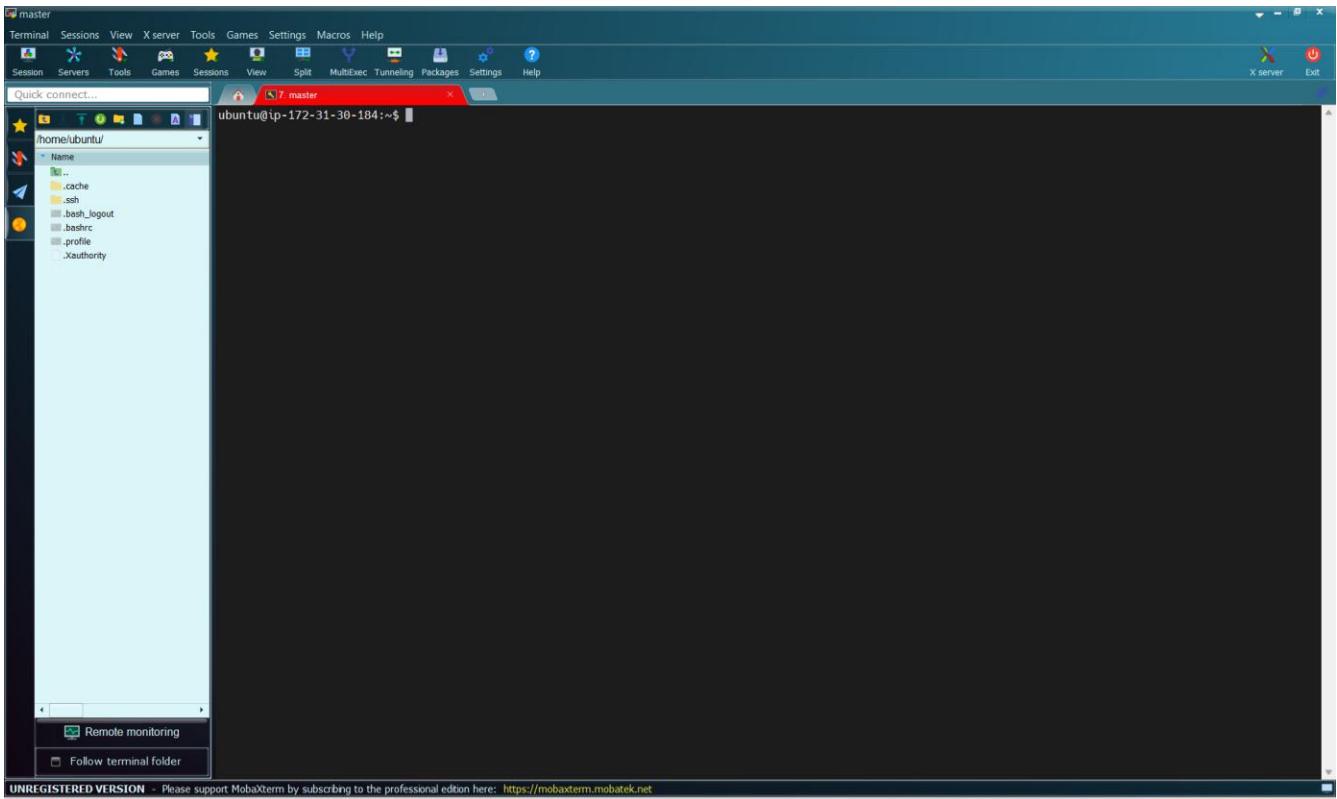
#### 3.3.1 Set up Kubernetes on Virtual Machine “Master”

We have to configure the virtual machine “Master”. Open the session “Master”



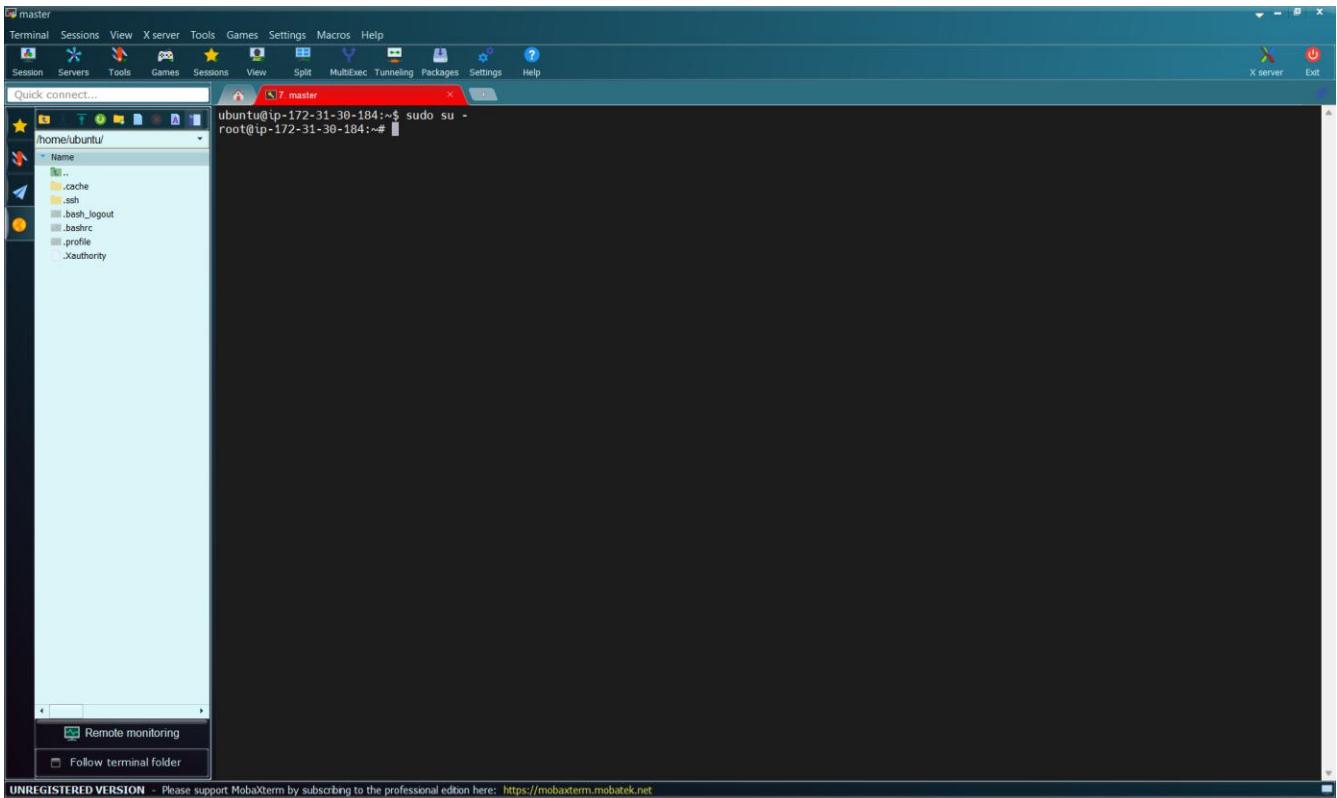
Clear the terminal using the command:

```
clear
```



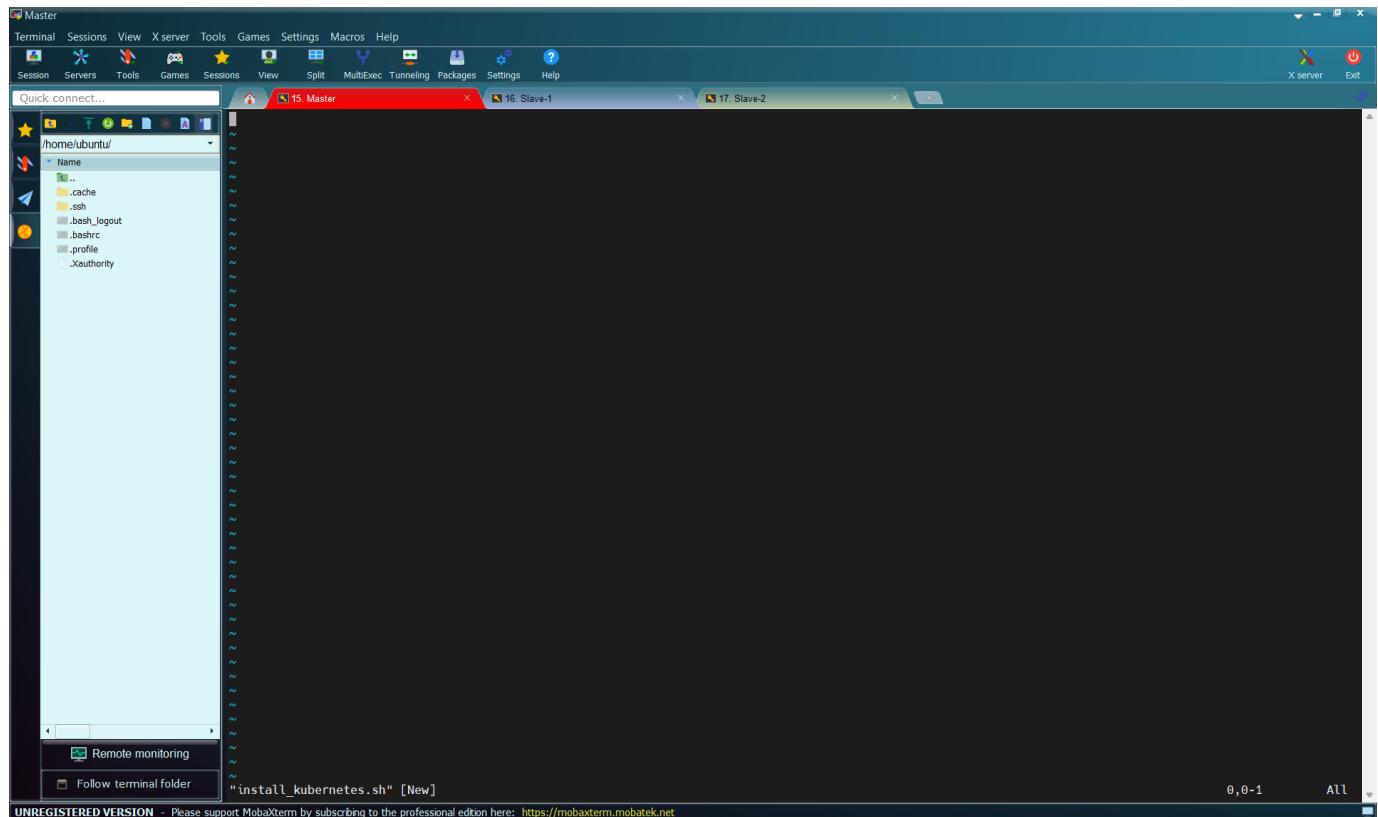
Then, let us give root user access using the command:

```
sudo su -
```



Now, let us install Kubernetes Components. We will do this by using a script. We will call the script "**install\_kubernetes.sh**". Let us create the file using the command:

```
vi install_kubernetes.sh
```



Paste these commands in the terminal

```
#!/bin/bash

curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt) /bin/linux/amd64/kubectl"
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt) /bin/linux/amd64/kubectl.sha256"
echo "$(cat kubectl.sha256)  kubectl" | sha256sum --check
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
chmod +x kubectl
mkdir -p ~/.local/bin
mv ./kubectl ~/.local/bin/kubectl

# and then append (or prepend) ~/.local/bin to $PATH
kubectl version --client

# disable swap
sudo swapoff -a

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
```

```

sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRI-O Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee /etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o

sudo systemctl daemon-reload
sudo systemctl enable crio --now
sudo systemctl start crio.service

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq

sudo systemctl enable --now kubelet
sudo systemctl start kubelet

```

```

Master
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15 Master 16 Slave-1 17 Slave-2
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRI-O Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee /etc/apt/sources.list.d/crio.list

sudo apt-get update -y
sudo apt-get install -y cri-o

sudo systemctl daemon-reload
sudo systemctl enable cri-o --now
sudo systemctl start cri-o.service

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /" | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq

sudo systemctl enable --now kubelet
sudo systemctl start kubelet

```

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

Save the file by pressing “**ESC**” followed by typing “**:wq**” and press “**Enter**”

```

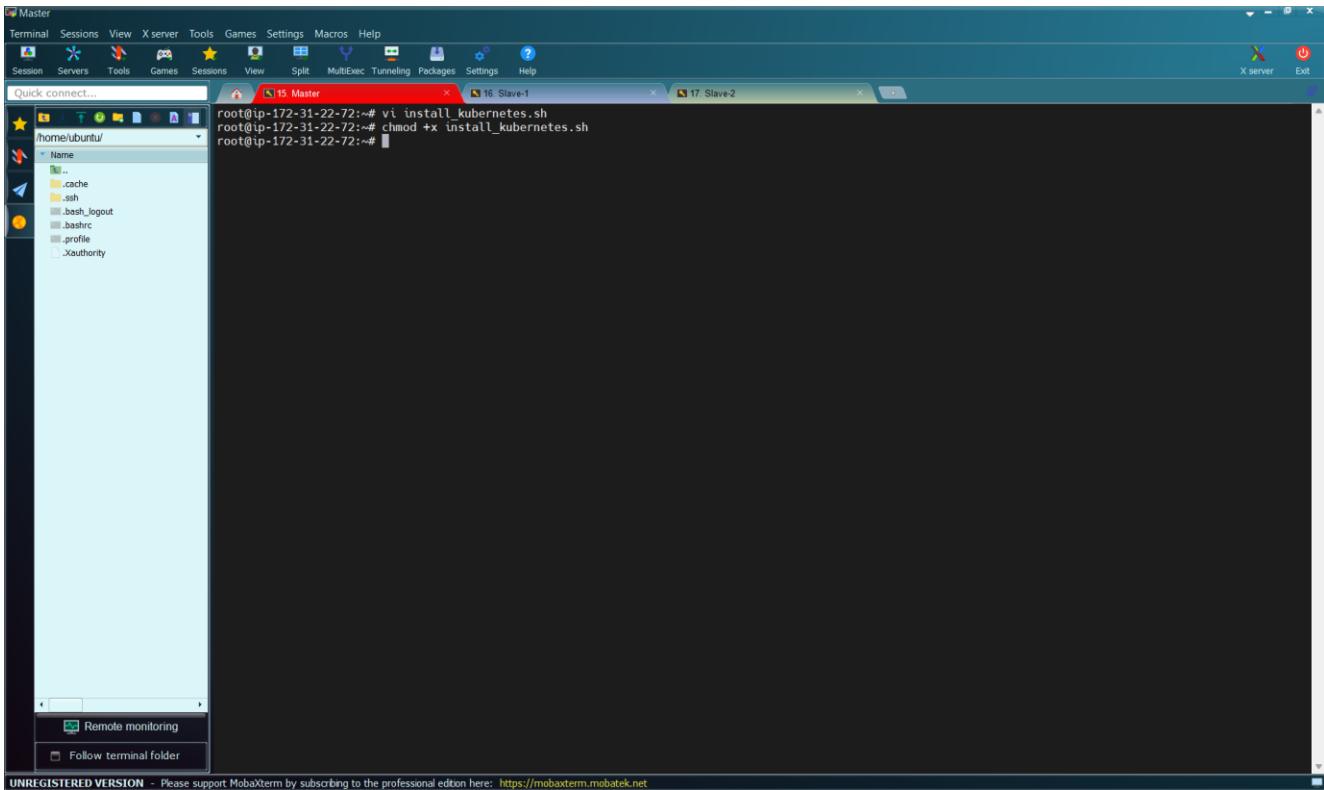
Master
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15 Master 16 Slave-1 17 Slave-2
root@ip-172-31-22-72:~# vi install_kubernetes.sh
root@ip-172-31-22-72:~#

```

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

Give the file executable permission using the command:

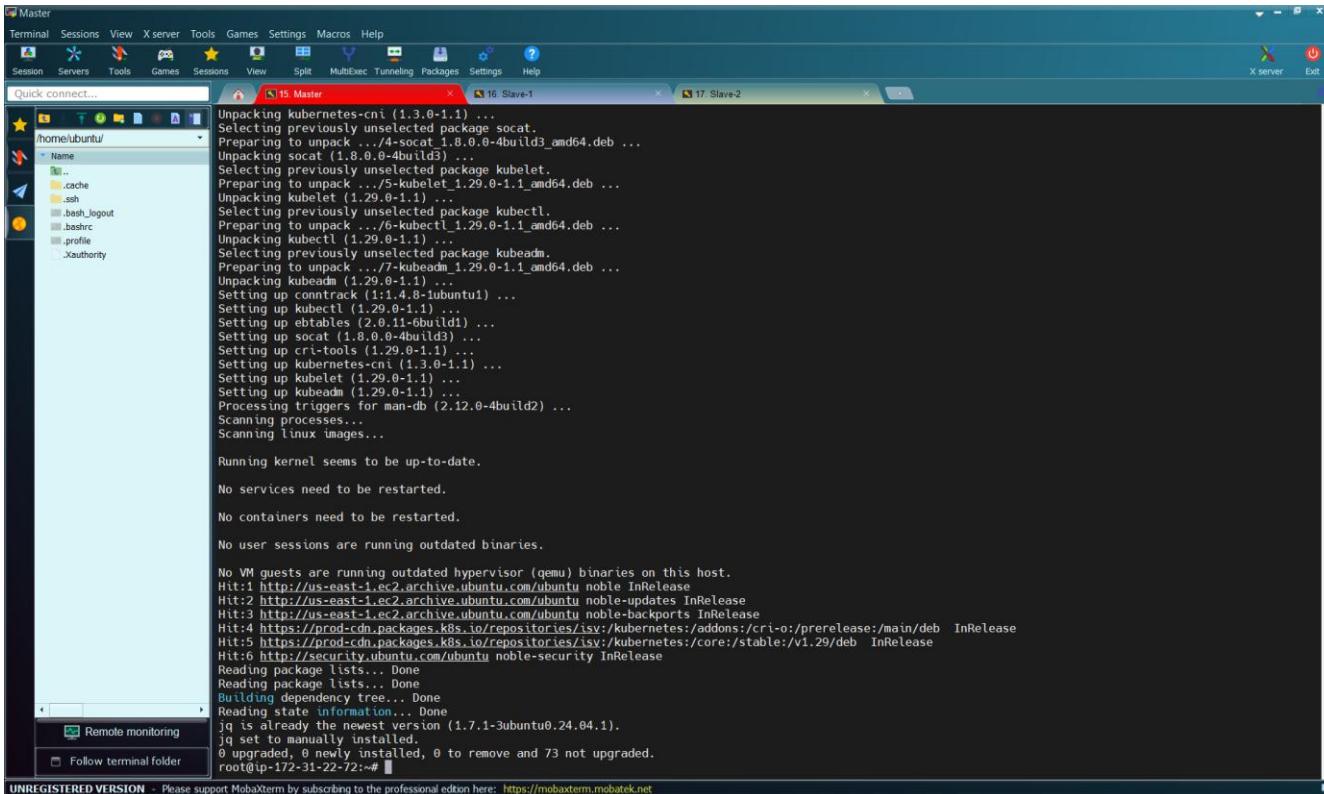
```
chmod +x install_kubernetes.sh
```



```
root@ip-172-31-22-72:~# vi install_kubernetes.sh
root@ip-172-31-22-72:~# chmod +x install_kubernetes.sh
root@ip-172-31-22-72:~#
```

Then execute the script using the command:

```
./install_kubernetes.sh
```



```
Unpacking kubernetes-cni (1.3.0-1.i) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-kubernetes-1.8.0-0-4build3_amd64.deb ...
Unpacking socat (1.8.0-0-4build3) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.29.0-1.i_amd64.deb ...
Unpacking kubelet (1.29.0-1.i) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.29.0-1.i_amd64.deb ...
Unpacking kubectl (1.29.0-1.i) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.29.0-1.i_amd64.deb ...
Unpacking kubeadm (1.29.0-1.i) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubelet (1.29.0-1.i) ...
Setting up ebtables (2.0.11-6build1) ...
Setting up socat (1.8.0-0-4build3) ...
Setting up cri-tools (1.29.0-1.i) ...
Setting up kubernetes-cni (1.3.0-1.i) ...
Setting up kubelet (1.29.0-1.i) ...
Setting up kubeadm (1.29.0-1.i) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

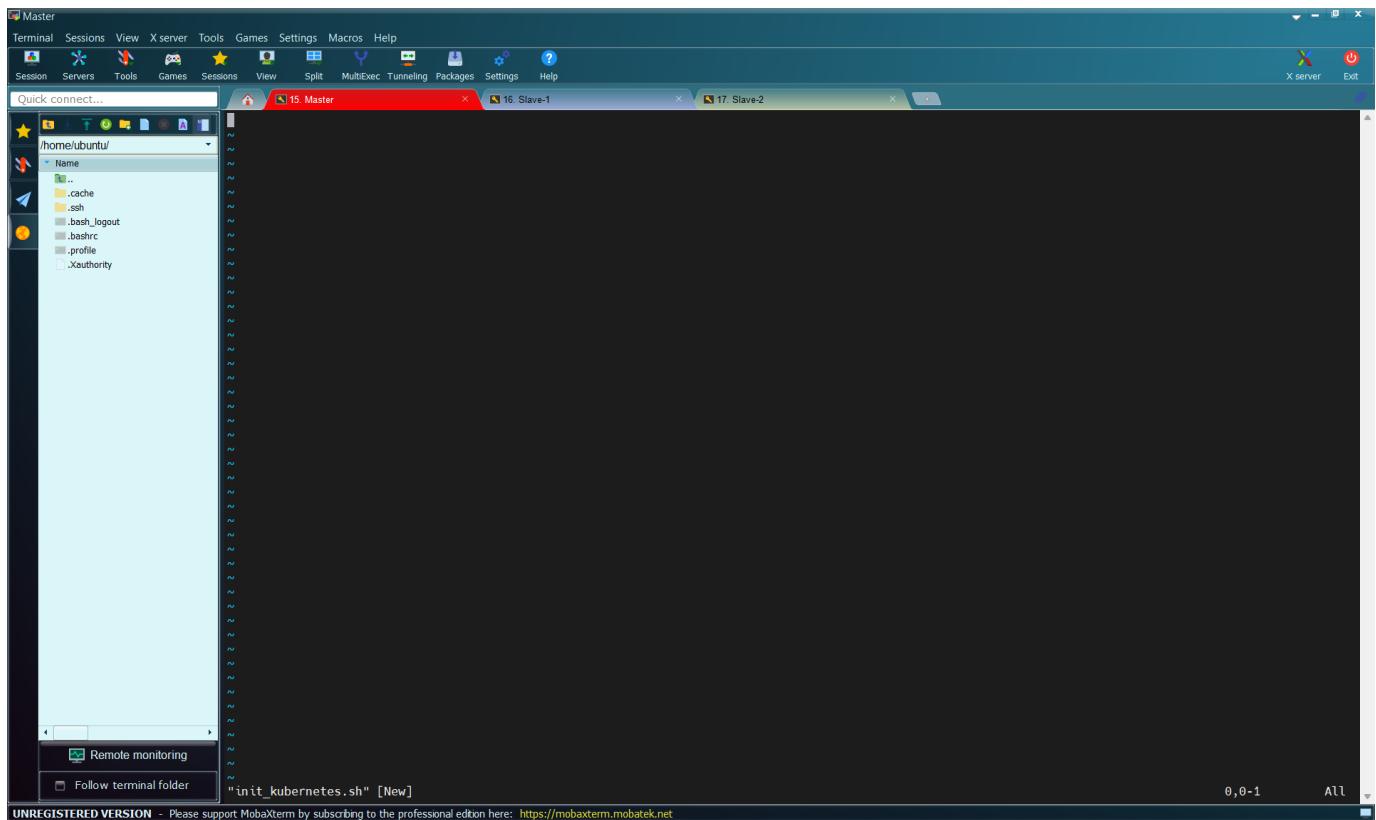
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://prod-cdn.packages.k8s.io/repositories/isy:/kubernetes-addons:cri-o:/prerelease:/main/deb InRelease
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isy:/kubernetes-core:/stable:/v1.29/deb InRelease
Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jq is already the newest version (1.7.1-3ubuntu0.24.04.1).
jq set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
root@ip-172-31-22-72:~#
```

Now, we are going to initialize Kubernetes Master Node. We will us shell scripting to do this too

Let us first create the script file called “**init\_kubernetes.sh**” using the command:

```
vi init_kubernetes.sh
```



Copy and paste the code below

```
#!/bin/bash

sudo kubeadm config images pull
sudo kubeadm init
mkdir -p "$HOME"/.kube
sudo cp -i /etc/kubernetes/admin.conf "$HOME"/.kube/config
sudo chown "$(id -u)":"$(id -g)" "$HOME"/.kube/config

# Network Plugin = calico
kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml
```

The screenshot shows a MobaXterm interface with three terminal sessions open:

- Session 15: Master (highlighted in red) contains a bash script to install Calico. The command `kubctl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml` is being run.
- Session 16: Slave-1
- Session 17: Slave-2

The left sidebar shows the file structure of the current directory: `/home/ubuntu/` with subfolders `.cache`, `.ssh`, `bash\_logout`, `bashrc`, `profile`, and `xauthority`. A context menu is open over the terminal window, showing options like "Remote monitoring" and "Follow terminal folder".

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

Save the file by pressing “**ESC**” followed by typing “**:wq**” and press “**Enter**”

The screenshot shows the same MobaXterm interface as before, but now terminal session 15 (Master) has the command `vi init\_kubernetes.sh` entered. The screen is mostly blank, indicating the file is being edited.

The left sidebar and bottom status bar are identical to the previous screenshot.

Give the file executable permission using the command:

```
chmod +x init_kubernetes.sh
```

```

root@ip-172-31-22-72:~# vi init_kubernetes.sh
root@ip-172-31-22-72:~# chmod +x init_kubernetes.sh
root@ip-172-31-22-72:~#

```

Then, we execute the command to run the file:

```
./init_kubernetes.sh
```

```

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:
export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:
kubeadm join 172.31.22.72:6443 --token 56rldd.3u1gtcj1ac5q22es \
--discovery-token-ca-cert-hash sha256:4d83248d2e26827628290e1e7d6ecc546dbdfc20da554943809570c3fd63d47e5
poddisruptionbudget.kube-controller-manager created
serviceaccount/calico-kube-controllers created
serviceaccount/calico-node created
serviceaccount/calico-cni-plugin created
configmap/calico-config created
customresourcedefinition.apirextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/bgpfilters.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/preservations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apirextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-cni-plugin created
daemonset.apps/calico-node created
deployment.apps/calico-kube-controllers created
root@ip-172-31-22-72:~#

```

The master node has been initialized and kube controllers have been created. Then check the status of the nodes using the command:

```
kubectl get nodes
```

```
root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-72   Ready   control-plane   78s   v1.29.0
```

After successfully running, your Kubernetes control plane will be initialized successfully. Let us now generate a token for worker nodes to join. To do this, run the command:

```
kubeadm token create --print-join-command
```

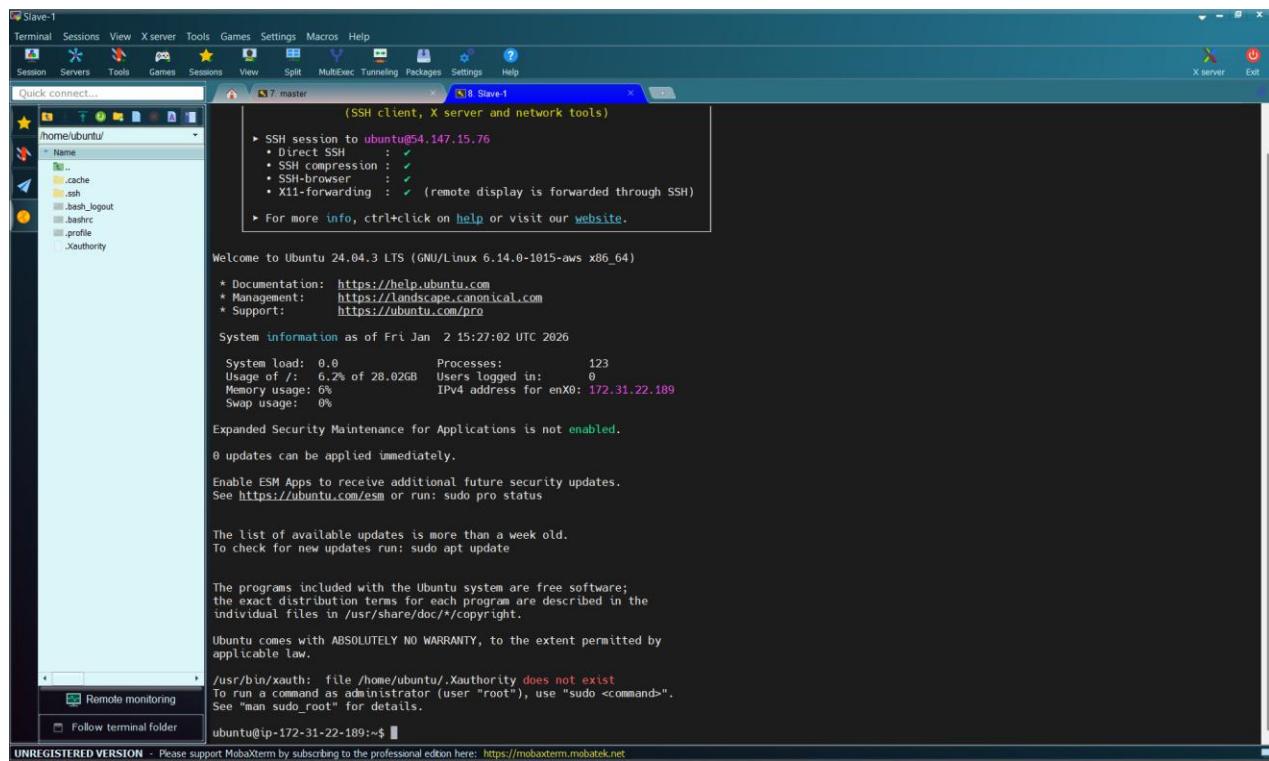
```
root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-72   Ready   control-plane   78s   v1.29.0
root@ip-172-31-22-72:~# kubeadm token create --print-join-command
kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd
63d47e5
root@ip-172-31-22-72:~# ^C
```

Copy these lines of code above, we are going to use it later

```
kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash
sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd63d47e5
```

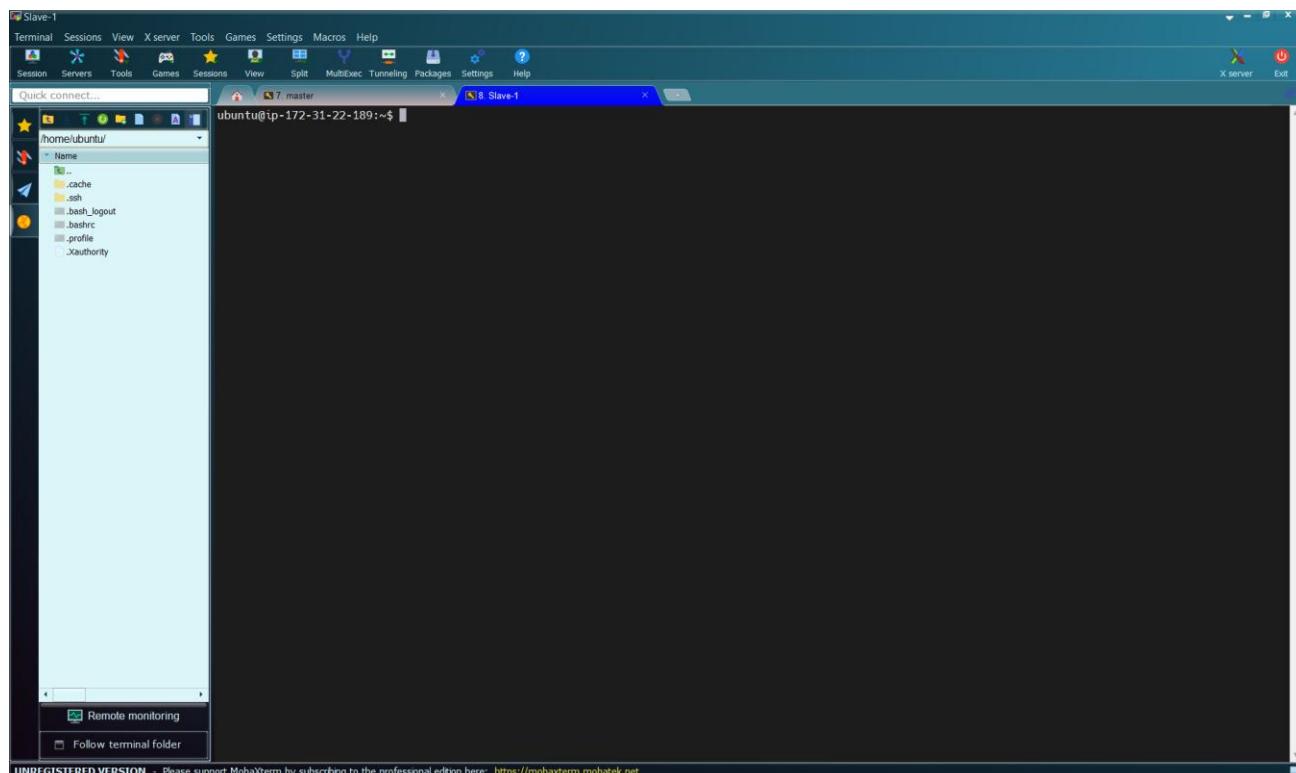
### 3.3.2 Set up Kubernetes on Virtual Machine “Slave-1”

We have to configure the virtual machine “Slave-1”. Open the session “Slave-1”



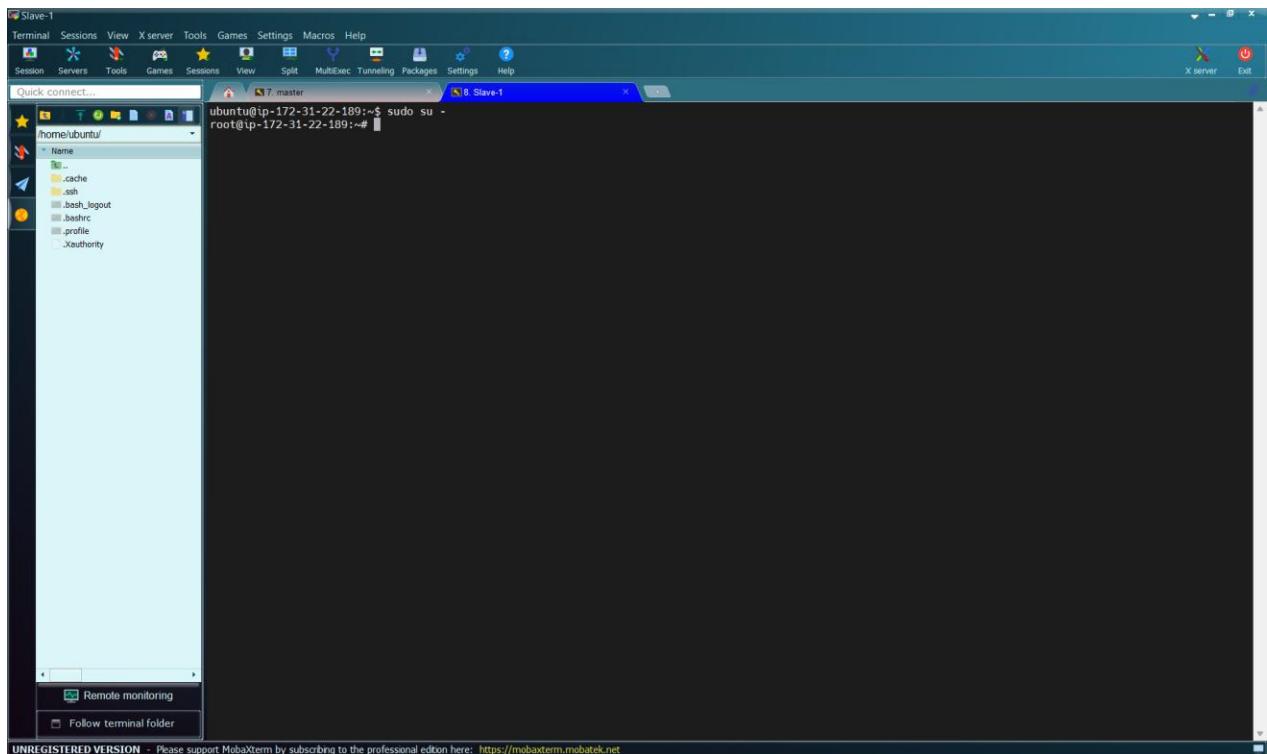
Clear the terminal using the command:

```
clear
```



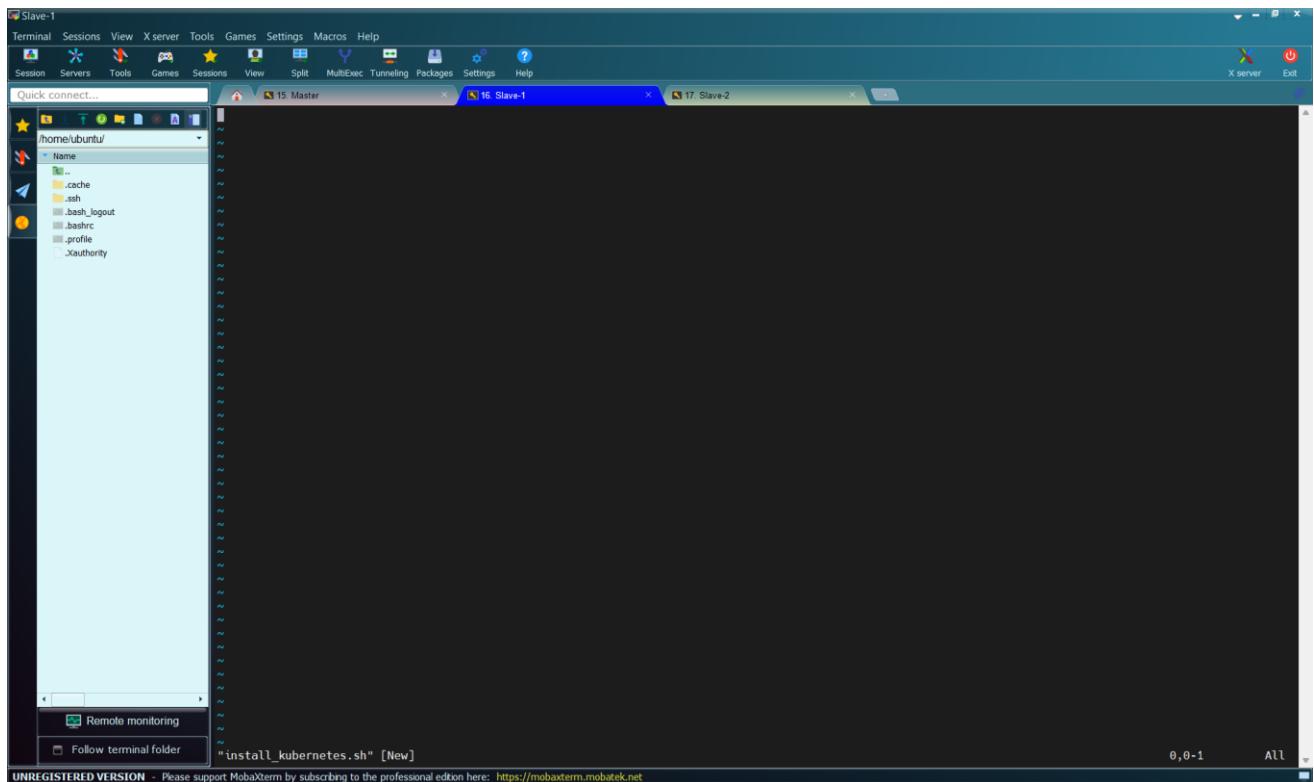
Then, let us give root user access using the command:

```
sudo su -
```



Now, let us install Kubernetes Components. We will do this by using a script. We will call the script "**install\_kubernetes.sh**". Let us create the file using the command:

```
vi install_kubernetes.sh
```



Paste these commands in the terminal

```
#!/bin/bash

curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"
echo "$(cat kubectl.sha256)  kubectl" | sha256sum --check
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
chmod +x kubectl
mkdir -p ~/.local/bin
mv ./kubectl ~/.local/bin/kubectl

# and then append (or prepend) ~/.local/bin to $PATH
kubectl version --client

# disable swap
sudo swapoff -a

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRI-O Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | 
sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg]
https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee
/etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o

sudo systemctl daemon-reload
sudo systemctl enable crio --now
sudo systemctl start crio.service

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages
```

```

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ ' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq

sudo systemctl enable --now kubelet
sudo systemctl start kubelet

```

```

Slave-1
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15. Master 16. Slave-1 17. Slave-2
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRI-O Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ " | sudo tee /etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o

sudo systemctl daemon-reload
sudo systemctl enable cri-o --now
sudo systemctl start cri-o.service

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ ' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq

sudo systemctl enable --now kubelet
sudo systemctl start kubelet

```

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

Save the file by pressing “**ESC**” followed by typing “**:wq**” and press “**Enter**”

```
root@ip-172-31-22-38:~# vi install_kubernetes.sh
```

Give the file executable permission using the command:

```
chmod +x install_kubernetes.sh
```

```
root@ip-172-31-22-38:~# vi install_kubernetes.sh
root@ip-172-31-22-38:~# chmod +x install_kubernetes.sh
root@ip-172-31-22-38:~#
```

Then execute the script using the command:

```
./install_kubernetes.sh
```

```

Slave-1
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 15 Master 16 Slave-1 17 Slave-2
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Slave-1
/home/ubuntu/
Name
..-
.cache
.ssh
.bash_logout
.bashrc
.profile
.xauthority
Unpacking kubernetes-cni (1.3.0-0.1) ...
Selecting previously unselected package socat.
Preparing to unpack .../4-socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.29.0-1.1_amd64.deb ...
Unpacking kubelet (1.29.0-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.29.0-1.1_amd64.deb ...
Unpacking kubectl (1.29.0-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.29.0-1.1_amd64.deb ...
Unpacking kubeadm (1.29.0-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubelet (1.29.0-1.1) ...
Setting up ebttables (2.0.11-6build1) ...
Setting up socat (1.8.0.0-4build3) ...
Setting up cri-tools (1.29.0-1.1) ...
Setting up kubernetes-cni (1.3.0-0.1) ...
Setting up kubelet (1.29.0-1.1) ...
Setting up kubeadm (1.29.0-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isy/:addons:/cri-o:/prerelease:/main/deb InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isy/:kubernetes:/core:/stable:/v1.29/deb InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jq is already the newest version (1.7.1-3ubuntu0.24.04.1).
jq set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
root@ip-172-31-22-38:~# 

```

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

We have Installed kubelet, kubeadm, and kubectl on the Slave-1 node.

Run the following commands on the worker node.

`sudo kubeadm reset pre-flight checks`

```

Slave-1
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect... 15 Master 16 Slave-1 17 Slave-2
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Slave-1
/home/ubuntu/
Name
..-
.cache
.ssh
.bash_logout
.bashrc
.profile
.xauthority
Preparing to unpack .../4-socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Selecting previously unselected package kubelet.
Preparing to unpack .../5-kubelet_1.29.0-1.1_amd64.deb ...
Unpacking kubelet (1.29.0-1.1) ...
Selecting previously unselected package kubectl.
Preparing to unpack .../6-kubectl_1.29.0-1.1_amd64.deb ...
Unpacking kubectl (1.29.0-1.1) ...
Selecting previously unselected package kubeadm.
Preparing to unpack .../7-kubeadm_1.29.0-1.1_amd64.deb ...
Unpacking kubeadm (1.29.0-1.1) ...
Setting up conntrack (1:1.4.8-1ubuntu1) ...
Setting up kubelet (1.29.0-1.1) ...
Setting up ebttables (2.0.11-6build1) ...
Setting up socat (1.8.0.0-4build3) ...
Setting up cri-tools (1.29.0-1.1) ...
Setting up kubernetes-cni (1.3.0-0.1) ...
Setting up kubelet (1.29.0-1.1) ...
Setting up kubeadm (1.29.0-1.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isy/:addons:/cri-o:/prerelease:/main/deb InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isy/:kubernetes:/core:/stable:/v1.29/deb InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jq is already the newest version (1.7.1-3ubuntu0.24.04.1).
jq set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
root@ip-172-31-22-38:~# sudo kubeadm reset pre-flight checks
W0111 16:00:01.106358 3844 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y

```

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

Type “y” and press “Enter”

```

Slave-1
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15 Master x 16 Slave-1 x 17 Slave-2 x
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isy/kubernetes:/addons:cri-o:/prerelease:/main/deb InRelease
Hit:6 https://prod-cdn.packages.k8s.io/repositories/isy/kubernetes:/core:/stable:/v1.29/deb InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jq is already the newest version (1.7.1-3ubuntu0.24.04.1).
jq set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
root@ip-172-31-22-38:~# sudo kubeadm reset pre-flight checks
W0111 16:00:01.106358 3844 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Pre-flight checks
W0111 16:00:33.331742 1344 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d
The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.
If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

root@ip-172-31-22-38:~#

```

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

Paste the join command you got from the master node and append `--v=5` at the end. Make sure either you are working as sudo user or use sudo before the command.

To do this, go to the “master” terminal on MobaXterm

```

Master
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15 Master x 16 Slave-1 x 17 Slave-2 x
root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS    ROLES   AGE     VERSION
ip-172-31-22-72 Ready    control-plane   78s   v1.29.0
root@ip-172-31-22-72:~# kubeadm token create --print-join-command
kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd
13d47e5
root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# ^C

root@ip-172-31-22-72:~#

```

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

Copy the above line of code:

```
kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd63d47e5
```

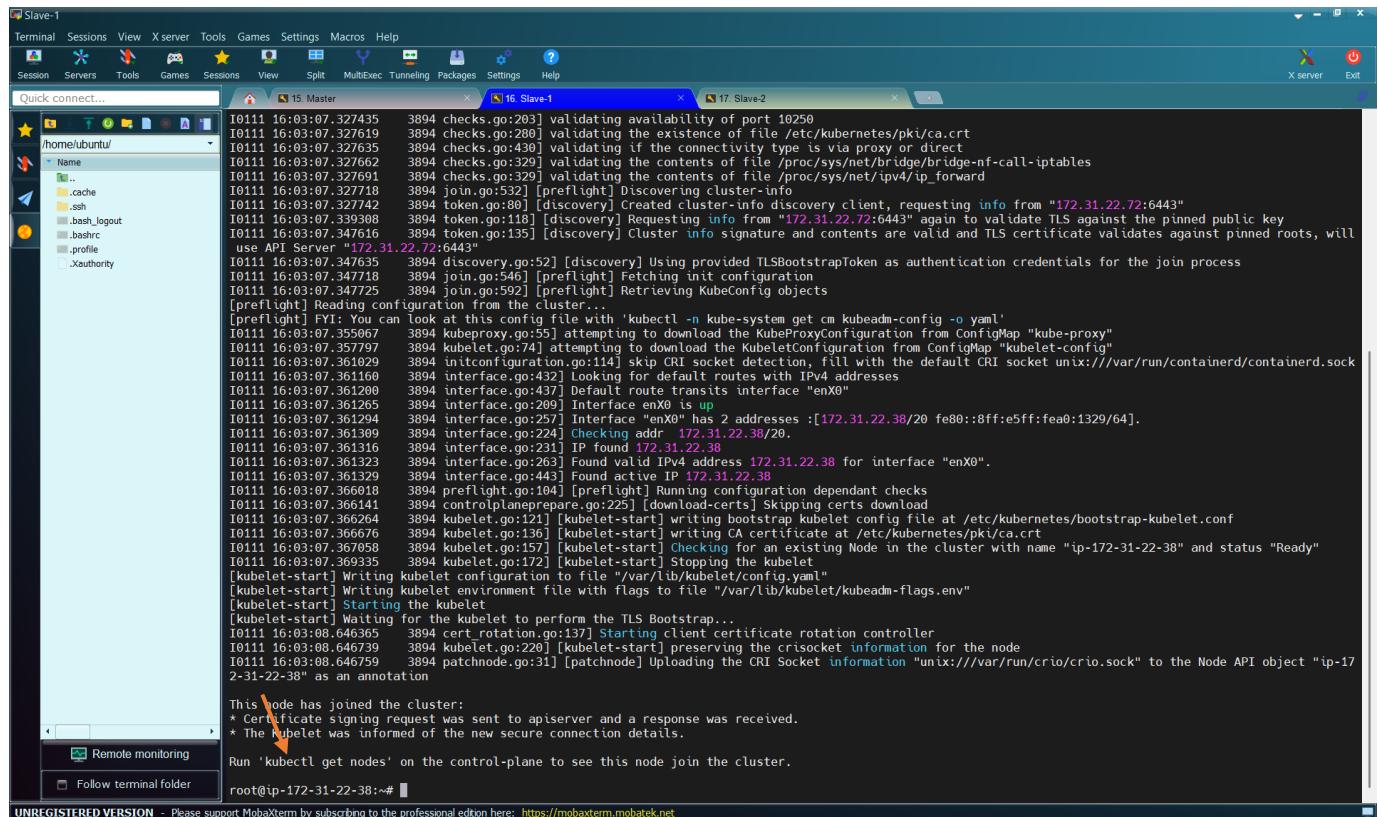
Add “**sudo**” inform of the line of code

```
sudo kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd63d47e5
```

Add “**--v=5**” at the end of the code

```
sudo kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd63d47e5 --v=5
```

And run this on the “**Slave-1**” terminal. This command is asking the worker node to be part of Master node.



```
I0111 16:03:07.327435 3894 checks.go:202] validating availability of port 10259
I0111 16:03:07.327619 3894 checks.go:280] validating the existence of file /etc/kubernetes/pki/ca.crt
I0111 16:03:07.327635 3894 checks.go:438] validating if the connectivity type is via proxy or direct
I0111 16:03:07.327662 3894 checks.go:329] validating the contents of file /proc/sys/net/bridge/bridge-nf-call-iptables
I0111 16:03:07.327691 3894 checks.go:329] validating the contents of file /proc/sys/net/ipv4/ip_forward
I0111 16:03:07.327718 3894 join.go:532] [preflight] Discovering cluster-info
I0111 16:03:07.327742 3894 token.go:86] [discovery] Created cluster-info discovery client, requesting info from "172.31.22.72:6443"
I0111 16:03:07.339308 3894 token.go:118] [discovery] Requesting info from "172.31.22.72:6443" again to validate TLS against the pinned public key
I0111 16:03:07.347616 3894 token.go:135] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will use API Server "172.31.22.72:6443"
I0111 16:03:07.347635 3894 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0111 16:03:07.347718 3894 join.go:546] [preflight] Fetching init configuration
I0111 16:03:07.347725 3894 join.go:592] [preflight] Retrieving KubeConfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with `kubectl -n kube-system get cm kubeadm-config -o yaml`
I0111 16:03:07.350567 3894 kubeProxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0111 16:03:07.357797 3894 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
I0111 16:03:07.361029 3894 unitconfig.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
I0111 16:03:07.361160 3894 interface.go:432] Looking for default routes with IPv4 addresses
I0111 16:03:07.361200 3894 interface.go:437] Default route transits interface "enX0"
I0111 16:03:07.361265 3894 interface.go:289] Interface enX0 is up
I0111 16:03:07.361294 3894 interface.go:257] Interface "enX0" has 2 addresses :[172.31.22.38/20 fe80::8ff:e5ff:fea0:1329/64].
I0111 16:03:07.361309 3894 interface.go:224] Checking addr 172.31.22.38/20.
I0111 16:03:07.361316 3894 interface.go:231] IP found 172.31.22.38
I0111 16:03:07.361323 3894 interface.go:263] Found valid IPv4 address 172.31.22.38 for interface "enX0".
I0111 16:03:07.361329 3894 interface.go:443] Found active IP 172.31.22.38
I0111 16:03:07.366018 3894 preflight.go:104] [preflight] Running configuration dependant checks
I0111 16:03:07.366141 3894 controlPlanePrepare.go:225] [download-certs] Skipping certs download
I0111 16:03:07.366264 3894 kubelet.go:121] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
I0111 16:03:07.366676 3894 kubelet.go:136] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
I0111 16:03:07.367058 3894 kubelet.go:157] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-22-38" and status "Ready"
I0111 16:03:07.369335 3894 kubelet.go:172] [kubelet-start] Stopping the kubelet
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...
I0111 16:03:08.646365 3894 cert_rotation.go:137] Starting client certificate rotation controller
I0111 16:03:08.646739 3894 kubelet.go:220] [kubelet-start] preserving the crisocket information for the node
I0111 16:03:08.646759 3894 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/crio/crio.sock" to the Node API object "ip-172-31-22-38" as an annotation
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The kubelet was informed of the new secure connection details.

Run 'kubctl get nodes' on the control-plane to see this node join the cluster.
root@ip-172-31-22-38:~#
```

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

Then, verify if it is working as expected by running this command on the “**Master**” node:

```
kubectl get nodes
```

```

root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-72   Ready    control-plane   78s   v1.29.0
root@ip-172-31-22-72:~# kubeadm token create --print-join-command
kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd
634bf353

root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-38   Ready    <none>    50s   v1.29.0
ip-172-31-22-72   Ready    control-plane   15m   v1.29.0
root@ip-172-31-22-72:~# 

```

You can see that we have the “**Master**” node, that is “**control-plane**” and the “**Slave-1**” node, that is “**none**” ready.

### 3.3.3 Set up Kubernetes on Virtual Machine “Slave-2”

We have to configure the virtual machine “**Slave-2**”. Open the session “**Slave-2**”

```

Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Fri Jan 2 16:06:27 UTC 2026
System load: 0.0          Processes:        121
Usage of /: 6.2% of 28.02GB Users logged in: 0
Memory usage: 6%           IPv4 address for enX0: 172.31.31.154
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright*.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

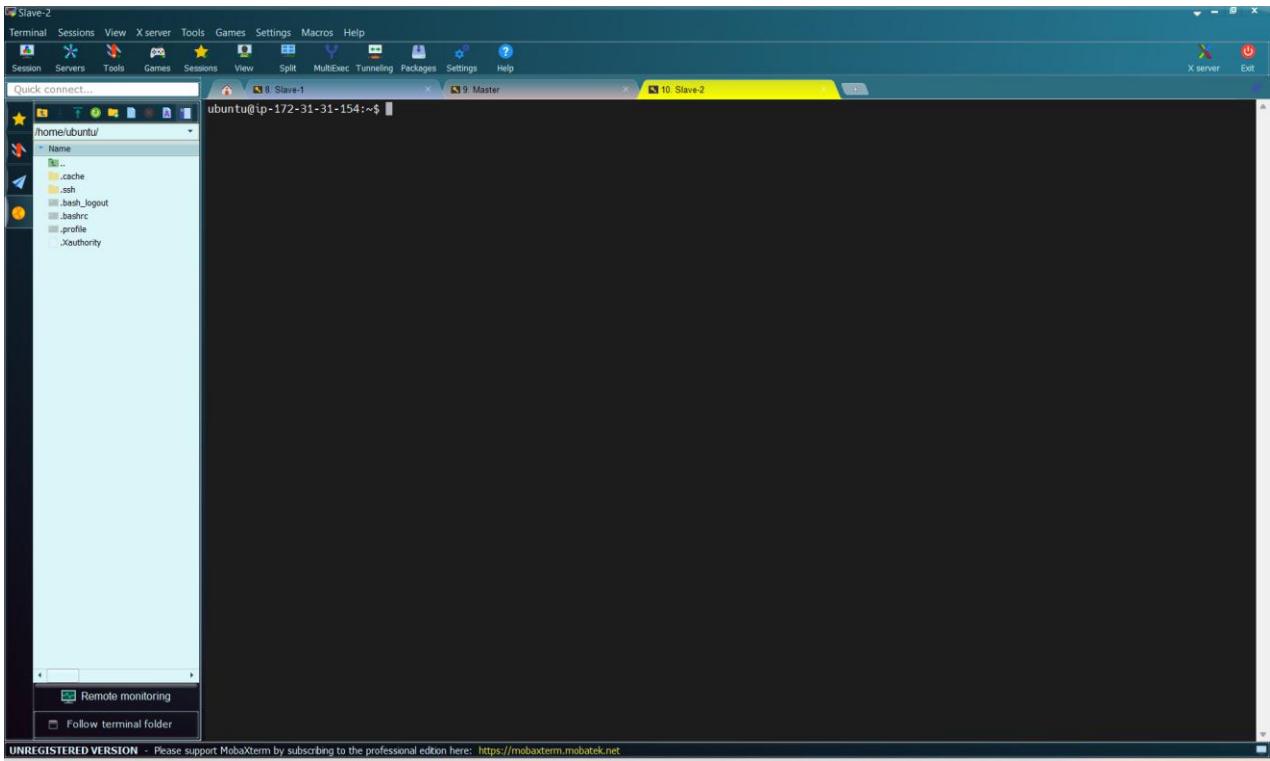
/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-31-154:~$ 

```

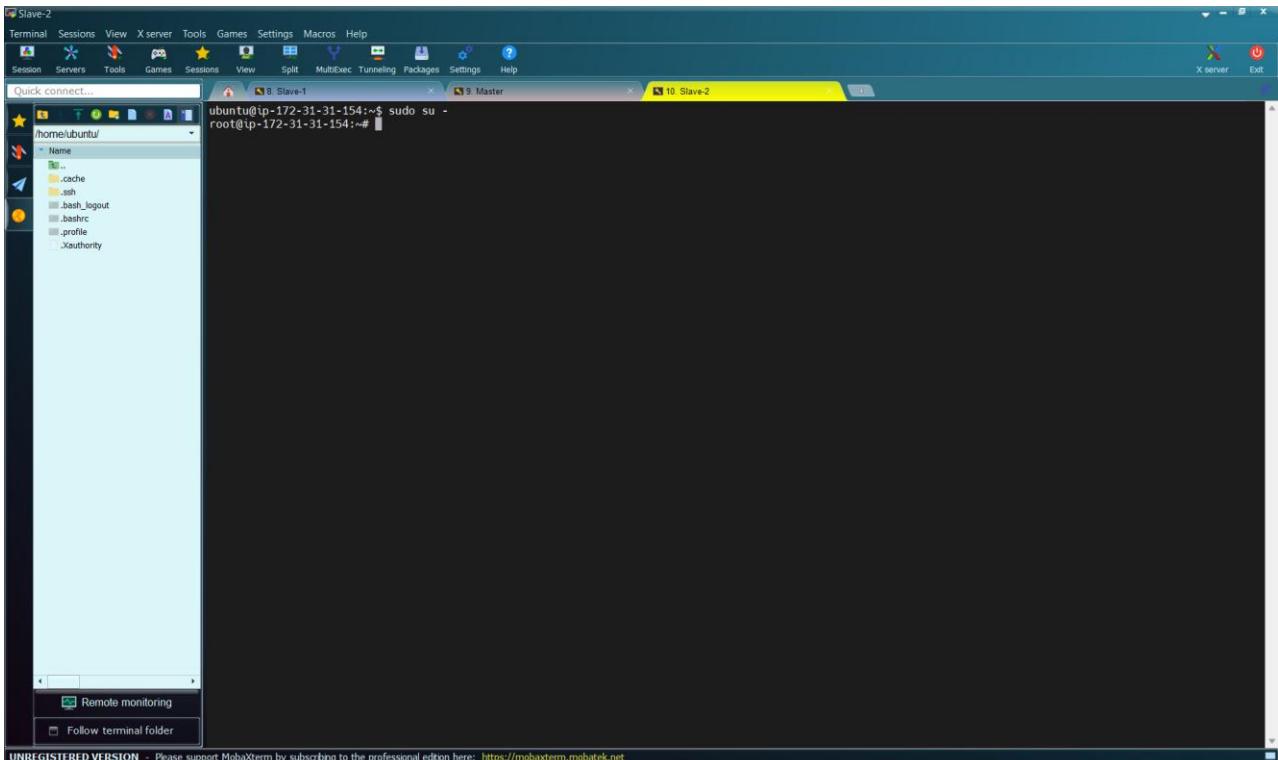
Clear the terminal using the command:

`clear`



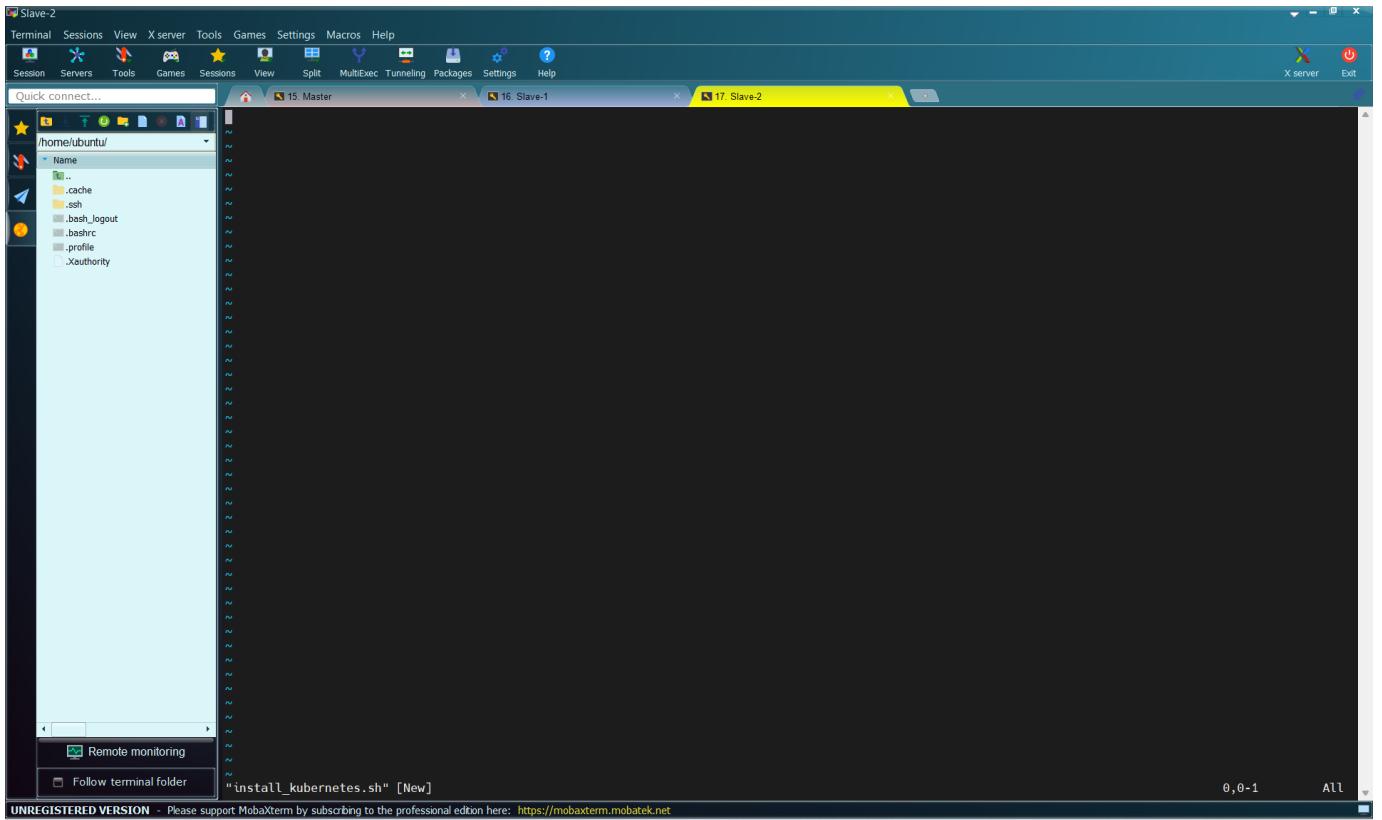
Then, let us give root user access using the command:

```
sudo su -
```



Now, let us install Kubernetes Components. We will do this by using a script. We will call the script "**installing-kubernetes.sh**". Let us create the file using the command:

```
vi install_kubernetes.sh
```



Paste these commands in the terminal

```
#!/bin/bash

curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"
echo "$(cat kubectl.sha256)  kubectl" | sha256sum --check
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
chmod +x kubectl
mkdir -p ~/.local/bin
mv ./kubectl ~/.local/bin/kubectl

# and then append (or prepend) ~/.local/bin to $PATH
kubectl version --client

# disable swap
sudo swapoff -a

# Create the .conf file to load the modules at bootup
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
EOF
```

```

net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRI-O Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee /etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o

sudo systemctl daemon-reload
sudo systemctl enable crio --now
sudo systemctl start crio.service

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq

sudo systemctl enable --now kubelet
sudo systemctl start kubelet

```

```

Slave-2
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15 Master x 16 Slave-1 x 17 Slave-2 x
overlay
br_nfnetfilter
EOF

sudo modprobe overlay
sudo modprobe br_nfnetfilter

# sysctl params required by setup, params persist across reboots
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ipTables = 1
net.ipv4.ip_forward = 1
EOF

# Apply sysctl params without reboot
sudo sysctl --system

## Install CRI-O Runtime
sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-certificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee /etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o

sudo systemctl daemon-reload
sudo systemctl enable crio --now
sudo systemctl start crio.service
echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /" | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
sudo apt-get update -y
sudo apt-get install -y jq

sudo systemctl enable --now kubelet
sudo systemctl start kubelet

```

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

Save the file by pressing “**ESC**” followed by typing “**:wq**” and press “**Enter**”

```

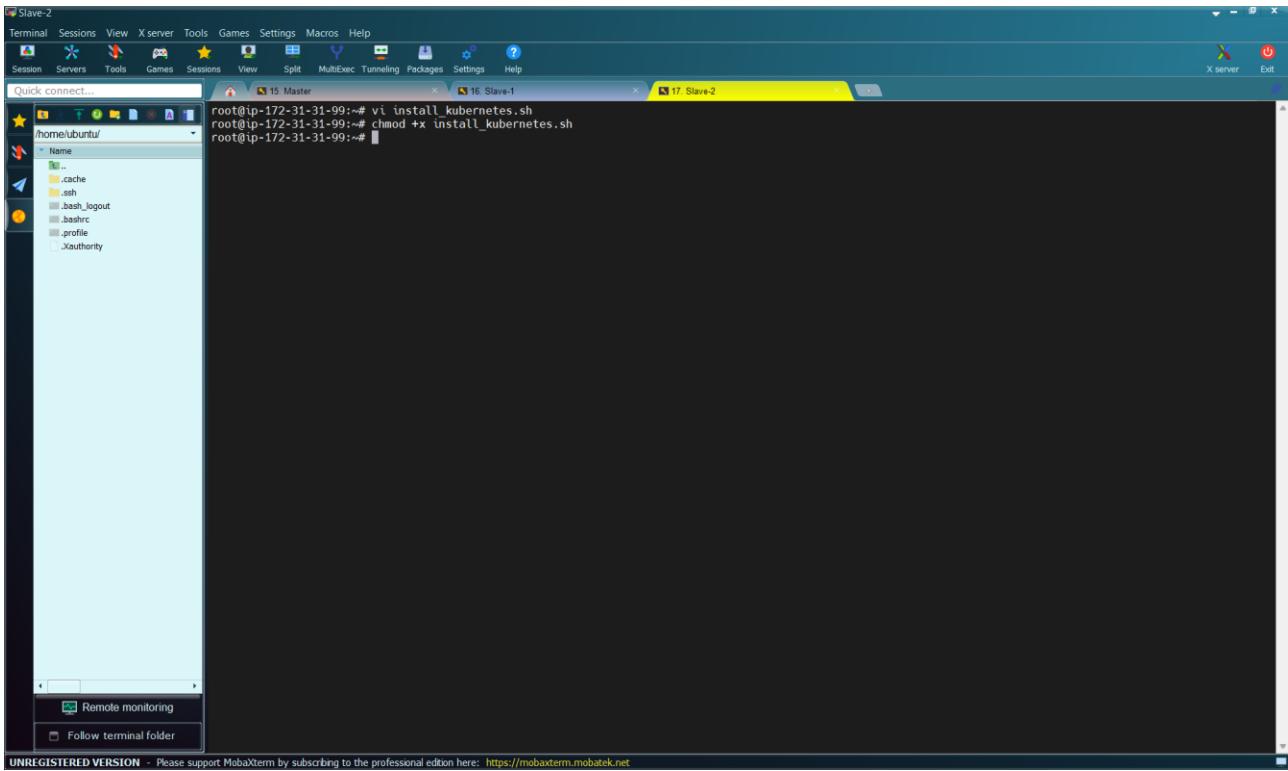
Slave-2
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15 Master x 16 Slave-1 x 17 Slave-2 x
root@ip-172-31-31-99:~# vi install_kubernetes.sh
root@ip-172-31-31-99:~#

```

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

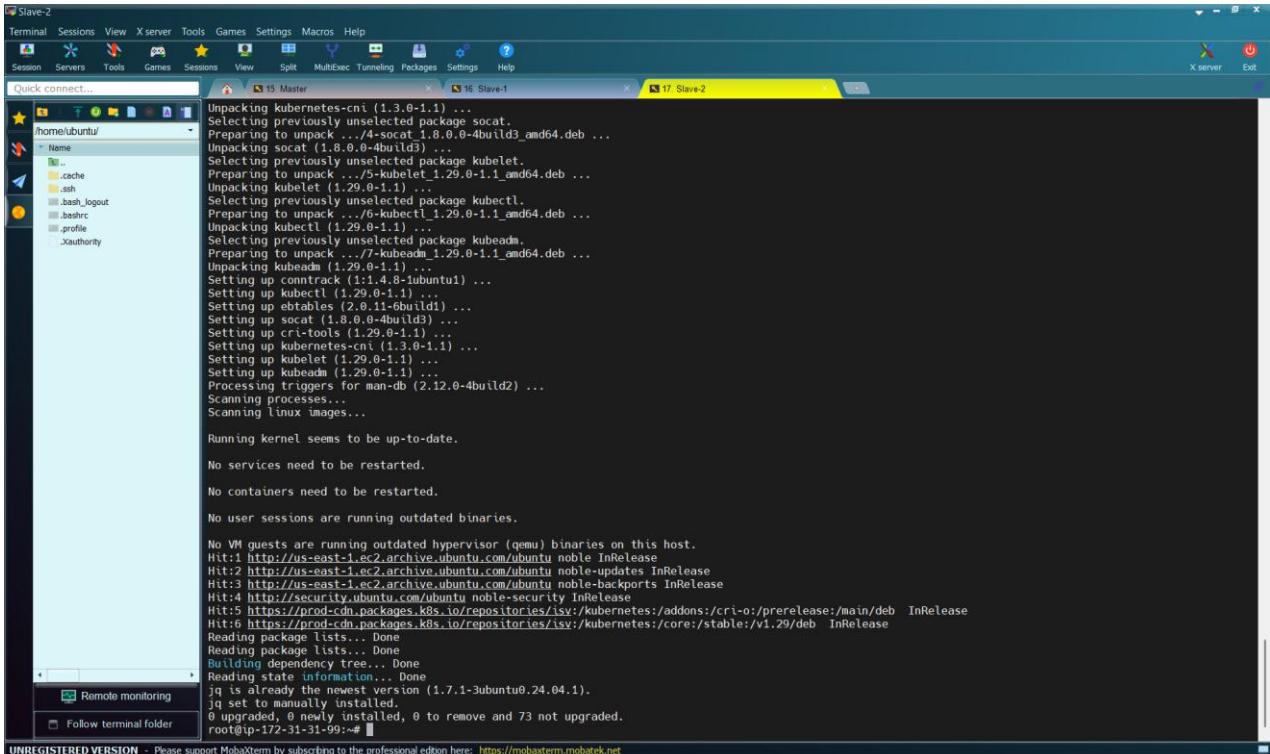
Give the file executable permission using the command:

```
chmod +x install_kubernetes.sh
```



Then execute the script using the command:

```
./install_kubernetes.sh
```



We have Installed kubelet, kubeadm, and kubectl on the Slave-2 node. Run the following commands on Slave-2 node.

```
sudo kubeadm reset pre-flight checks
```

```
root@ip-172-31-31-99:~# sudo kubeadm reset pre-flight checks
W0111 16:10:22.208817 3821 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
```

Type “y” and press “Enter”

```
root@ip-172-31-31-99:~# sudo kubeadm reset pre-flight checks
W0111 16:10:22.208817 3821 preflight.go:56] [reset] WARNING: Changes made to this host by 'kubeadm init' or 'kubeadm join' will be reverted.
[reset] Are you sure you want to proceed? [y/N]: y
[preflight] Running pre-flight checks
W0111 16:10:52.668318 3821 removeetcdmember.go:106] [reset] No kubeadm config, using etcd pod spec to get data directory
[reset] Deleted contents of the etcd data directory: /var/lib/etcd
[reset] Stopping the kubelet service
[reset] Unmounting mounted directories in "/var/lib/kubelet"
[reset] Deleting contents of directories: [/etc/kubernetes/manifests /var/lib/kubelet /etc/kubernetes/pki]
[reset] Deleting files: [/etc/kubernetes/admin.conf /etc/kubernetes/super-admin.conf /etc/kubernetes/kubelet.conf /etc/kubernetes/bootstrap-kubelet.conf /etc/kubernetes/controller-manager.conf /etc/kubernetes/scheduler.conf]

The reset process does not clean CNI configuration. To do so, you must remove /etc/cni/net.d

The reset process does not reset or clean up iptables rules or IPVS tables.
If you wish to reset iptables, you must do so manually by using the "iptables" command.

If your cluster was setup to utilize IPVS, run ipvsadm --clear (or similar)
to reset your system's IPVS tables.

The reset process does not clean your kubeconfig files and you must remove them manually.
Please, check the contents of the ${HOME}/.kube/config file.
root@ip-172-31-31-99:~#
```

Paste the join command you got from the master node and append --v=5 at the end. Make sure either you are working as sudo user or use sudo before the command.

Copy the above line of code:

```
kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd63d47e5
```

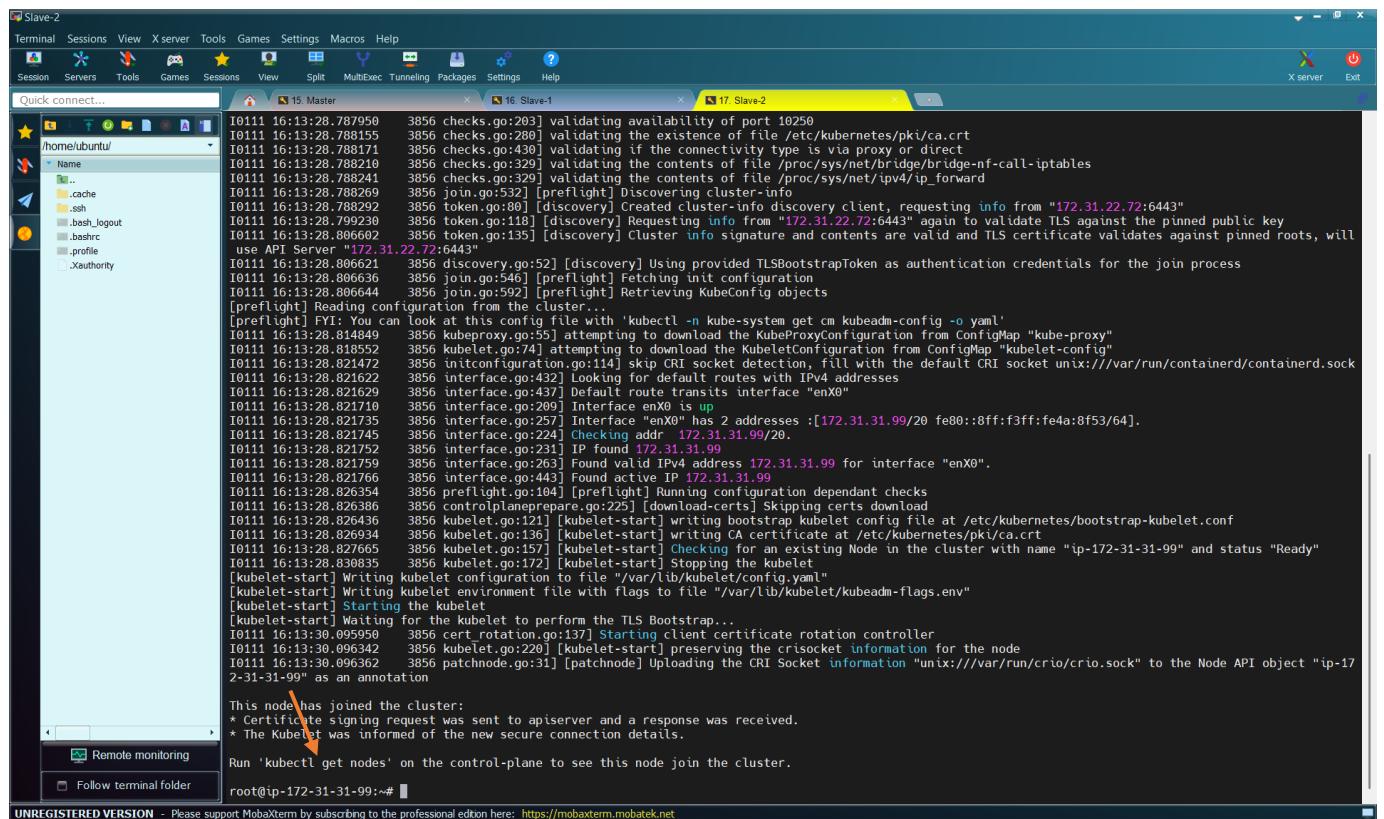
## Add “sudo” inform of the line of code

```
sudo kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd63d47e5
```

## Add “--v=5” at the end of the code

```
sudo kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd63d47e5 --v=5
```

And run this on the “Slave-2” terminal. This command is asking the worker node to be part of Master node.



```
I0111 16:13:28.787950 3856 checks.go:203] validating availability of port 10250
I0111 16:13:28.788155 3856 checks.go:280] validating the existence of file /etc/kubernetes/pki/ca.crt
I0111 16:13:28.788171 3856 checks.go:430] validating if the connectivity type is via proxy or direct
I0111 16:13:28.788210 3856 checks.go:329] validating the contents of file /proc/sys/net/bridge/bridge-nf-call-iptables
I0111 16:13:28.788241 3856 checks.go:329] validating the contents of file /proc/sys/net/ipv4/ip_forward
I0111 16:13:28.788269 3856 join.go:532] [preflight] Discovering cluster-info
I0111 16:13:28.788292 3856 token.go:80] [discovery] Created cluster-info discovery client, requesting info from "172.31.22.72:6443"
I0111 16:13:28.799230 3856 token.go:118] [discovery] Requesting info from "172.31.22.72:6443" again to validate TLS against the pinned public key
I0111 16:13:28.806602 3856 token.go:135] [discovery] Cluster info signature and contents are valid and TLS certificate validates against pinned roots, will use API Server "172.31.22.72:6443"
I0111 16:13:28.806621 3856 discovery.go:52] [discovery] Using provided TLSBootstrapToken as authentication credentials for the join process
I0111 16:13:28.806636 3856 join.go:546] [preflight] Fetching init configuration
I0111 16:13:28.806644 3856 join.go:592] [preflight] Retrieving KubeConfig objects
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
I0111 16:13:28.814849 3856 kubeProxy.go:55] attempting to download the KubeProxyConfiguration from ConfigMap "kube-proxy"
I0111 16:13:28.818552 3856 kubelet.go:74] attempting to download the KubeletConfiguration from ConfigMap "kubelet-config"
I0111 16:13:28.821472 3856 initConfiguration.go:114] skip CRI socket detection, fill with the default CRI socket unix:///var/run/containerd/containerd.sock
I0111 16:13:28.821622 3856 interface.go:432] Looking for default routes with IPv4 addresses
I0111 16:13:28.821629 3856 interface.go:437] Default route transits interface "enX0"
I0111 16:13:28.821710 3856 interface.go:299] Interface enX0 is up
I0111 16:13:28.821735 3856 interface.go:257] Interface "enX0" has 2 addresses :[172.31.31.99/20 fe80::8ff:f3ff:fe4a:8f53/64].
I0111 16:13:28.821745 3856 interface.go:224] Checking addr 172.31.31.99/20.
I0111 16:13:28.821752 3856 interface.go:231] IP found 172.31.31.99
I0111 16:13:28.821759 3856 interface.go:263] Found valid IPv4 address 172.31.31.99 for interface "enX0".
I0111 16:13:28.821766 3856 interface.go:443] Found active IP 172.31.31.99
I0111 16:13:28.826354 3856 preflight.go:184] [preflight] Running configuration dependant checks
I0111 16:13:28.826386 3856 controlPlanePrepare.go:225] [download-certs] Skipping certs download
I0111 16:13:28.826436 3856 kubelet.go:121] [kubelet-start] writing bootstrap kubelet config file at /etc/kubernetes/bootstrap-kubelet.conf
I0111 16:13:28.826934 3856 kubelet.go:136] [kubelet-start] writing CA certificate at /etc/kubernetes/pki/ca.crt
I0111 16:13:28.827665 3856 kubelet.go:157] [kubelet-start] Checking for an existing Node in the cluster with name "ip-172-31-31-99" and status "Ready"
I0111 16:13:28.830835 3856 kubelet.go:172] [kubelet-start] Stopping the kubelet
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...
I0111 16:13:30.095950 3856 cert_rotation.go:137] Starting client certificate rotation controller
I0111 16:13:30.096342 3856 kubelet.go:220] [kubelet-start] preserving the crisocket information for the node
I0111 16:13:30.096362 3856 patchnode.go:31] [patchnode] Uploading the CRI Socket information "unix:///var/run/crio/crio.sock" to the Node API object "ip-172-31-31-99" as an annotation
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@ip-172-31-31-99:~#
```

Then, verify if it is working as expected by running this command on the “Master” node:

```
kubectl get nodes
```

```

root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-72 Ready    control-plane   78s  v1.29.0
root@ip-172-31-22-72:~# kubeadm token create --print-join-command
kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943889570c3fd
63d47e5
root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-38 Ready    <none>    50s  v1.29.0
ip-172-31-22-72 Ready    control-plane  15m  v1.29.0
root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-38 Ready    <none>    11m  v1.29.0
ip-172-31-22-72 Ready    control-plane  26m  v1.29.0
ip-172-31-31-99 Ready    <none>    56s  v1.29.0
root@ip-172-31-22-72:~#

```

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

You can see that we have the “**Master**” node, that is “**control-plane**”, the “**Slave-1**” node, that is “**none**” ready and “**Slave-2**” with another “**none**” that is also ready.

### 3.3.4 Deploy Sample Nginx Microservice on Kubernetes

Let us create a deployment on master node named “**nginx-deploy**” using YAML. To do this we will create the file called “**nginx-deploy.yaml**” by using the command on “**Master**”:

`vi nginx-deploy.yaml`

```

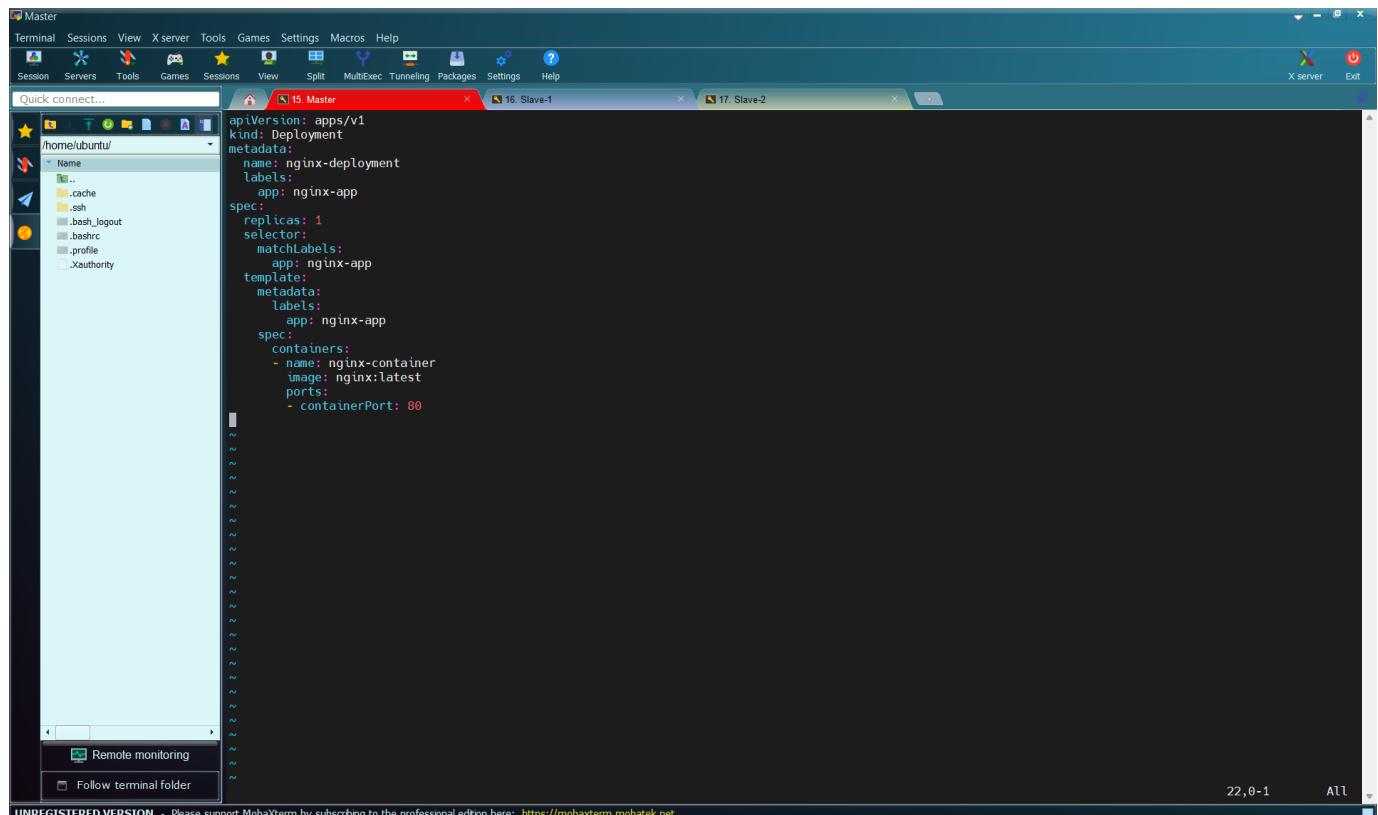
root@ip-172-31-22-72:~# vi nginx-deploy.yaml [New]

```

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

Paste the code below

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-app
  template:
    metadata:
      labels:
        app: nginx-app
    spec:
      containers:
        - name: nginx-container
          image: nginx:latest
          ports:
            - containerPort: 80
```



Save and exit by pressing “**ESC**”, followed by “**:wq**” and press “**Enter**”

```

root@ip-172-31-22-72:~# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-22-72   Ready   control-plane   78s   v1.29.0
root@ip-172-31-22-72:~# kubeadm token create --print-join-command
kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943889570c3fd
63d47e5
root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-22-38   Ready   <none>   59s   v1.29.0
ip-172-31-22-72   Ready   control-plane   15m   v1.29.0
root@ip-172-31-22-72:~# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-22-38   Ready   <none>   11m   v1.29.0
ip-172-31-22-72   Ready   control-plane   26m   v1.29.0
ip-172-31-31-99   Ready   <none>   56s   v1.29.0
root@ip-172-31-22-72:~# vi nginx-deploy.yaml
root@ip-172-31-22-72:~#

```

Then, run the command to create the deployment:

```
kubectl apply -f nginx-deploy.yaml
```

```

root@ip-172-31-22-72:~# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-22-72   Ready   control-plane   78s   v1.29.0
root@ip-172-31-22-72:~# kubeadm token create --print-join-command
kubeadm join 172.31.22.72:6443 --token gyp4eo.9z19c8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943889570c3fd
63d47e5
root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-22-38   Ready   <none>   59s   v1.29.0
ip-172-31-22-72   Ready   control-plane   15m   v1.29.0
root@ip-172-31-22-72:~# kubectl get nodes
NAME      STATUS   ROLES    AGE     VERSION
ip-172-31-22-38   Ready   <none>   11m   v1.29.0
ip-172-31-22-72   Ready   control-plane   26m   v1.29.0
ip-172-31-31-99   Ready   <none>   56s   v1.29.0
root@ip-172-31-22-72:~# vi nginx-deploy.yaml
root@ip-172-31-22-72:~# kubectl apply -f nginx-deploy.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-22-72:~#

```

The deployment called “**nginx-deployment**” has been created. Let us check the deployment using the command:

```
kubectl get deployments
```

```

root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-72   Ready    control-plane   78s   v1.29.0
root@ip-172-31-22-72:~# kubeadm token create --print-join-command
kubeadm join 172.31.22.72:6443 --token gyp4eo.9zlc8n15fz5opbx --discovery-token-ca-cert-hash sha256:4d03248d2e2682762820e1e7d6ecc546dbdfc20da554943809570c3fd63d47e5
root@ip-172-31-22-72:~# ^C
root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-38   Ready    <none>    50s   v1.29.0
ip-172-31-22-72   Ready    control-plane   15m   v1.29.0
root@ip-172-31-22-72:~# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-22-38   Ready    <none>    11m   v1.29.0
ip-172-31-22-72   Ready    control-plane   26m   v1.29.0
ip-172-31-31-99   Ready    <none>    56s   v1.29.0
root@ip-172-31-22-72:~# vi nginx-deploy.yaml
root@ip-172-31-22-72:~# kubectl apply -f nginx-deploy.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-22-72:~# kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/1       1           1          41s
root@ip-172-31-22-72:~# 

```

You can see the deployment we just created. We have defined two replicas and zero of them are running. Let us check the pods again using the command:

```
kubectl get pods
```

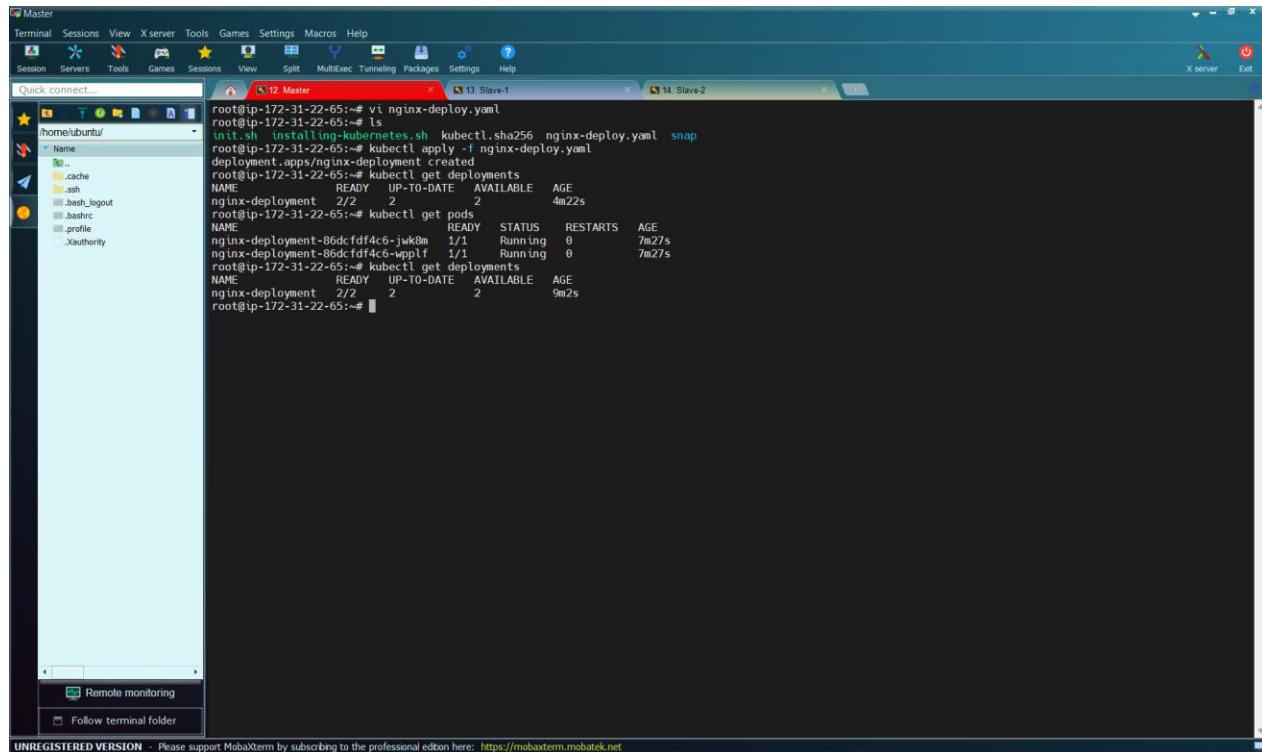
```

root@ip-172-31-22-65:~# vi nginx-deploy.yaml
root@ip-172-31-22-65:~# ls
init.sh  installing-kubernetes.sh  kubelet.sha256  nginx-deploy.yaml  snap
root@ip-172-31-22-65:~# kubectl apply -f nginx-deploy.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-22-65:~# kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2       2           2          4m22s
root@ip-172-31-22-65:~# kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
nginx-deployment-86dcfdf4c6-jwk8m  1/1     Running   0          7m27s
nginx-deployment-86dcfdf4c6-wpplf  1/1     Running   0          7m27s
root@ip-172-31-22-65:~# 

```

You can see that two pods are running. Let us check the deployments again using the command

```
kubectl get deployments
```



The screenshot shows a terminal window titled 'Master' in a 'Mobaxterm' interface. The terminal displays the following command execution:

```
root@ip-172-31-22-65:~# vi nginx-deploy.yaml
root@ip-172-31-22-65:~# ls
init.sh  installing-kubernetes.sh  kubelet.sha256  nginx-deploy.yaml  snap
deployment.apps/nginx-deployment created
root@ip-172-31-22-65:~# kubectl apply -f nginx-deploy.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-22-65:~# kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2           2          4m22s
root@ip-172-31-22-65:~# kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx-deployment-86dcfdf4c6-jvk8m  1/1     Running   0          7m27s
nginx-deployment-86dcfdf4c6-vpp1f  1/1     Running   0          7m27s
root@ip-172-31-22-65:~# kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2           2          9m2s
root@ip-172-31-22-65:~#
```

Below the terminal window, the Mobaxterm interface includes a file browser on the left and various system icons at the top. A status bar at the bottom indicates 'UNREGISTERED VERSION'.

You can see now that we have two replicas and all the two replicas are running.

### 3.4 Installing Kube Audit for Trivy

We will scan Kubernetes Cluster for any Kind of issues, for that we have multiple tools like Trivy. Also, we have but Trivy may not work always. So, for that reason we are just going to go with Kube Audit. It also a tool that can be used for Scanning.

Go to this link

<https://github.com/shopify/kubeaudit/releases>

This screenshot shows the GitHub releases page for the `kubeaudit` repository. The release `v0.22.2` was published on Aug 21, 2024. The changelog notes the addition of a deprecation notice for Kubeaudit and recommends transitioning to `Kubebench`. The 'What's Changed' section lists updates to Go dependencies and the version number.

**Aug 21, 2024**  
github-actions  
v0.22.2  
92c3147

**kubeaudit-v0.22.2** Latest

- Added deprecation notice for Kubeaudit, planning for deprecation by October 2024.
- Recommended transition to [Kubebench](#) for similar functionality.

**Changelog**

- [7cfaffd](#) Deprecating notice for Kubeaudit ([#594](#))
- [7e8696a](#) Update go ([#581](#))
- [92c3147](#) Update version to v0.22.2 ([#595](#))

**What's Changed**

- Update go by [@lynnish](#) in [#581](#)
- Deprecating notice for Kubeaudit by [@Shariati](#) in [#594](#)
- Update version to v0.22.2 by [@Shariati](#) in [#595](#)

Scroll down to “Contributors”

This screenshot shows the GitHub releases page for the `kubeaudit` repository. The release `v0.22.1` was published on Nov 29, 2023. The changelog lists a capture exit code for `sarif` and a version bump. The assets section shows various binary files, with an orange arrow pointing to the `kubeaudit_0.22.2_linux_amd64` file.

**Nov 29, 2023**  
github-actions  
v0.22.1  
76fe452

**Contributors**  
lynnish and Shariati

**Assets** 11

<a href="#">kubeaudit_0.22.2_checksums.txt</a>	820 Bytes	Aug 21, 2024	
<a href="#">kubeaudit_0.22.2_darwin_amd64.tar.gz</a>	9.86 MB	Aug 21, 2024	
<a href="#">kubeaudit_0.22.2_darwin_arm64.tar.gz</a>	9.11 MB	Aug 21, 2024	
<a href="#">kubeaudit_0.22.2_linux_386.tar.gz</a>	8.95 MB	Aug 21, 2024	
<a href="#">kubeaudit_0.22.2_linux_amd64.tar.gz</a>	9.71 MB	Aug 21, 2024	
<a href="#">kubeaudit_0.22.2_linux_arm64.tar.gz</a>	8.72 MB	Aug 21, 2024	
<a href="#">kubeaudit_0.22.2_windows_386.tar.gz</a>	9.44 MB	Aug 21, 2024	
<a href="#">kubeaudit_0.22.2_windows_amd64.tar.gz</a>	9.97 MB	Aug 21, 2024	
<a href="#">kubeaudit_0.22.2_windows_arm64.tar.gz</a>	8.79 MB	Aug 21, 2024	
<a href="#">Source code (zip)</a>		Aug 21, 2024	
<a href="#">Source code (tar.gz)</a>		Aug 21, 2024	

**kubeaudit-v0.22.1**

**Changelog**

- [db8a302](#) capture exit code for sarif ([#576](#))
- [76fe452](#) version bump; patch; prepare release ([#577](#))

Right-click on “`kubeaudit_0.22.2_Linux_amd_64`”

The screenshot shows a browser window with the URL [github.com/shopify/kubeaudit/releases](https://github.com/shopify/kubeaudit/releases). The page displays the 'Contributors' section and a list of 'Assets' for the v0.22.2 release. One of the assets is 'kubeaudit\_0.22.2\_linux\_amd64.tar.gz'. A context menu is open over this file, with the option 'Copy link address' highlighted.

## Select “copy the link address”

[https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit\\_0.22.2\\_linux\\_amd64.tar.gz](https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz)

Then go to the “Master” terminal and run the command:

```
wget https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz
```

The screenshot shows a MobaXterm window titled 'Master'. In the terminal session, the command `wget https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz` is being executed. The output shows the progress of the download, indicating a speed of 9.71M and a total size of 111 MB/s. The terminal also shows the file being saved to the current directory.

```
root@ip-172-31-22-65:~# wget https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz
--2026-01-03 05:40:55-- https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz
Resolving github.com (github.com)... 140.82.112.3:443... connected.
Connecting to github.com (github.com) [140.82.112.3]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset[...]
...
2026-01-03T06:39:47Z [download] 30 kubeaudit_0.22.2_linux_amd64.tar.gz saved [10181094/10181094]
```

Extract using the command:

```
tar -xvf kubeaudit_0.22.2_linux_amd64.tar.gz
```

```
Master
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiSec Tunneling Packages Settings Help
Quick connect...
Reconnect SSH-browser
S12 Master 13 Slave-1 14 Slave-2
root@ip-172-31-22-65:~# vi nginx-deploy.yaml
root@ip-172-31-22-65:~# ls
untit.sh  installing-kubernetes.sh  kubectl.sha256  nginx-deploy.yaml  snap
root@ip-172-31-22-65:~# kubectl apply -f nginx-deploy.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-22-65:~# kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2      2           2           4m22s
root@ip-172-31-22-65:~# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-deployment-86dcfcfd4c-wplkf   1/1      Running   0          7m27s
nginx-deployment-86dcfcfd4c-wpllf   1/1      Running   0          7m27s
root@ip-172-31-22-65:~# kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2      2           2           9m2s
root@ip-172-31-22-65:~# wget https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz
--2926-01-05 04:58:45-- https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz
Resolving github.com (github.com)... 140.82.112.3
Connecting to github.com (github.com)|140.82.112.3|:443... connected.
HTTP request sent, awaiting response... 304 Found
Location: https://release-assets.githubusercontent.com/github-production-release-asset/103579225/b7d91b50-e7f1-4085-9d2a-5db1ff0f7779sp=r&sv=2018-11-09&sr=b&spr=https&se=2026-01-03T09:34:00Z&st=attachment%3Bfilename%3Dkubeaudit_0.22.2_linux_amd64.tar.gz&sc=applications%2Foctet-stream&skid=2d410-5711-43a1-aedd-ab1947aa7ab8&ktsid=398a6654-997b-47e9-b12b-9515b964de6skt=2026-01-03T05%3A31%3A772&ses=2026-01-03T08%3A32%3A047&sks=b&svs=2018-11-09&gs=GS8n7gg1TEyfU%20%7fN%2FsTRNNDP4CvJWkLhbg101Ju21N19.yevlp3M0!uaXr0iuY9tIw1VYXKtjocmVs7WFz751h3N1dhMz210hVdXNlcmVnRnb0uY9tIw1a2v51oja2v5MS1mV4C16MTC2h2o2x0TEIN1wlmjmjoxNz3NDE400U2LChYXR0jioicmVs7WFz2WFzC2WFchJyHViGd1vb51b9G1LnNcmulu2lu693cysuX0if0.goknPmQg11t(kKH40W1z4d1mno0q1w1Bw&response-content-disposition=attachment%3B20f1lename%3Dkubeaudit_0.22.2_linux_amd64.tar.gz&response-content-type=application%2Foctet-stream%20(following)
--2926-01-05 05:00:00-- https://release-assets.githubusercontent.com/github-production-release-asset/103579225/b7d91b50-e7f1-4085-9d2a-5db1ff0f7779sp=r&sv=2018-11-09&sr=b&spr=https&se=2026-01-03T09:34:00Z&st=attachment%3Bfilename%3Dkubeaudit_0.22.2_linux_amd64.tar.gz&sc=applications%2Foctet-stream&skid=2d410-5711-43a1-aedd-ab1947aa7ab8&ktsid=398a6654-997b-47e9-b12b-9515b964de6skt=2026-01-03T05%3A31%3A772&ses=2026-01-03T08%3A32%3A047&sks=b&svs=2018-11-09&gs=GS8n7gg1TEyfU%20%7fN%2FsTRNNDP4CvJWkLhbg101Ju21N19.yevlp3M0!uaXr0iuY9tIw1VYXKtjocmVs7WFz751h3N1dhMz210hVdXNlcmVnRnb0uY9tIw1a2v51oja2v5MS1mV4C16MTC2h2o2x0TEIN1wlmjmjoxNz3NDE400U2LChYXR0jioicmVs7WFz2WFzC2WFchJyHViGd1vb51b9G1LnNcmulu2lu693cysuX0if0.goknPmQg11t(kKH40W1z4d1mno0q1w1Bw&response-content-disposition=attachment%3B20f1lename%3Dkubeaudit_0.22.2_linux_amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving release-assets.githubusercontent.com (release-assets.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10181994 (9.7M) [application/octet-stream]
Saving to: "kubeaudit_0.22.2_linux_amd64.tar.gz"

kubeaudit_0.22.2_linux_amd64.tar.gz 100%[=====] 9.71M --.-KB/s in 0.09s

2026-01-03 05:40:56 (111 MB/s) - "kubeaudit_0.22.2_linux_amd64.tar.gz" saved [10181994/10181994]

root@ip-172-31-22-65:~# tar -xvf kubeaudit_0.22.2_linux_amd64.tar.gz
README.md
kubeaudit
root@ip-172-31-22-65:~#
```

Then, move the executable file using the command:

```
sudo mv kubeaudit /usr/local/bin/
```

The screenshot shows the Mobaxterm application window. At the top, there's a menu bar with 'Master' selected, followed by 'Terminal', 'Sessions', 'View', 'X server', 'Tools', 'Games', 'Settings', 'Macros', and 'Help'. Below the menu is a toolbar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. The main area has a 'Quick connect...' search bar. There are four terminal tabs open: '12 Master' (root shell), '13 Slave-1' (root shell), '14 Slave-2' (root shell), and '15 Slave-3' (root shell). The '12 Master' tab shows the command 'root@lp-172-31-22-65:~# sudo mv kubeaudit /usr/local/bin/' being typed. To the left of the terminals is a file browser window titled '/home/ubuntu/'. It shows a directory structure with a file named 'Name'. The bottom of the screen features a dock with 'Remote monitoring' and 'Follow terminal folder' options.

Then, execute the command:

```
kubeaudit all
```

The screenshot shows a MobaXterm window titled "Master" with three tabs open: "15 Master", "16 Slave-1", and "17 Slave-2". The "15 Master" tab is active and displays the following error log from a kube-proxy container:

```
name: kube-proxy
namespace: kube-system

-----
-- [error] AppArmorAnnotationMissing
Message: AppArmor annotation missing. The annotation 'container.apparmor.security.beta.kubernetes.io/kube-proxy' should be added.
Metadata:
  Container: kube-proxy
  MissingAnnotation: container.apparmor.security.beta.kubernetes.io/kube-proxy

-- [error] CapabilityOrSecurityContextMissing
Message: Security Context not set. The Security Context should be specified and all Capabilities should be dropped by setting the Drop list to ALL.
Metadata:
  Container: kube-proxy

-- [error] NamespaceHostNetworkTrue
Message: hostNetwork is set to 'true' in PodSpec. It should be set to 'false'.

-- [warning] LimitsNotSet
Message: Resource limits not set.
Metadata:
  Container: kube-proxy

-- [error] RunAsNonRootPSCNilCSCNil
Message: runAsNonRoot should be set to true or runAsUser should be set to a value > 0 either in the container SecurityContext or PodSecurityContext.
Metadata:
  Container: kube-proxy

-- [error] AllowPrivilegeEscalationNil
Message: allowPrivilegeEscalation not set which allows privilege escalation. It should be set to 'false'.
Metadata:
  Container: kube-proxy

-- [error] PrivilegedTrue
Message: privileged is set to 'true' in container SecurityContext. It should be set to 'false'.
Metadata:
  Container: kube-proxy

-- [error] ReadOnlyRootFilesystemNil
Message: readOnlyRootFilesystem is not set in container SecurityContext. It should be set to 'true'.
Metadata:
  Container: kube-proxy

-- [error] SeccompProfileMissing
Message: Pod Seccomp profile is missing. Seccomp profile should be added to the pod SecurityContext.
root@ip-172-31-22-72:~#
```

At the bottom of the terminal window, there is a message: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

### 3.5 Create and Configure Virtual Machine for SonarQube

We are going to create multiple virtual machines in a secured network. We are going to set up servers for SonarQube and Nexus. Then for tools like Jenkins and monitoring tools.

#### 3.5.1 Create Virtual Machine for SonarQube Server

We will start by creating the virtual machine for SonarQube called “**SonarQube**”. Go AWS Management console.

The screenshot shows the AWS EC2 'Launch an instance' interface. In the 'Name and tags' section, the 'Name' field contains 'e.g. My Web Server'. An orange arrow points from this field to the 'Software Image (AMI)' section. The 'Software Image (AMI)' dropdown is open, showing 'Amazon Linux 2023 kernel-6.1 AMI' with the identifier 'ami-068c0051b15cdb816'. The 'Virtual server type (instance type)' is set to 't3.micro'. The 'Launch instance' button is highlighted in orange at the bottom right.

Let us give the virtual machine a name, we will call it “**SonarQube**”

The screenshot shows the 'Name and tags' section of the 'Launch an instance' page. The 'Name' field contains 'SonarQube'. The 'Add additional tags' link is visible to the right. The rest of the page is mostly hidden or identical to the previous screenshot.

Then on “Application and OS Images (Amazon Machine Image)” and select “**Ubuntu**”

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

**Recents**    **Quick Start**



Amazon Linux



macOS



Ubuntu



Windows



Red Hat



SUSE Linux



Debian

🔍 [Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
 ami-0ecb62995f68bb549 (64-bit (x86)) / ami-01b9f1e7dc427266e (64-bit (Arm))  
 Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username	Verified provider
64-bit (x86) ▾	ami-0ecb62995f68bb549	2025-10-22	ubuntu	<span style="background-color: green; color: white; padding: 2px 5px;">Verified provider</span>

Scroll down to “**Instance Type**” and select “**t2.medium**”

▼ Instance type [Info](#) | [Get advice](#)

**Instance type**

t2.medium  
 Family: t2 2 vCPU 4 GiB Memory Current generation: true  
 On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour On-Demand Linux base pricing: 0.0464 USD per Hour  
 On-Demand RHEL base pricing: 0.0752 USD per Hour On-Demand Windows base pricing: 0.0644 USD per Hour  
 On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations
[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

Scroll down to “**Key Pair**” and select the key we created previously

▼ Instance type [Info](#) | [Get advice](#)

**Instance type**

t2.medium  
 Family: t2 2 vCPU 4 GiB Memory Current generation: true  
 On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour On-Demand Linux base pricing: 0.0464 USD per Hour  
 On-Demand RHEL base pricing: 0.0752 USD per Hour On-Demand Windows base pricing: 0.0644 USD per Hour  
 On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations
[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

Scroll down to “**Network Settings**” and select the security group we created.

**▼ Network settings** [Info](#)

[Edit](#)

**Network** | [Info](#)  
vpc-0d74d3736a240e572

**Subnet** | [Info](#)  
No preference (Default subnet in any availability zone)

**Auto-assign public IP** | [Info](#)  
Enable

**Firewall (security groups)** | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group       Select existing security group

**Common security groups** | [Info](#)

Select security groups ▾

Primary-SG sg-002d4edfb66259799 [X](#)  
VPC: vpc-0d74d3736a240e572

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Scroll down to “Configure Storage” and make it “20GiB”

**▼ Configure storage** [Info](#)

[Advanced](#)

1x  GiB [gp3](#) Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

**▶ Advanced details** [Info](#)

**Storage (volumes)**  
1 volume(s) - 20 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

Then, click on “Launch Instance”

The screenshot shows the AWS EC2 Instances launch success page. At the top, there is a green success message: "Success Successfully initiated launch of instance (i-072e7267a412f30e2)". Below this, there is a "Launch log" link. A red arrow points from the text "Click on ‘Instances’" to the "Instances" link in the breadcrumb navigation bar. The main content area is titled "Next Steps" and contains several cards with links to other AWS services:

- Create billing usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds. Link: [Create billing alerts](#)
- Connect to your instance**: Once your instance is running, log into it from your local computer. Link: [Connect to instance](#)
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Link: [Connect an RDS database](#)
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Link: [Create EBS snapshot policy](#)
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. Link: [Manage detailed monitoring](#)
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Link: [Create Load Balancer](#)
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. Link: [Create AWS budget](#)
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Link: [Manage CloudWatch alarms](#)

At the bottom, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and links for Privacy, Terms, and Cookie preferences.

Click on “Instances”

The screenshot shows the AWS EC2 Instances list page. On the left, there is a navigation sidebar with categories like EC2, Instances, Images, Elastic Block Store, Network & Security, and more. A red arrow points from the text "Click on ‘Instances’" to the "Instances" link in the sidebar. The main content area is titled "Instances (4) Info" and displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0d72ba387d698a6b0	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Slave-1	i-0b025e08ef003624c	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Slave-2	i-0afb570889b99b7d9	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
SonarQube	i-072e7267a412f30e2	Running	t2.medium	Initializing	<a href="#">View alarms +</a>	us-east-1c

A red arrow points from the text "We have created the instance, let us wait for it to pass the “2/2 check”" to the "SonarQube" instance in the list.

We have created the instance, let us wait for it to pass the “2/2 check”

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links for EC2, Instances, Images, Elastic Block Store, Network & Security, and more. The main area displays a table of instances:

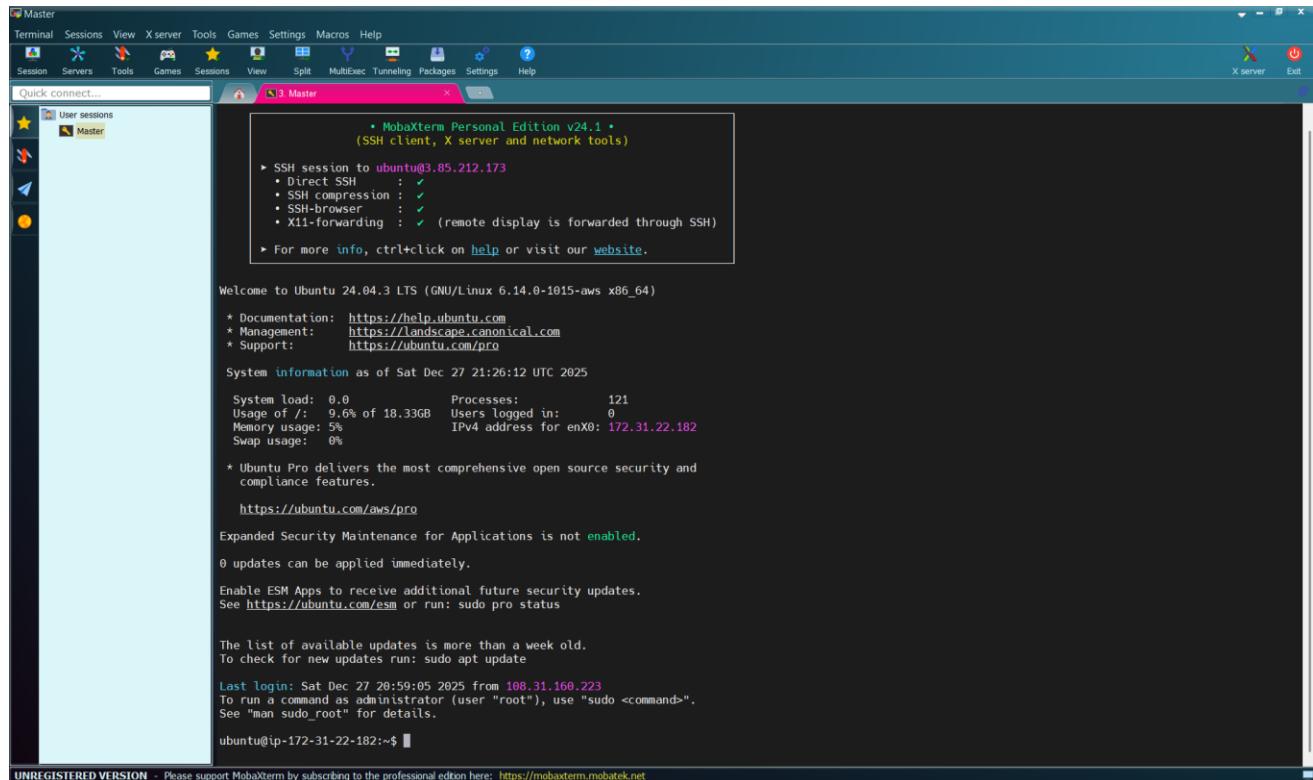
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0d72ba387d698a6b0	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Slave-1	i-0b025e08ef003624c	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Slave-2	i-0afb570889b99b7d9	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
SonarQube	i-072e7267a412f30e2	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c

An orange arrow points to the "2/2 checks passed" status for the SonarQube instance.

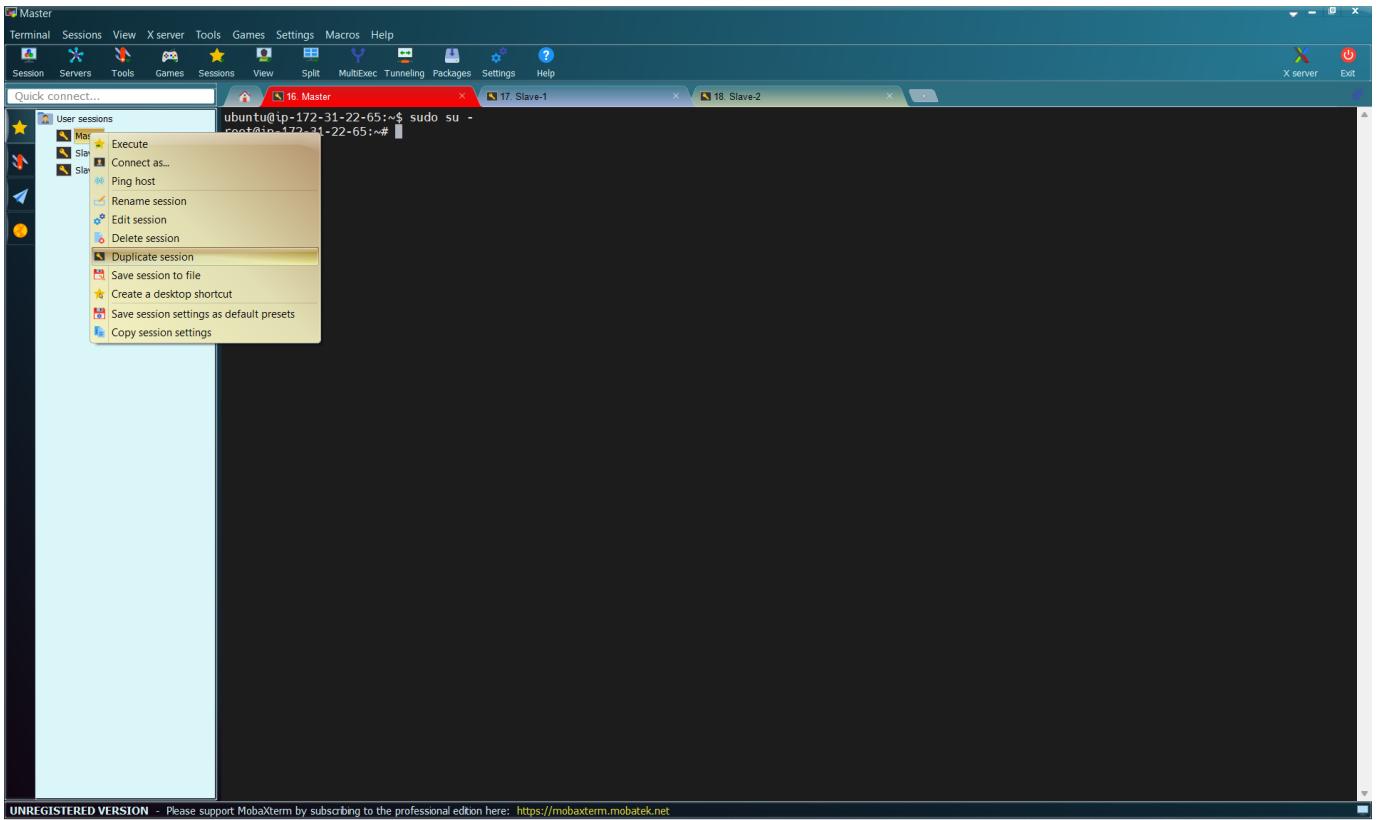
The virtual machine has passed the “2/2 Check”

### 3.5.2 SSH Connect to SonarQube Server

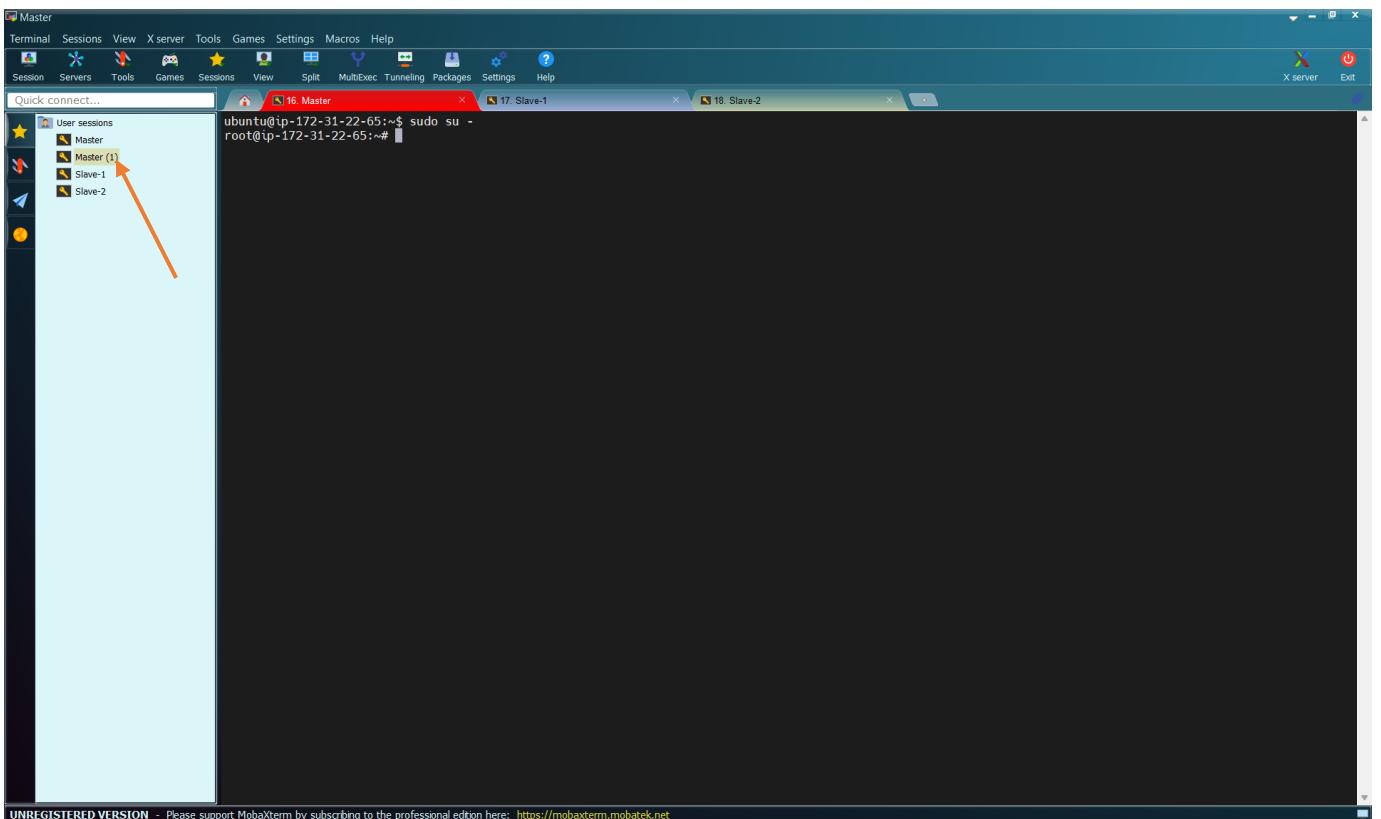
Let us create a duplicate of the session “SonarQube”.



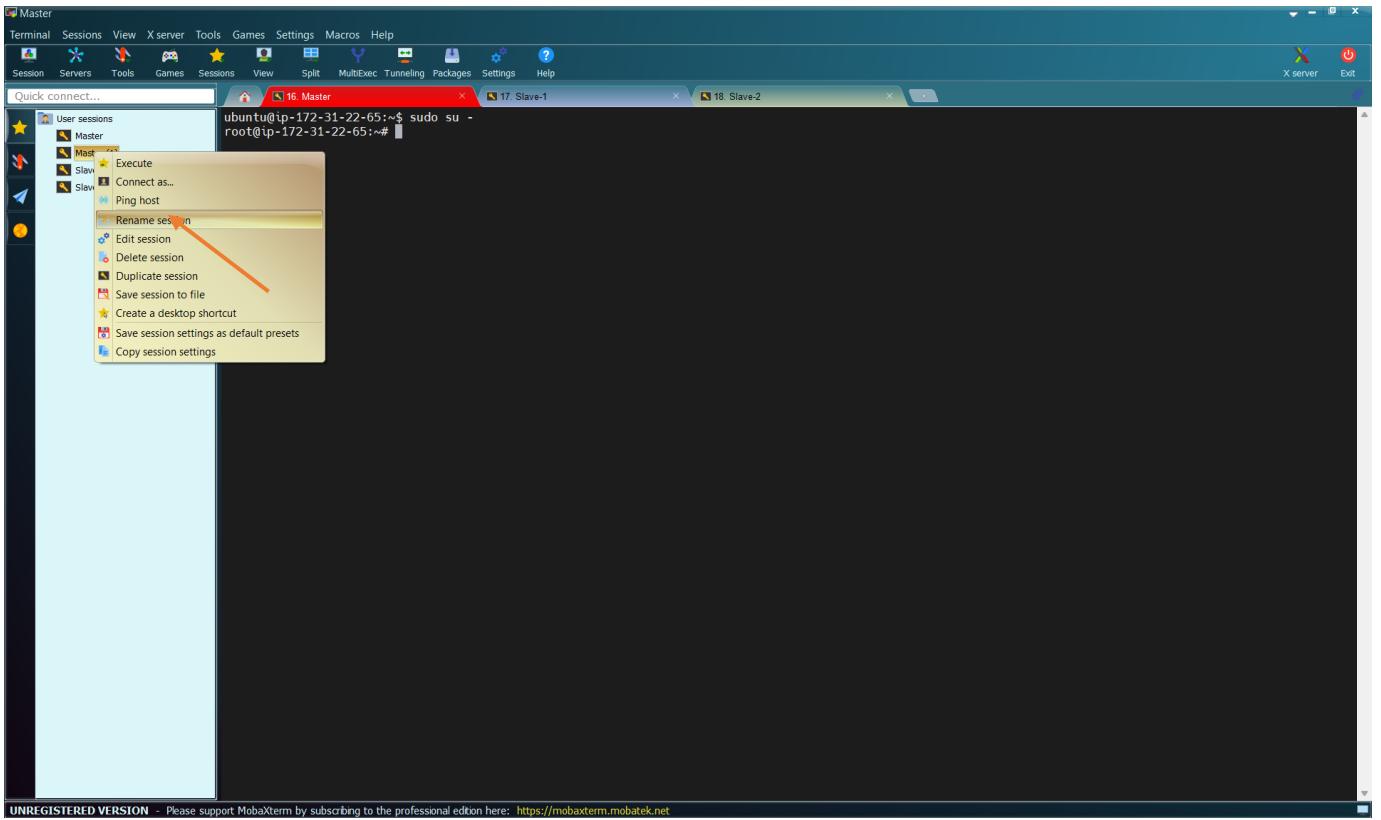
Right-click on the session name “Master”



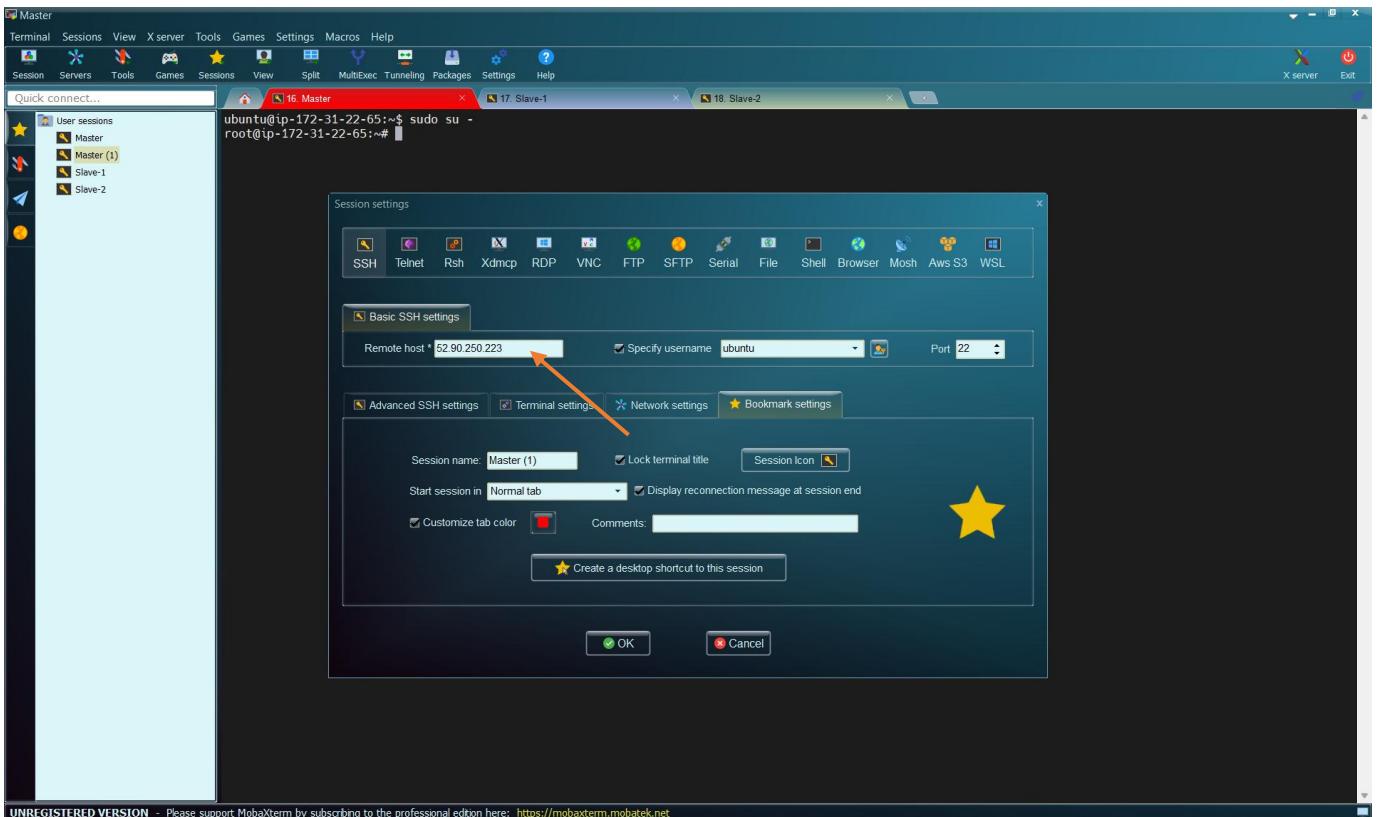
Select “Duplicate Session”



We want to rename “Master(1)” to “SonarQube”. Right-click on “Master(1)”



## Select “Rename Session”



Copy the Public IP address of our “SonarQube” virtual machine and paste here.

AWS EC2 Instances (1/5) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0d72ba387d698a6b0	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Slave-1	i-0b025e08ef003624c	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Slave-2	i-0afb570889b99b7d9	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
SonarQube	i-072e7267a412f30e2	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Nexus	i-0a783c8bc1d7f52ae	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c

i-072e7267a412f30e2 (SonarQube)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID: i-072e7267a412f30e2  
 Public IPv4 address: 3.80.70.234 [open address]  
 Instance state: Running  
 Private IP DNS name (IPv4 only): ip-172-31-25-143.ec2.internal  
 Instance type: t2.medium  
 VPC ID: vpc-0d74d3736a240e572 [open]

IPv6 address: -  
 Hostname type: IP name: ip-172-31-25-143.ec2.internal  
 Answer private resource DNS name: IPv4 (A)  
 Auto-assigned IP address: 3.80.70.234 [Public IP]  
 Private IPv4 addresses: 172.31.25.143  
 Public DNS: ec2-3-80-70-234.compute-1.amazonaws.com [open address]  
 Elastic IP addresses: -  
 AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.

Copy the Public IP: 3.80.70.234 and paste the MobaXterm

Master Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect... 16. Master 17. Slave-1 18. Slave-2

Session settings

Basic SSH settings: Remote host: 3.80.70.234, Specify username: ubuntu, Port: 22

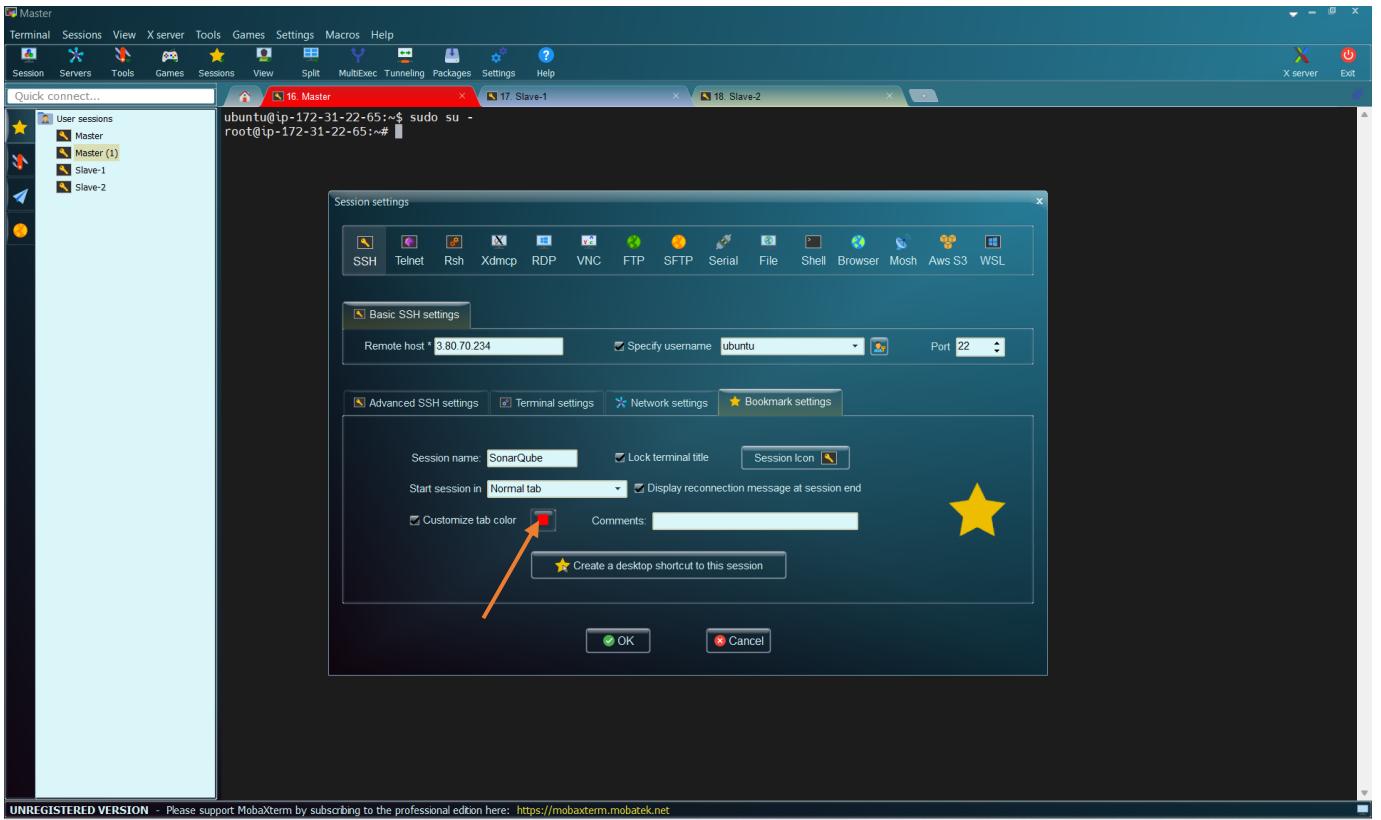
Advanced SSH settings: Session name: Master (1), Lock terminal title, Start session in: Normal tab, Display reconnection message at session end, Customize tab color, Comments: [ ]

OK Cancel

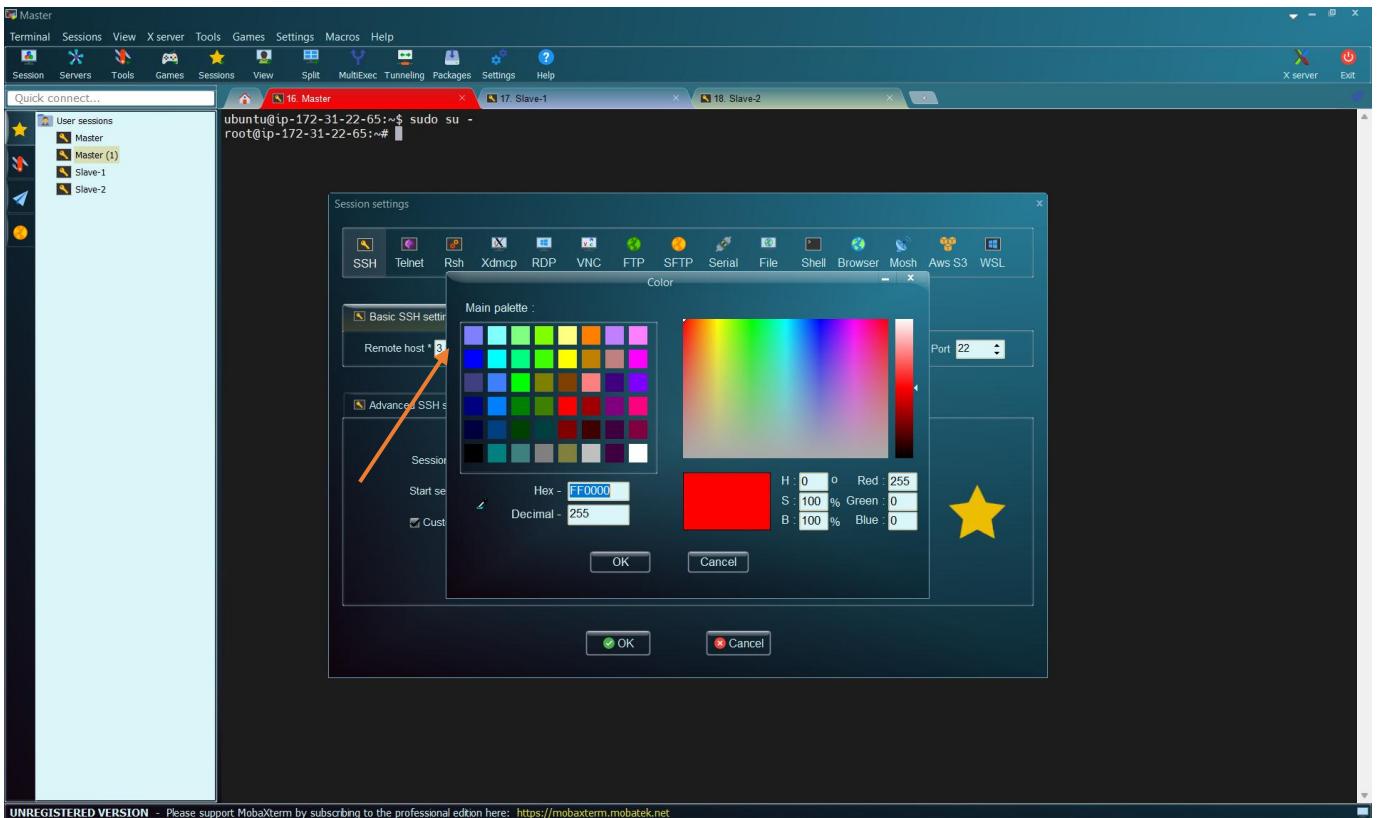
User sessions: Master, Master (1), Slave-1, Slave-2

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

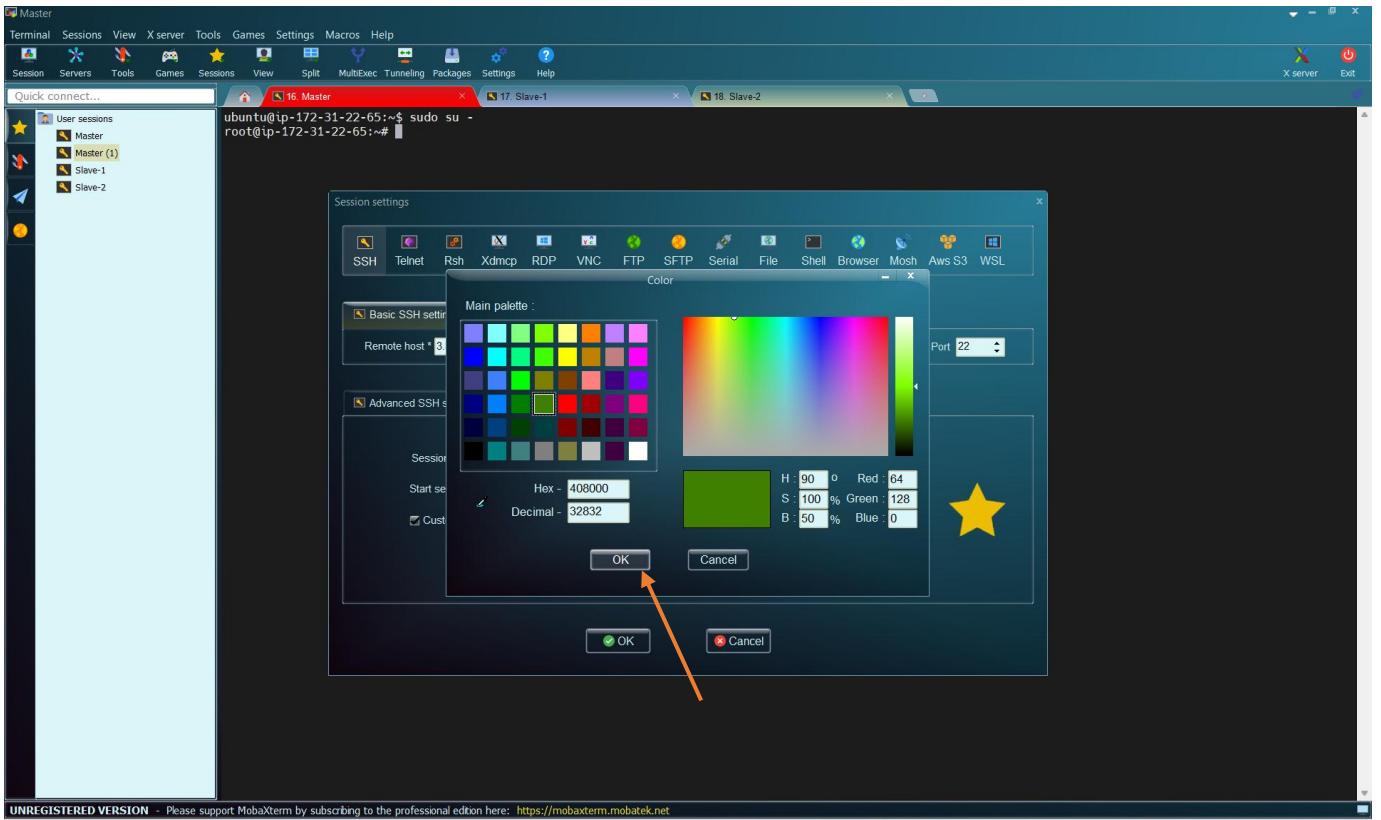
Then, change the name to "SonarQube"



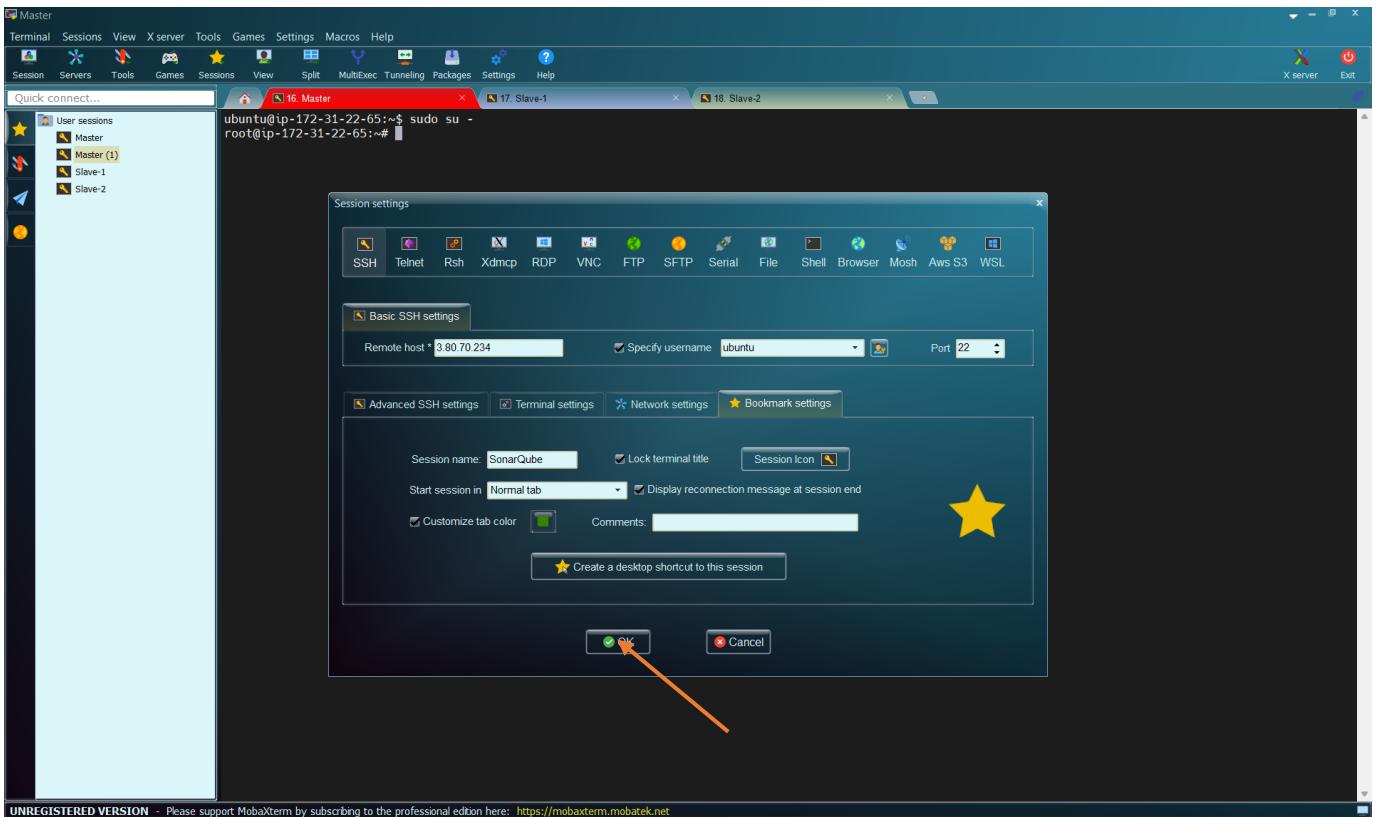
Then, let us change the color. Click there



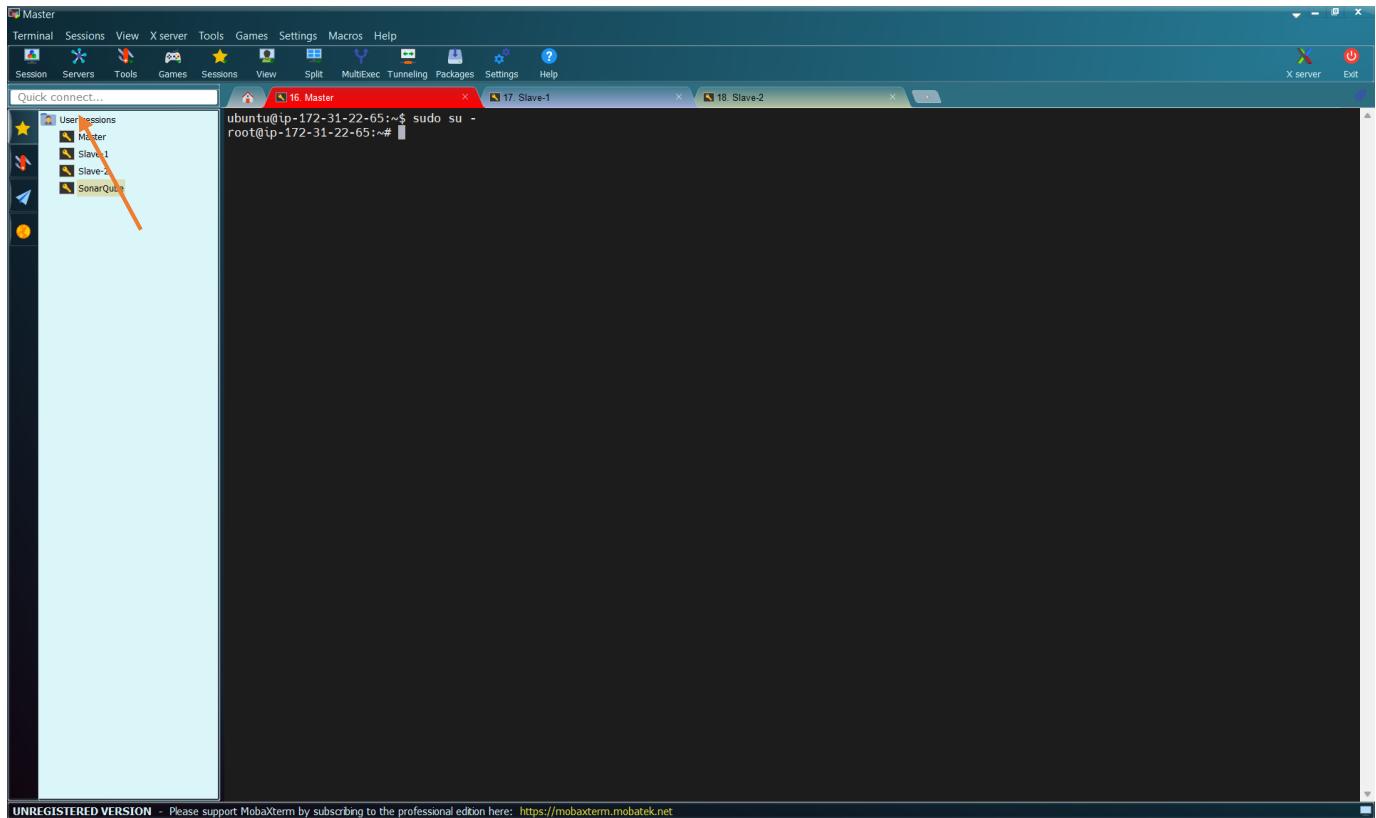
Let us select "Green"



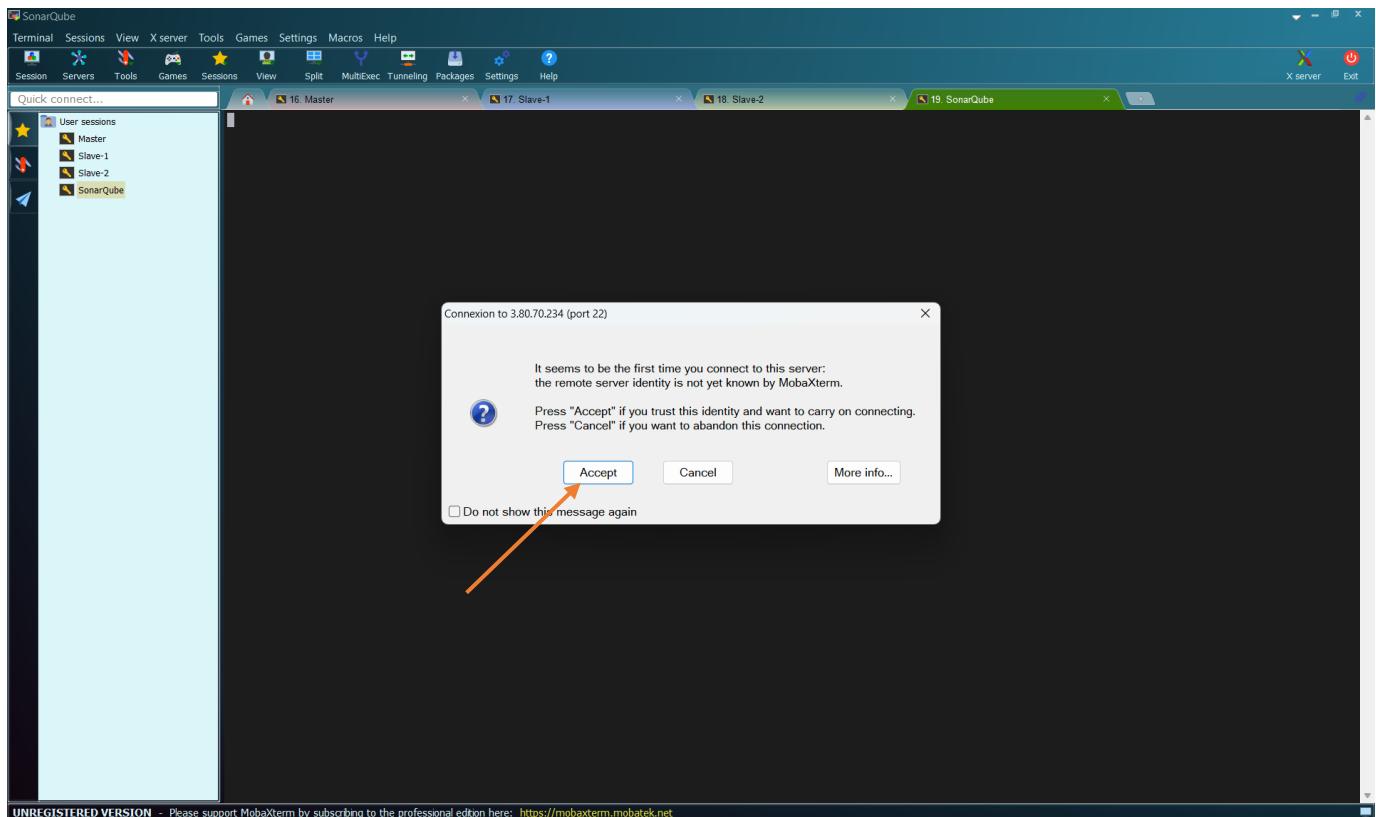
Click on “OK”



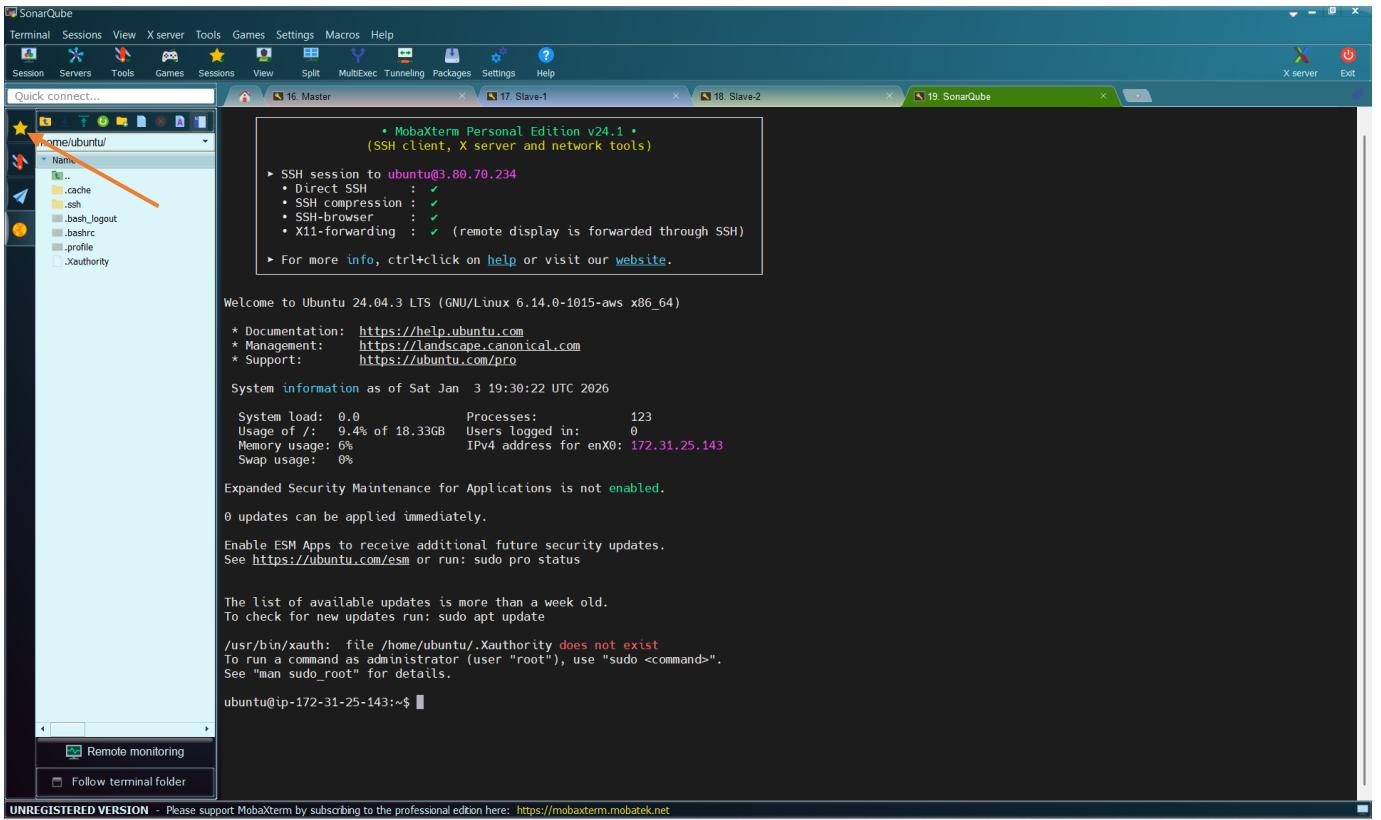
And click on “OK” again



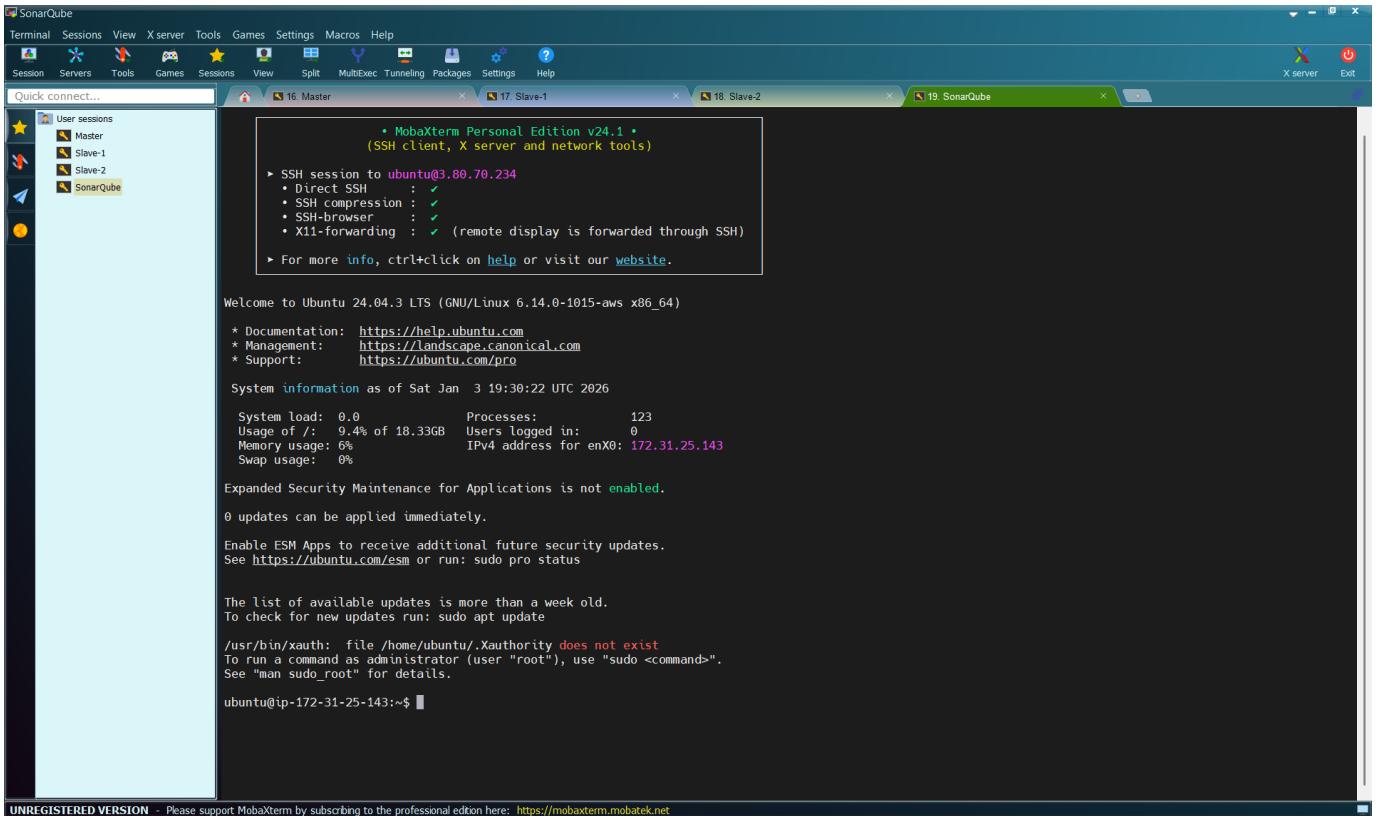
Then, double-click on the session “SonarQube”



Click on “Accept”



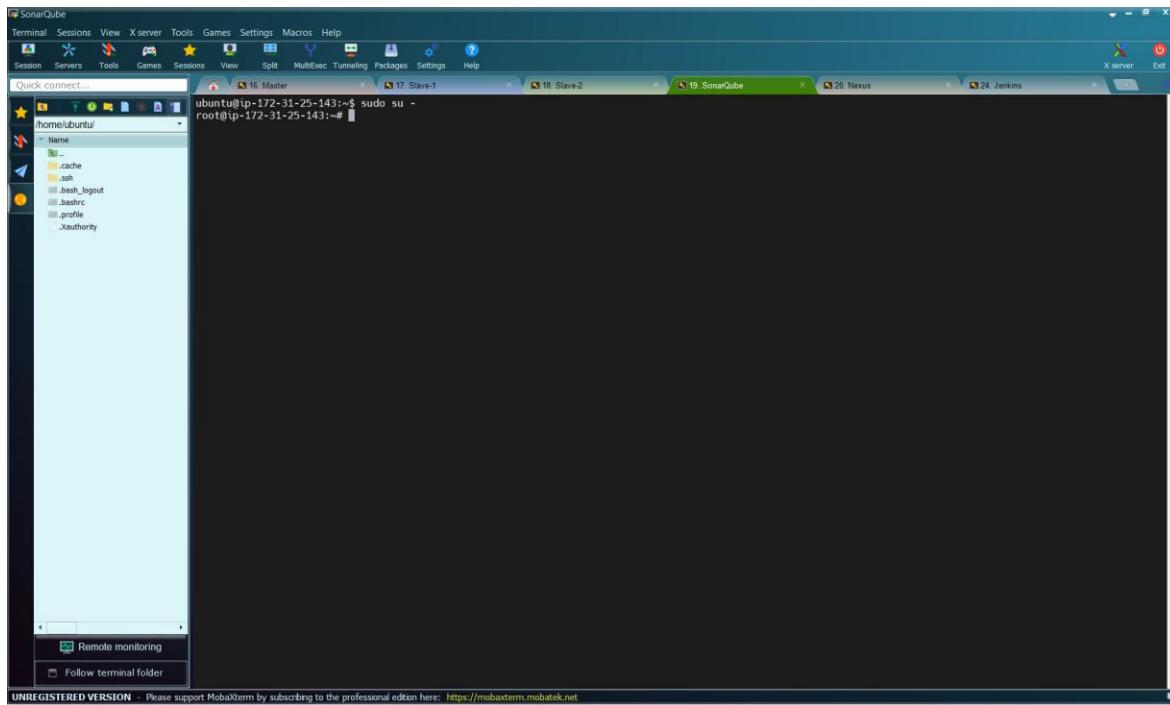
You can see that the tab of “SonarQube” is “Green”. Click on the star



### 3.5.3 Configure SonarQube Server

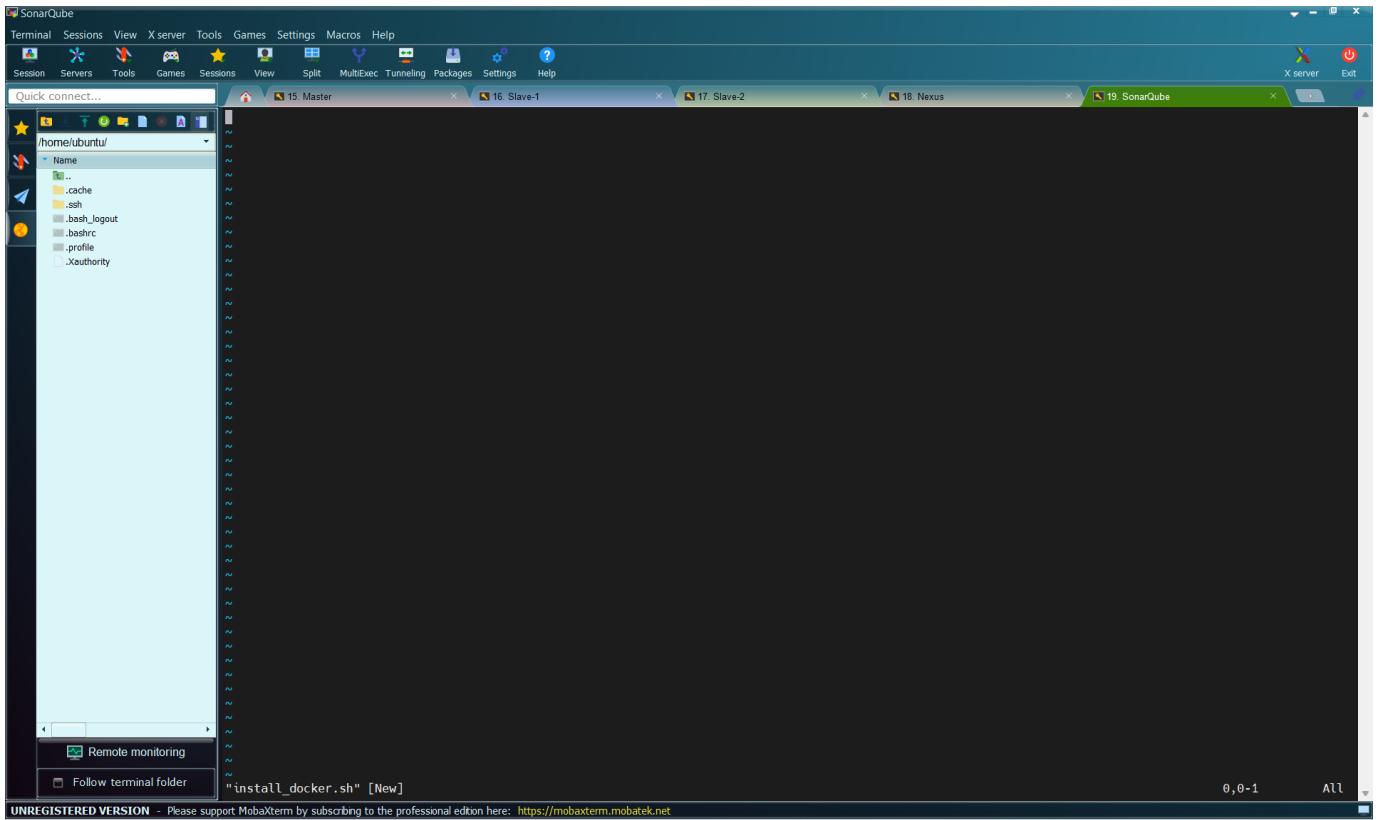
We will now configure the SonarQube server. Let us first grant it root user privilege by running the command:

```
sudo su -
```



We will now install Docker. To do this, we will use a shell script called “**install\_docker.sh**”. Create the script using the command:

```
vi install_docker.sh
```



## Paste the code to install Docker Package

```
#!/bin/bash

# Update package manager repositories
sudo apt-get update

# Install necessary dependencies
sudo apt-get install -y ca-certificates curl

# Create directory for Docker GPG key
sudo install -m 0755 -d /etc/apt/keyrings

# Download Docker's GPG key
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

# Ensure proper permissions for the key
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add Docker repository to Apt sources
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Update package manager repositories
sudo apt-get update

sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```
#!/bin/bash
# Update package manager repositories
sudo apt-get update
# Install necessary dependencies
sudo apt-get install -y ca-certificates curl
# Create directory for Docker GPG key
sudo install -m 0755 -d /etc/apt/keyrings
# Download Docker's GPG key
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
# Ensure proper permissions for the key
sudo chmod a+r /etc/apt/keyrings/docker.asc
# Add Docker repository to Apt sources
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$(lsb_release -sc) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
# Update package manager repositories
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

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

Save and Exit the file by pressing “**ESC**”, followed by **(:wq)** and then press “**Enter**”

```
root@ip-172-31-19-153:~# vi install_docker.sh
root@ip-172-31-19-153:~#
```

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

Give the file an executable permission using the command:

```
chmod +x install_docker.sh
```

```

root@ip-172-31-19-153:~# vi install_docker.sh

```

Then, execute the file using the command:

```
./install_docker.sh
```

```

Selecting previously unselected package docker-ce.
Preparing to unpack .../2-docker-ce_5%3a29.1.4-1~ubuntu.24.04-noble_amd64.deb ...
Unpacking docker-ce (5:29.1.4-1~ubuntu.24.04-noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../3-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../4-docker-buildx-plugin_0.30.1-1~ubuntu.24.04-noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.30.1-1~ubuntu.24.04-noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a29.1.4-1~ubuntu.24.04-noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04-noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_5.0.1-1~ubuntu.24.04-noble_amd64.deb ...
Unpacking docker-compose-plugin (5.0.1-1~ubuntu.24.04-noble) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../7-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../8-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.30.1-1~ubuntu.24.04-noble) ...
Setting up containerd.io (2.2.1-1~ubuntu.24.04-noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (5.0.1-1~ubuntu.24.04-noble) ...
Setting up docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04-noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04-noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:29.1.4-1~ubuntu.24.04-noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-19-153:~#

```

Docker has been installed but at the moment other users cannot access it. Only the root user can access it. In order to grant access to other users, we have to run the command:

```
sudo chmod 666 /var/run/docker.sock
```

```

SonarQube
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15. Master 16. Slave-1 17. Slave-2 18. Nexus 19. SonarQube X server Exit
/home/ubuntu/
Name
.. .cache .ssh .bash_logout .bashrc .profile .xauthority
Preparing to unpack .../2-docker-ce_5%3a29.1.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package pigz ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../4-docker-buildx-plugin_0.30.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a29.1.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_5.0.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../7-libslirp0_4.7.0~ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0~ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../8-slirp4netns_1.2.1~1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1~1build2) ...
Setting up docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (2.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service -> /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Setting up docker-ce-cli (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up docker-ce-pkcs11 (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1~1build2) ...
Setting up docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service -> /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket -> /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu0.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-19-153:~# sudo chmod 666 /var/run/docker.sock
root@ip-172-31-19-153:~#

```

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

Let us run the command to verify that Docker is running:

```
sudo systemctl status docker
```

```

SonarQube
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15. Master 16. Slave-1 17. Slave-2 18. Nexus 19. SonarQube X server Exit
/home/ubuntu/
Name
.. .cache .ssh .bash_logout .bashrc .profile .xauthority
Setting up docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Setting up docker-ce-cli (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0~ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1~1build2) ...
Setting up docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service -> /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket -> /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu0.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-19-153:~# sudo chmod 666 /var/run/docker.sock
root@ip-172-31-19-153:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Sun 2026-01-11 17:00:25 UTC; 1min 23s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
      Main PID: 2465 (dockerd)
        Tasks: 9
       Memory: 25.9M (peak: 26.4M)
         CPU: 29.3ms
        CGroup: /system.slice/docker.service
                └─2465 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jan 11 17:00:24 ip-172-31-19-153 dockerd[2465]: t:times"2026-01-11T17:00:24.861324351Z" level=info msg="Restoring containers: start."
Jan 11 17:00:24 ip-172-31-19-153 dockerd[2465]: t:times"2026-01-11T17:00:24.891498634Z" level=info msg="Deleting nftables IPv4 rules" error="exit status 1"
Jan 11 17:00:24 ip-172-31-19-153 dockerd[2465]: t:times"2026-01-11T17:00:24.896579872Z" level=info msg="Deleting nftables IPv6 rules" error="exit status 1"
Jan 11 17:00:25 ip-172-31-19-153 dockerd[2465]: t:times"2026-01-11T17:00:25.188087862Z" level=info msg="Loading containers: done."
Jan 11 17:00:25 ip-172-31-19-153 dockerd[2465]: t:times"2026-01-11T17:00:25.188766264Z" level=info msg="Docker daemon" commit=08440b6 containerd-snapshotter=tr
Jan 11 17:00:25 ip-172-31-19-153 dockerd[2465]: t:times"2026-01-11T17:00:25.188849871Z" level=info msg="Initializing buildkit"
Jan 11 17:00:25 ip-172-31-19-153 dockerd[2465]: t:times"2026-01-11T17:00:25.232618401Z" level=info msg="Completed buildkit initialization"
Jan 11 17:00:25 ip-172-31-19-153 systemd[1]: Started docker.service - Docker Application Container Engine.
Jan 11 17:00:25 ip-172-31-19-153 systemd[1]: docker.service - Docker Application Container Engine started.
Jan 11 17:00:25 ip-172-31-19-153 dockerd[2465]: t:times"2026-01-11T17:00:25.237263337Z" level=info msg="Daemon has completed initialization"
Jan 11 17:00:25 ip-172-31-19-153 dockerd[2465]: t:times"2026-01-11T17:00:25.237585426Z" level=info msg="API listen on /run/docker.sock"
[Lines 1-22/22 (END)]
```

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

You can see that Docker is Active and running.

## Create Sonarqube Docker container

Let us create a Docker container for SonarQube using the command:

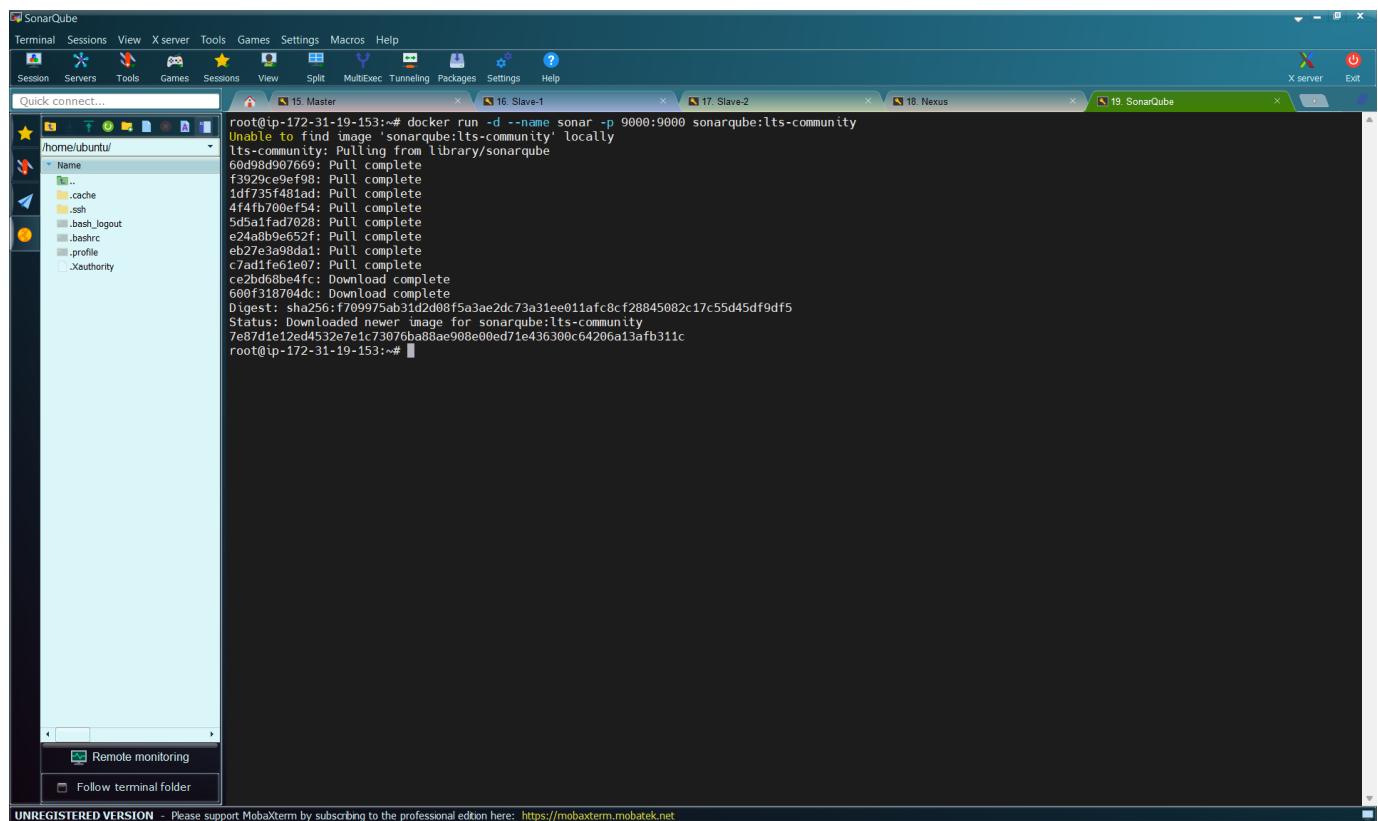
```
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

This command will download the sonarqube:lts-community Docker image from Docker Hub if it is not already available locally. Then, it will create a container named "sonar" from this image, running it in detached mode (-d flag) and mapping port 9000 on the host machine to port 9000 in the container (-p 9000:9000 flag).

Where:

**sonarqube** is the name of the image which we want to use to create this container

**lts-community** is the tag or the specific version we want to use.



The container has been created. Let us verify if the container has been created and it is running using the command:

```
docker ps
```

```

SonarQube
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15. Master 16. Slave-1 17. Slave-2 18. Nexus 19. SonarQube X server Exit

root@ip-172-31-19-153:~# docker run -d --name sonar 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
60d98d907669: Pull complete
f3929ce9ef98: Pull complete
1df7357481ad: Pull complete
4f4fb700ef54: Pull complete
5d5a1fad7028: Pull complete
e24aabb9e52f: Pull complete
eb27e3a98da1: Pull complete
c7ad1fe61e07: Pull complete
ce2bd68be4fc: Download complete
600f318704dc: Download complete
Digest: sha256:f799975ab312d2d08f5a3ae2dc73a31ee011afc8cf28845082c17c55d45df9df5
Status: Downloaded newer image for sonarqube:lts-community
7e87d1e12ed4532e7e1c73076ba88ae908e00ed71e436300c64206a13afb311c
root@ip-172-31-19-153:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
7e87d1e12ed4 sonarqube:lts-community "/opt/sonarqube/dock..." 26 seconds ago Up 25 seconds 0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp sonar
root@ip-172-31-19-153:#

```

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

We can see the Docker image is running with the name “**sonar**”

### 3.5.4 Accessing SonarQube through Browser

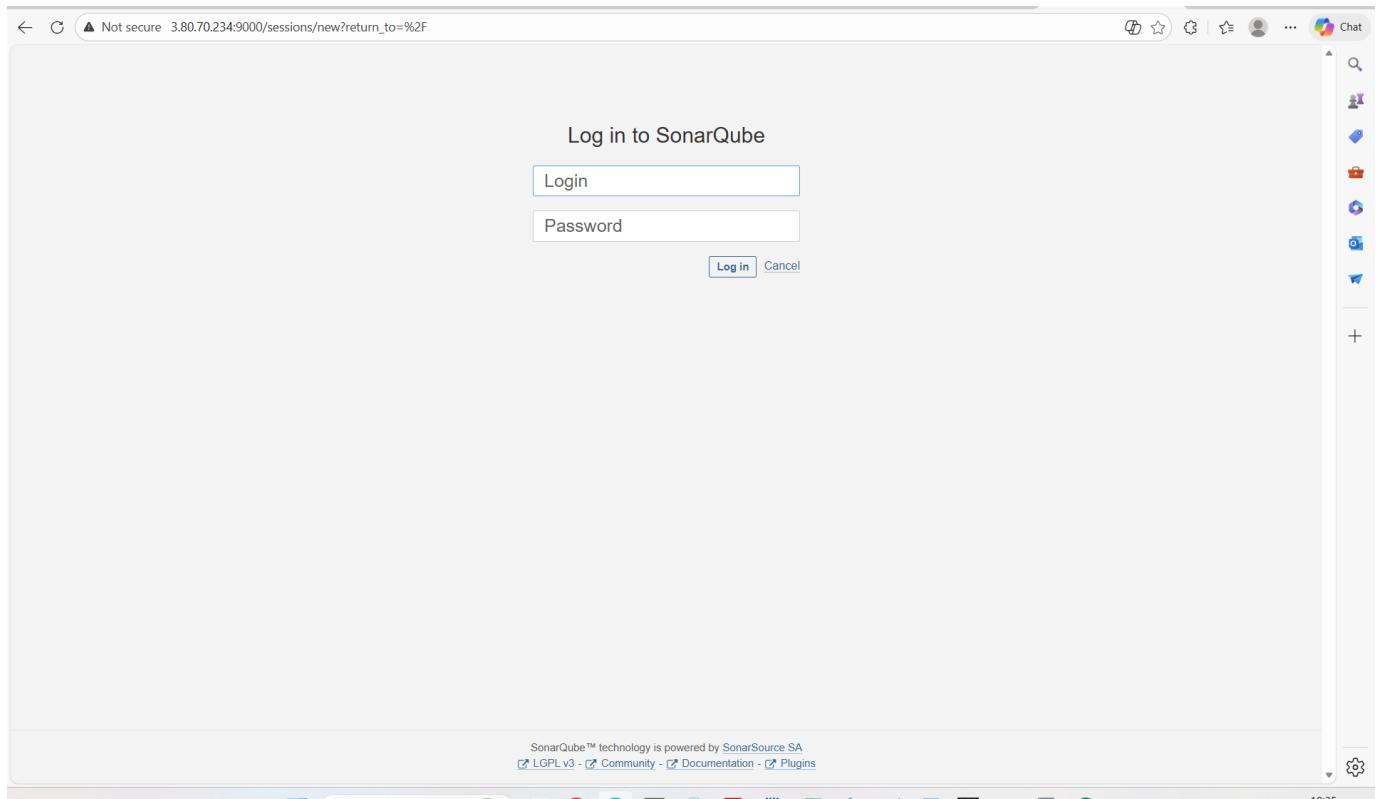
Let us access SonarQube through the browser by using the Public IP address of the SonarQube server. This is done as follows:

`http://<Public IP of SonarQube Server>:9000`

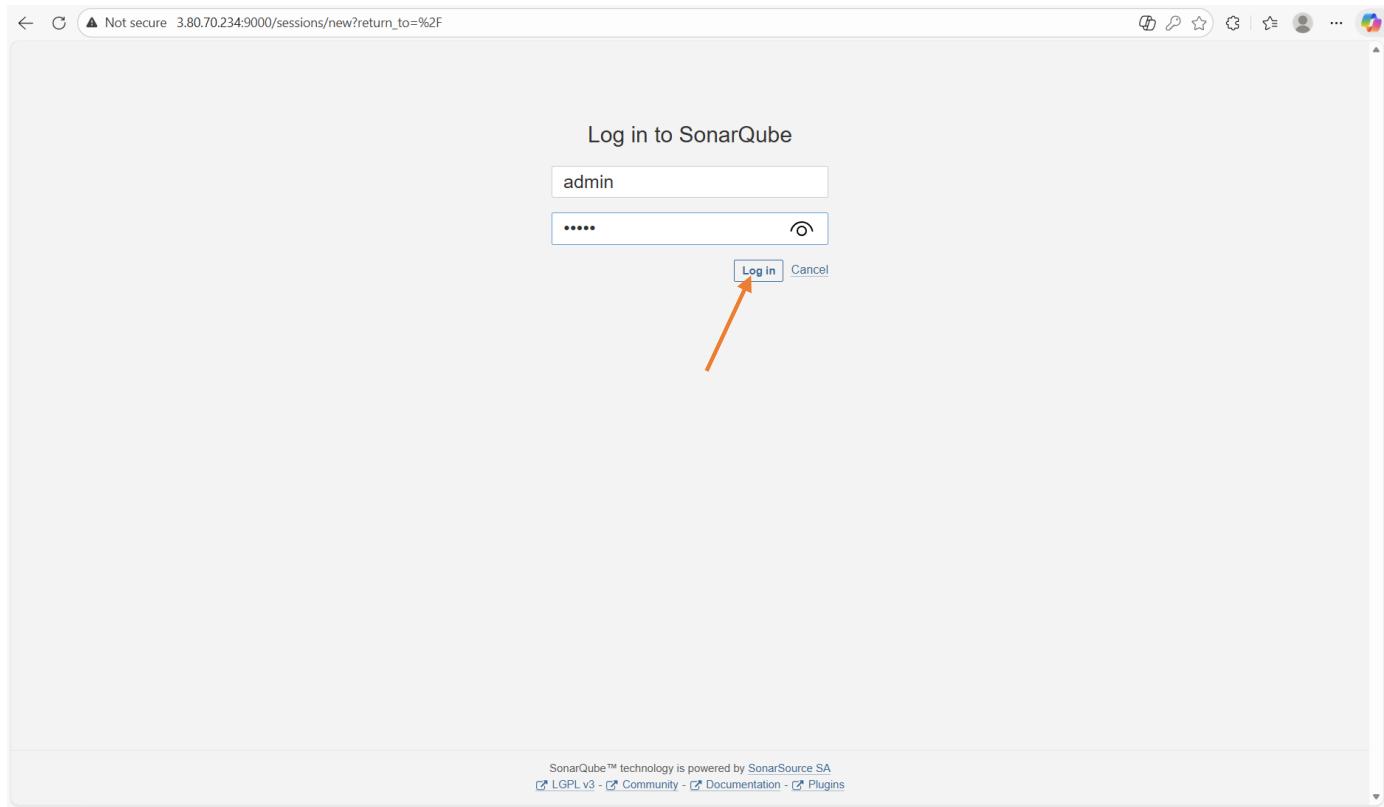
That is

`http://34.203.223.19:9000`

This will start the SonarQube server, and you should be able to access it using the provided URL. If you are running Docker on a remote server or a different port, replace localhost with the appropriate hostname or IP address and adjust the port accordingly.



Enter the default username and password. Both are “**admin**”



Click on “**Log in**”

Not secure 3.80.70.234:9000/account/reset\_password

Summarize ⚡ 🔍 ☆ ⚙️ 💾 🌐 ...

### Update your password

This account should not use the default password.

Enter a new password

All fields marked with \* are required

Old Password \*

New Password \*

Confirm Password \*

**Update**

You can now change the password. Enter the old password, followed by the new password and confirm it

Not secure 3.80.70.234:9000/account/reset\_password

Summarize ⚡ 🔍 ☆ ⚙️ 💾 🌐 ...

### Update your password

This account should not use the default password.

Enter a new password

All fields marked with \* are required

Old Password \*

.....

New Password \*

.....

Confirm Password \*

.....

**Update**

Then, click on “**update**”

The screenshot shows the SonarQube 'Create Project' page. At the top, a message says 'You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately.' Below this, there are five cards for importing from Azure DevOps, Bitbucket Server, Bitbucket Cloud, GitHub, and GitLab. Underneath these, a section for manual project creation is shown, featuring a 'Manually' button with a double-angle bracket icon. A yellow warning box at the bottom left states: '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.'

### 3.5.5 Generate SonarQube Token

We will now generate SonarQube Token. To do this, we go to the SonarQube browser

The screenshot shows the SonarQube 'Create Project' page. An orange arrow points to the 'Administration' tab in the top navigation bar. The rest of the page content is identical to the first screenshot, including the 'From Azure DevOps' card, the 'Manually' button, and the yellow warning box about the embedded database.

Click on “Administration”

You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately. [Learn More](#)

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects... A

Administration

Configuration Security Projects System Marketplace

General Settings

Edit global settings for this SonarQube instance.

Find in Settings

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.cpdl.cross\_project

Email

SMTP host

For example "smtp.gmail.com". Leave blank to disable email sending.

Key: email.smtp\_host.secured

Get the most out of SonarQube!

Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.

Learn More Dismiss

Click on the drop down on “security” and select “Users”

You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately. [Learn More](#)

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects... A

Administration

Configuration Security Projects System Marketplace

Users

Create and administer individual users.

Create User

Search by login or name...

SCM Accounts	Last connection	Groups	Tokens
	< 1 hour ago	sonar-administrators sonar-users	0

1 of 1 shown

Click on “Tokens”

Tokens of Administrator

Generate Tokens

Name Expires in

Enter Token Name  30 days

Name	Type	Project	Last use	Created	Expiration
No tokens					

Done

Give the token a name, we will call it “**sonar-token**”

The screenshot shows the 'Tokens of Administrator' page. In the 'Generate Tokens' section, the 'Name' field contains 'sonar-token' and the 'Expires in' dropdown is set to '30 days'. A blue arrow points from the text 'We will leave "Expires in" as "30days" and click on "Generate"' to the 'Generate' button. Below this section is a table with columns: Name, Type, Project, Last use, Created, and Expiration. The table currently displays 'No tokens'.

We will leave “Expires in” as “30days” and click on “Generate”

The screenshot shows the 'Tokens of Administrator' page after generating a token. The 'Generate Tokens' section now shows an empty 'Enter Token Name' field and the 'Expires in' dropdown still set to '30 days'. A blue arrow points from the text 'Copy the token: squ\_6ba30f3338a46b4d00a16948795a7c22e8e635c3' to the 'Copy' button. A yellow notification bar at the top states: 'New token "sonar-token" has been created. Make sure you copy it now, you won't be able to see it again!' Below the notification, the token value 'squ\_6ba30f3338a46b4d00a16948795a7c22e8e635c3' is displayed in a box with a 'Copy' button. The main table below the notification shows one row for the generated token: Name 'sonar-token', Type 'User', Project (empty), Last use 'Never', Created 'January 11, 2026', Expiration 'February 9, 2026', and a 'Revoke' button.

Copy the token: **squ\_6ba30f3338a46b4d00a16948795a7c22e8e635c3**

We will use this token later

## 3.6 Create and configure Virtual Machine for Nexus

Let us create and configure the virtual machine that will serve as our server for Nexus.

### 3.6.1 Create Virtual Machine for Nexus

We will start by creating the virtual machine for Nexus called “**Nexus**”. Go AWS Management console.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' step, the 'Name' field contains 'My Web Server'. In the 'Application and OS Images (Amazon Machine Image)' step, the 'Ubuntu' option is selected. On the right, the 'Summary' step shows 1 instance being launched with the 'Amazon Linux 2023 kernel-6.1 AMI' selected. Other configuration details like 'Virtual server type (instance type) t3.micro' and 'Storage (volumes) 1 volume(s) - 8 GiB' are also visible.

Let us give the virtual machine a name, we will call it “**Nexus**”

The screenshot shows the 'Name and tags' step of the 'Launch an instance' wizard. The 'Name' field contains 'Nexus'. The 'Add additional tags' button is visible next to the name field.

Then on “**Application and OS Images (Amazon Machine Image)**” and select “**Ubuntu**”

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

[Recents](#) [Quick Start](#)



Amazon Linux



macOS



Ubuntu



Windows



Red Hat



SUSE Linux



Debian

[Browse more AMIs](#) 

Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
 ami-0ecb62995f68bb549 (64-bit (x86)) / ami-01b9f1e7dc427266e (64-bit (Arm))  
 Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Publish Date	Username	<a href="#">i</a>
64-bit (x86) ▾	ami-0ecb62995f68bb549	2025-10-22	ubuntu	<span style="background-color: green; color: white; padding: 2px;">Verified provider</span>

Scroll down to “**Instance Type**” and select “**t2.medium**”

▼ Instance type [Info](#) | [Get advice](#)

**Instance type**

t2.medium  
 Family: t2 2 vCPU 4 GiB Memory Current generation: true  
 On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour On-Demand Linux base pricing: 0.0464 USD per Hour  
 On-Demand RHEL base pricing: 0.0752 USD per Hour On-Demand Windows base pricing: 0.0644 USD per Hour  
 On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

Scroll down to “**Key Pair**” and select the key we created previously

▼ Instance type [Info](#) | [Get advice](#)

**Instance type**

t2.medium  
 Family: t2 2 vCPU 4 GiB Memory Current generation: true  
 On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour On-Demand Linux base pricing: 0.0464 USD per Hour  
 On-Demand RHEL base pricing: 0.0752 USD per Hour On-Demand Windows base pricing: 0.0644 USD per Hour  
 On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

Scroll down to “**Network Settings**” and select the security group we created.

**▼ Network settings** [Info](#)

[Edit](#)

**Network** | [Info](#)  
vpc-0d74d3736a240e572

**Subnet** | [Info](#)  
No preference (Default subnet in any availability zone)

**Auto-assign public IP** | [Info](#)  
Enable

**Firewall (security groups)** | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group       Select existing security group

**Common security groups** | [Info](#)

Select security groups ▾

Primary-SG sg-002d4edfb66259799 [X](#)  
VPC: vpc-0d74d3736a240e572

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Scroll down to “Configure Storage” and make it “20GiB”

**▼ Configure storage** [Info](#)

[Advanced](#)

1x  GiB [gp3](#) Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance

Click refresh to view backup information  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

**▶ Advanced details** [Info](#)

**Storage (volumes)**  
1 volume(s) - 20 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

Then, click on “Launch Instance”

A screenshot of the AWS EC2 Instances launch success page. At the top, there is a green success message: "Success Successfully initiated launch of instance (i-0a783c8bc1d7f52ae)". Below this, there is a "Launch log" button. A red arrow points from the text "Click on 'Instances'" to the "Instances" link in the breadcrumb navigation bar. The main content area shows "Next Steps" with several options: "Create billing usage alerts", "Connect to your instance", "Connect an RDS database", "Create EBS snapshot policy", "Manage detailed monitoring", "Create Load Balancer", "Create AWS budget", and "Manage CloudWatch alarms". Each option has a corresponding "Create" or "Learn more" button.

Click on “Instances”

A screenshot of the AWS EC2 Instances list page. The left sidebar shows navigation links for EC2, Instances, Images, Elastic Block Store, and Network & Security. The main content area displays a table titled "Instances (5) Info" with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One instance, "Nexus", is shown as "Initializing". A red arrow points from the text "Wait for it to pass the ‘2/2 check’" to the "Initializing" status of the Nexus instance. The table also includes a "Find Instance by attribute or tag (case-sensitive)" search bar and buttons for "Connect", "Actions", and "Launch instances".

We have created the instance, let us wait for it to pass the “2/2 check”

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, CloudShell, Feedback, and Console Mobile App. The main area is titled "Instances (5) Info" and lists five instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0d72ba387d698a6b0	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Slave-1	i-0b025e08ef003624c	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Slave-2	i-0afb570889b99b7d9	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
SonarQube	i-072e7267a412f30e2	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Nexus	i-0a783c8bc1d7f52ae	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c

At the bottom of the main area, it says "Select an instance". The footer includes links for CloudShell, Feedback, Console Mobile App, and various AWS terms like Privacy, Terms, and Cookie preferences.

The virtual machine has passed the “**2/2 Check**”

### 3.6.2 SSH Connect to Nexus Server

Let us create a duplicate of the session “**Master**”.

The screenshot shows the MobaXterm application interface. On the left, there's a sidebar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. The main window shows several terminal sessions:

- Session 16: Master (selected)
- Session 17: Slave-1
- Session 18: Slave-2
- Session 19: SonarQube

The terminal window for session 16 (Master) displays the following content:

```

        • MobaXterm Personal Edition v24.1 •
        (SSH client, X server and network tools)

    SSH session to ubuntu@172.31.25.143
    • Direct SSH : ✓
    • SSH compression : ✓
    • SSH-browser : ✓
    • X11-forwarding : ✓ (remote display is forwarded through SSH)

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

Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Sat Jan 3 19:30:22 UTC 2026
  System load: 0.0          Processes: 123
  Usage of /: 9.4% of 18.33GB  Users logged in: 0
  Memory usage: 6%           IPv4 address for enX0: 172.31.25.143
  Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

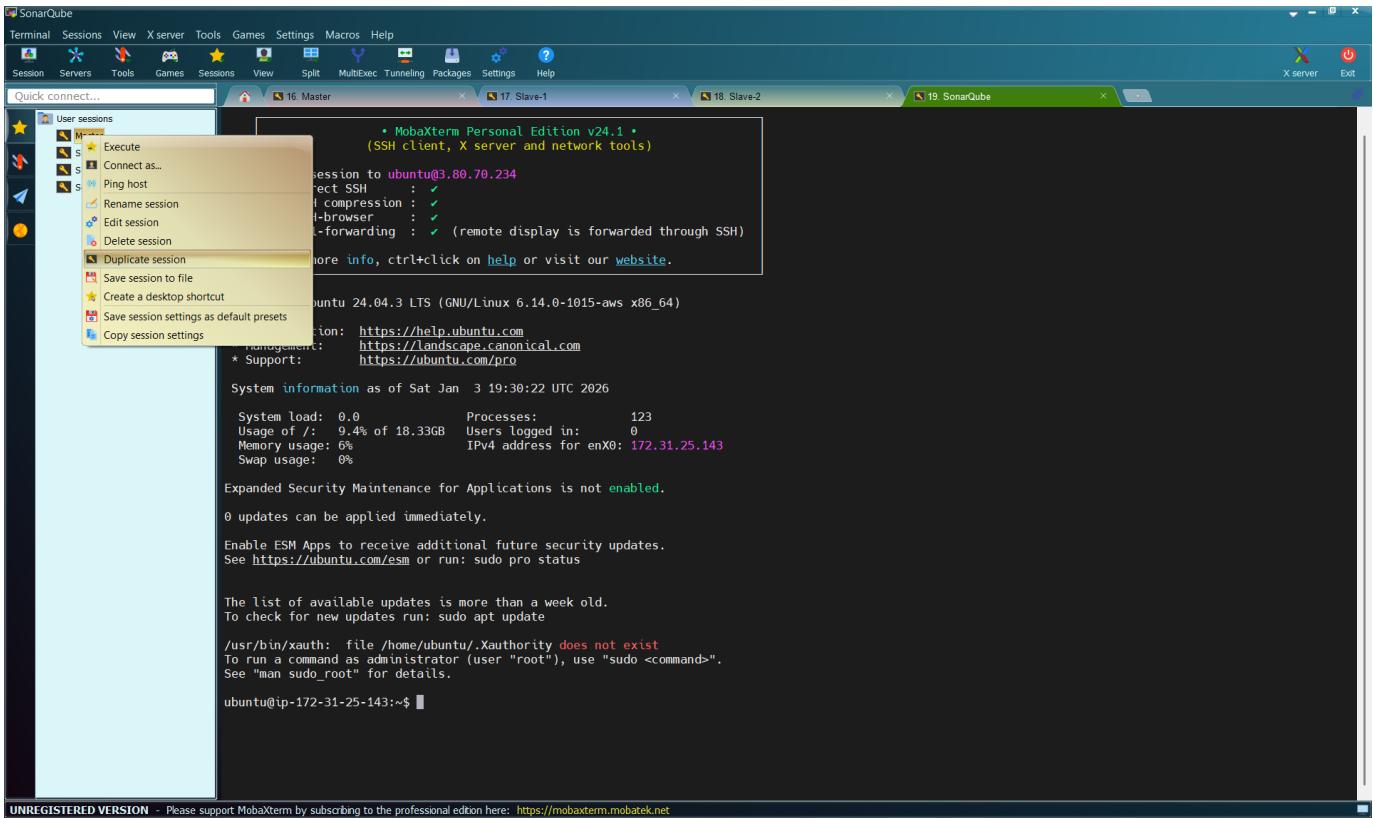
/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-25-143:~$ 

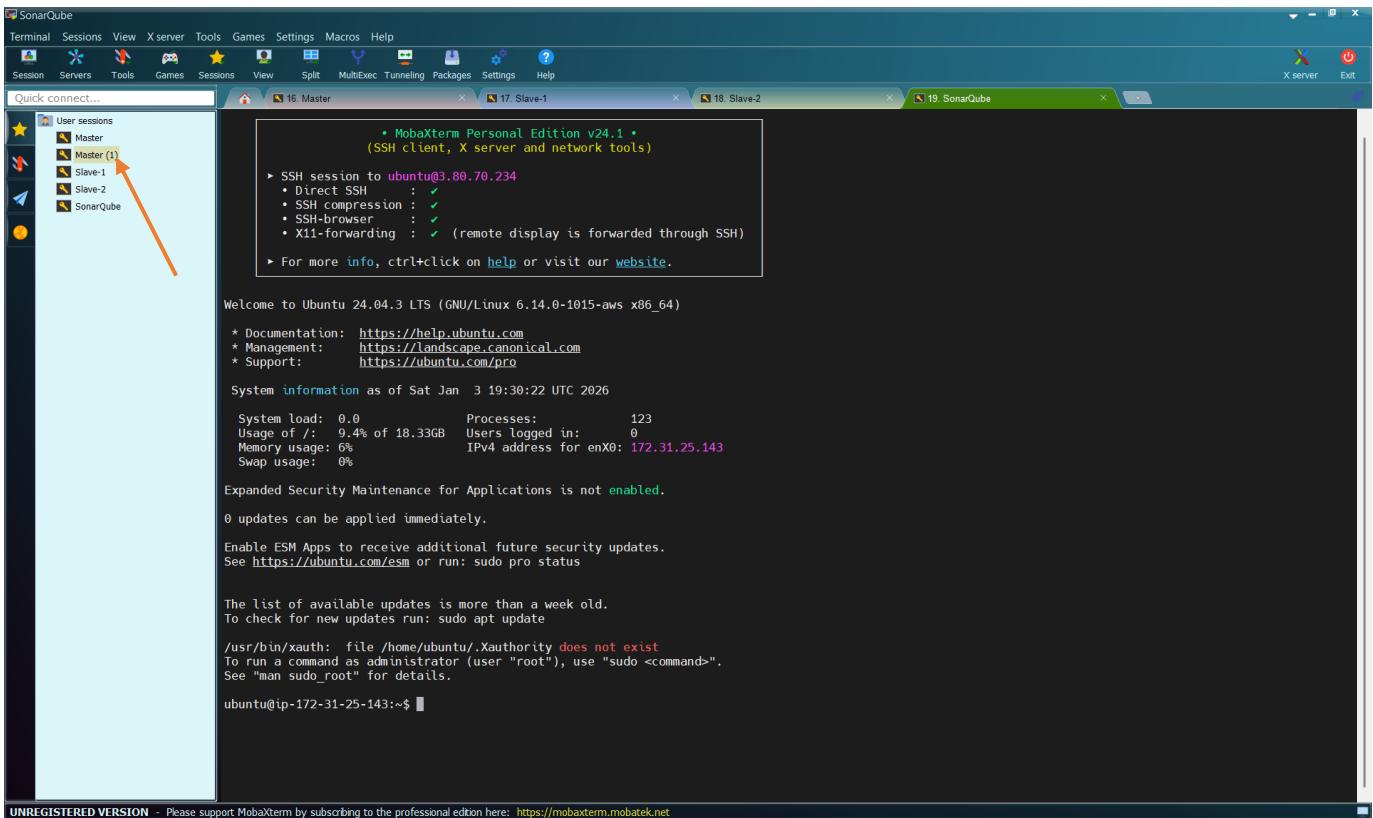
```

At the bottom of the terminal window, a message reads: "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>".

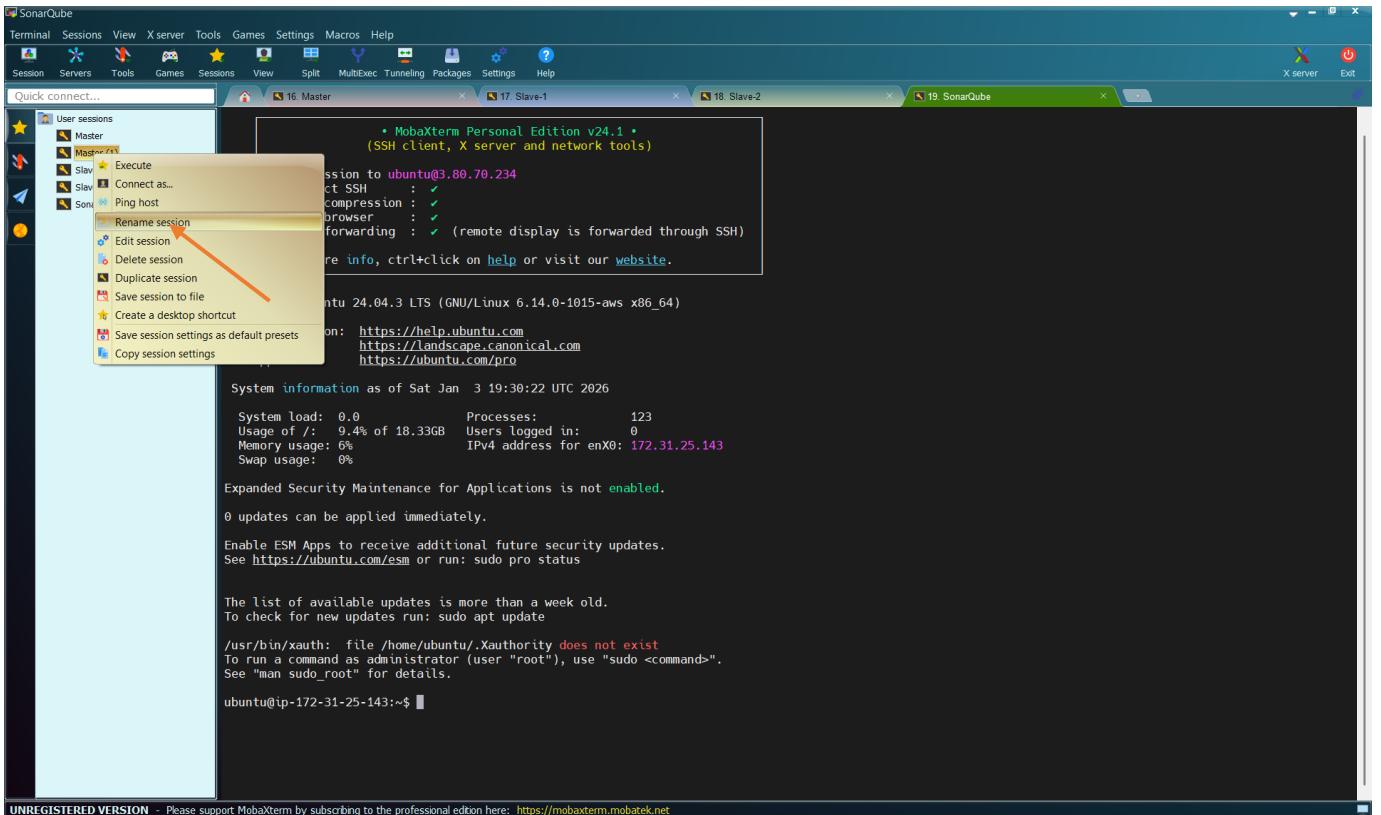
Right-click on the session name “**Master**”



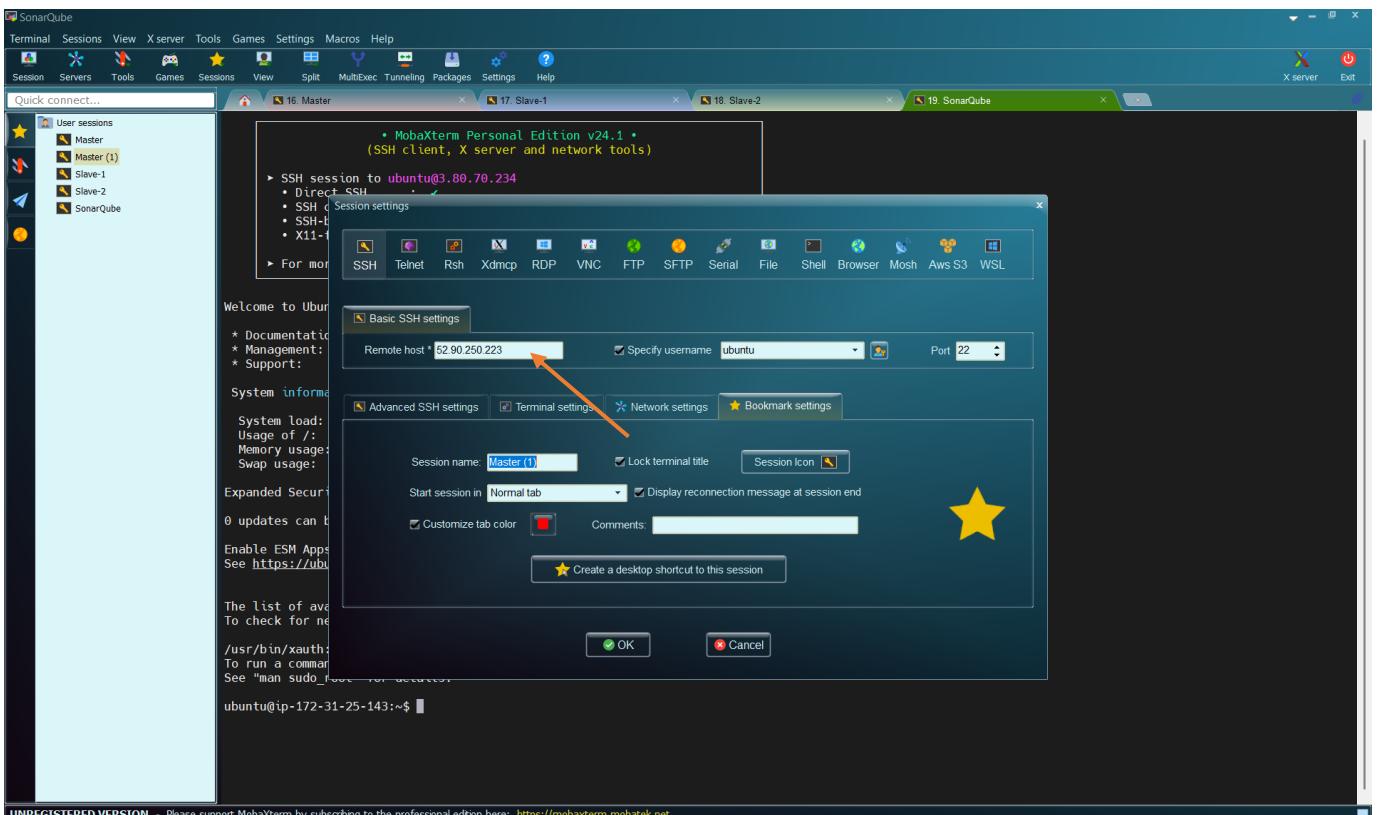
## Select “Duplicate Session”



We want to rename “Master(1)” to “Nexus”. Right-click on “Master(1)”



## Select “Rename Session”



Copy the Public IP address of our “Nexus” virtual machine and paste here.

Instances (1/5) **Nexus**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0d72ba387d698a6b0	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Slave-1	i-0b025e08ef003624c	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
Slave-2	i-0afb570889b99b7d9	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
SonarQube	i-072e7267a412f30e2	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c
<b>Nexus</b>	i-0a783c8bc1d7f52ae	Running	t2.medium	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c

**i-0a783c8bc1d7f52ae (Nexus)**

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

**Instance summary**

Instance ID: i-0a783c8bc1d7f52ae

Public IPv4 address: 52.55.215.200 | [open address](#)

Private IP addresses: 172.31.23.72

Public DNS: ec2-52-55-215-200.compute-1.amazonaws.com | [open address](#)

IPv6 address: -

Instance state: Running

Hostname type: IP name: ip-172-31-23-72.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-23-72.ec2.internal

Answer private resource DNS name: IPv4 (A)

Instance type: t2.medium

Auto-assigned IP address: 172.31.23.72

VPC ID: vpc-00000000

Elastic IP addresses: -

AWS Compute Optimizer finding: -

Copy the Public IP: **52.55.215.200** and paste the MobaXterm

SonarQube Terminal Sessions View X server Tools Games Settings Macros Help Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect... User sessions Master (1) Slave-1 Slave-2 SonarQube

16. Master 17. Slave-1 18. Slave-2 19. SonarQube

MobaXterm Personal Edition v24.1 (SSH client, X server and network tools)

SSH session to ubuntu@3.80.70.234

- Direct SSH
- SSH connection settings
- SSH-1
- X11-1

Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-102-generic x86\_64)

System information

Session name: Master (1)

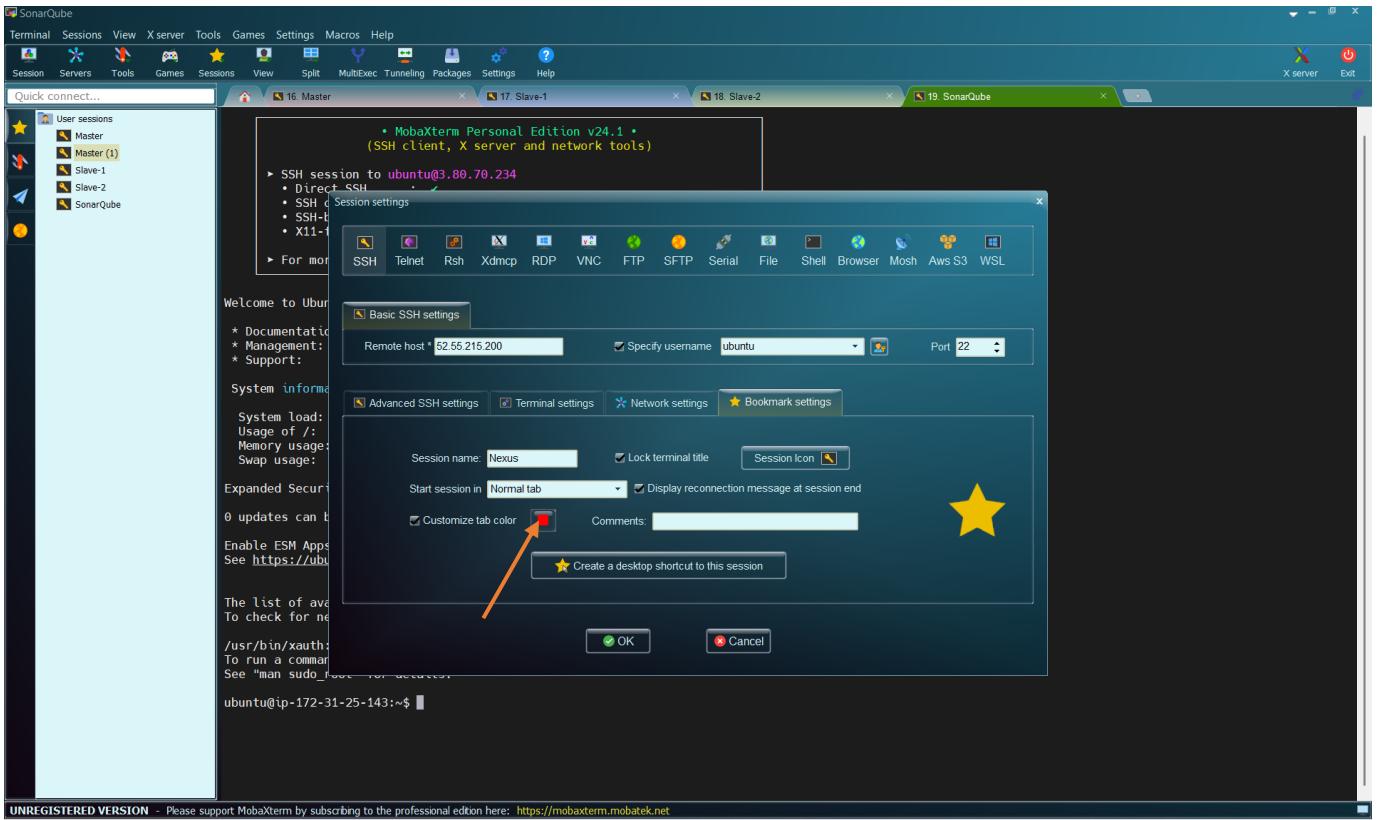
Start session in: Normal tab

Comments:

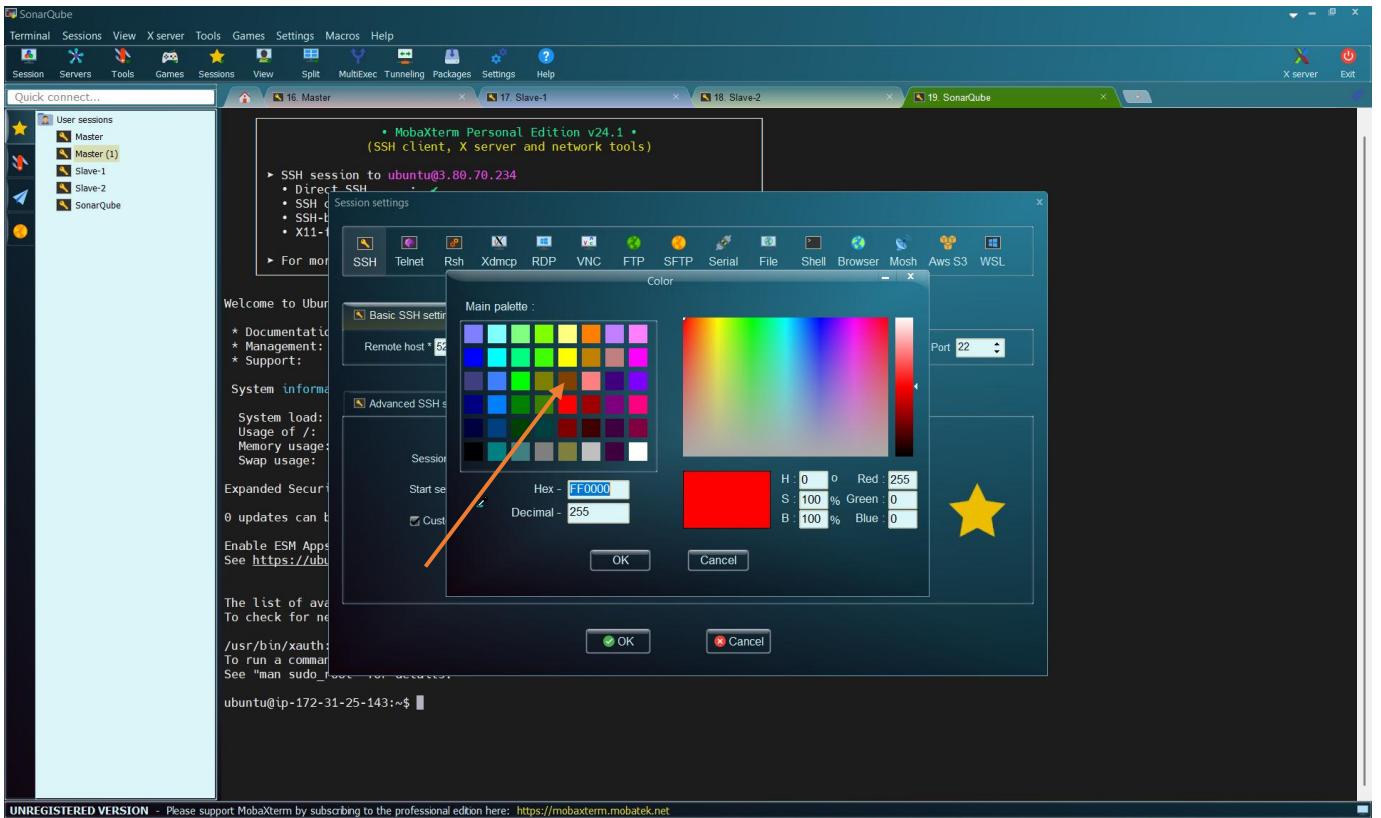
OK Cancel

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

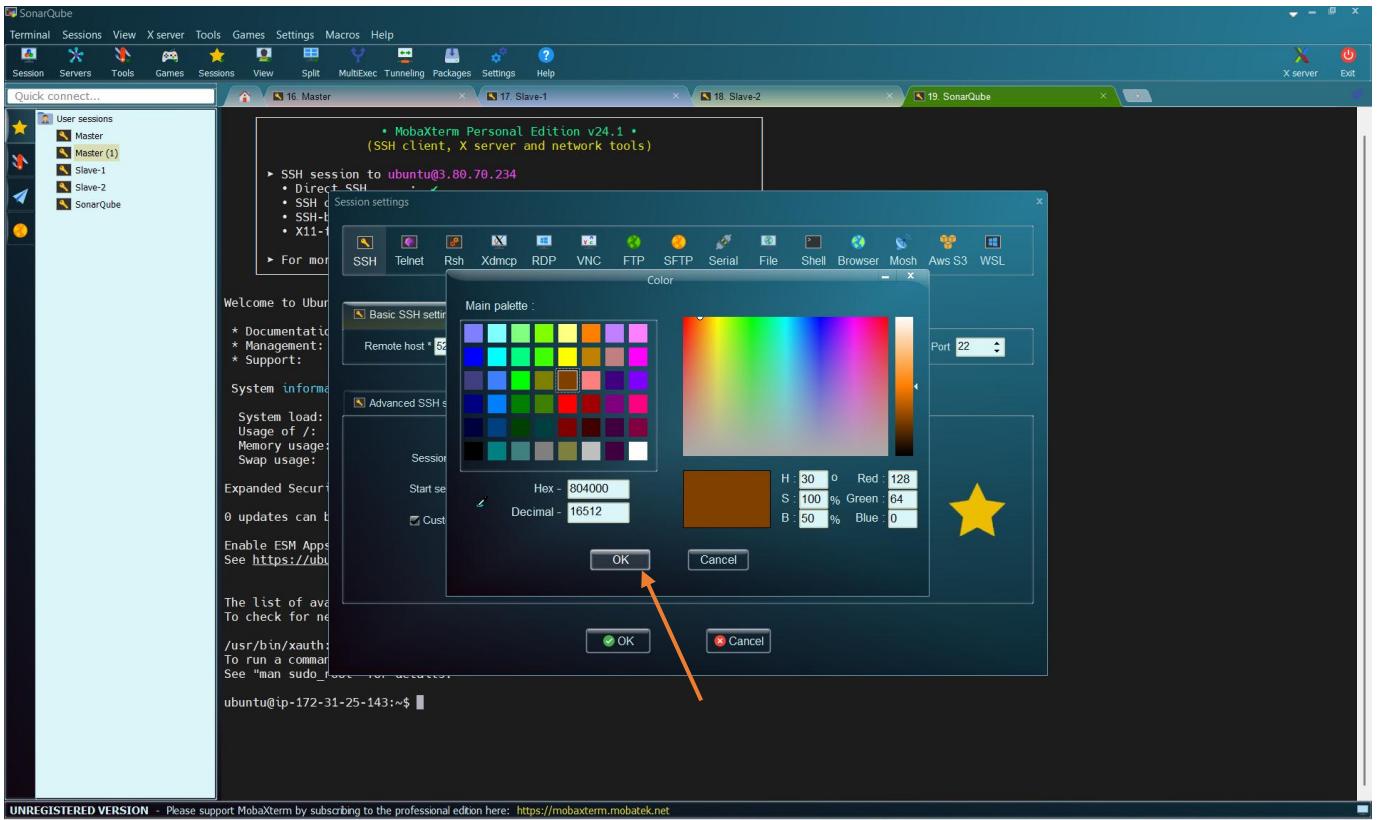
Then, change the name to “**Nexus**”



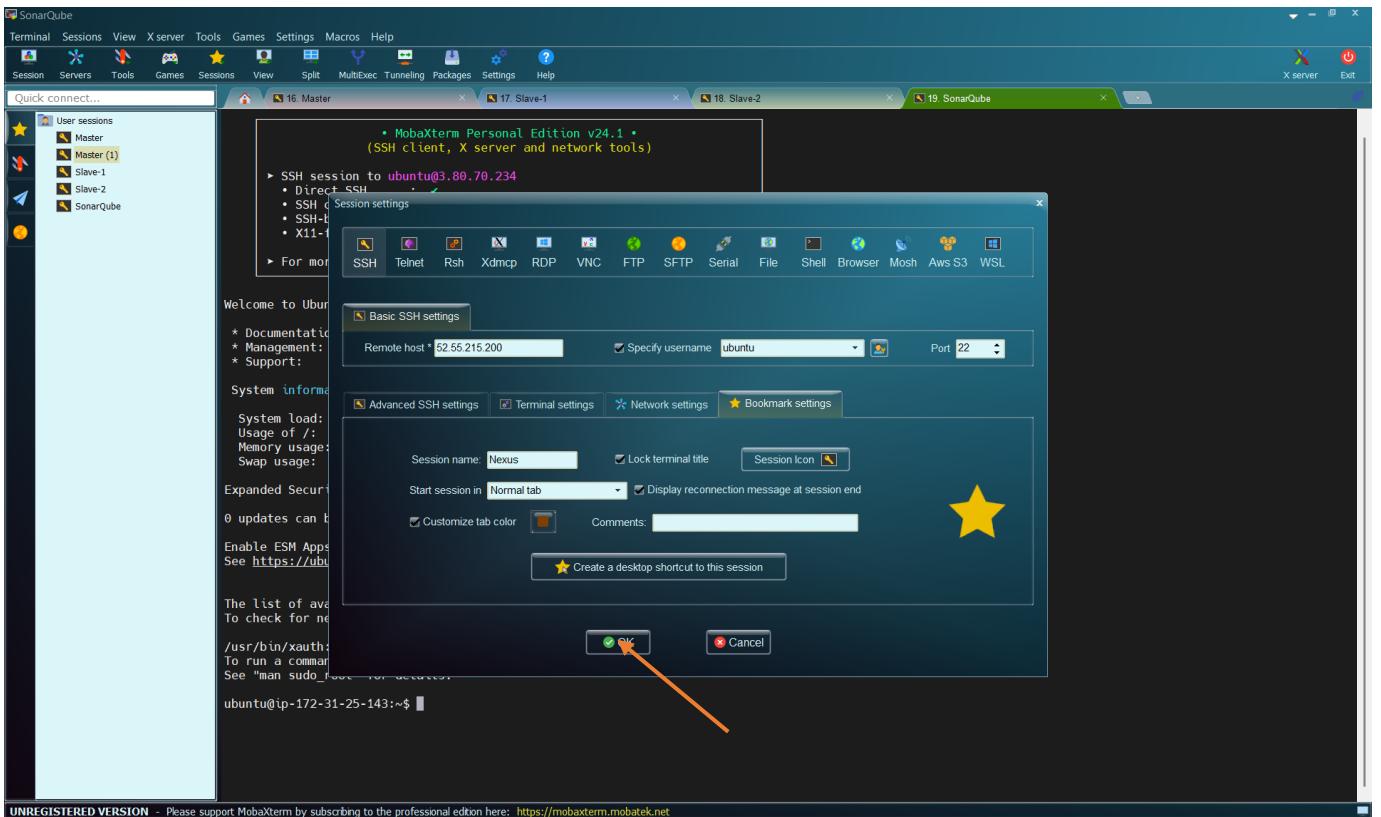
Then, let us change the color. Click there



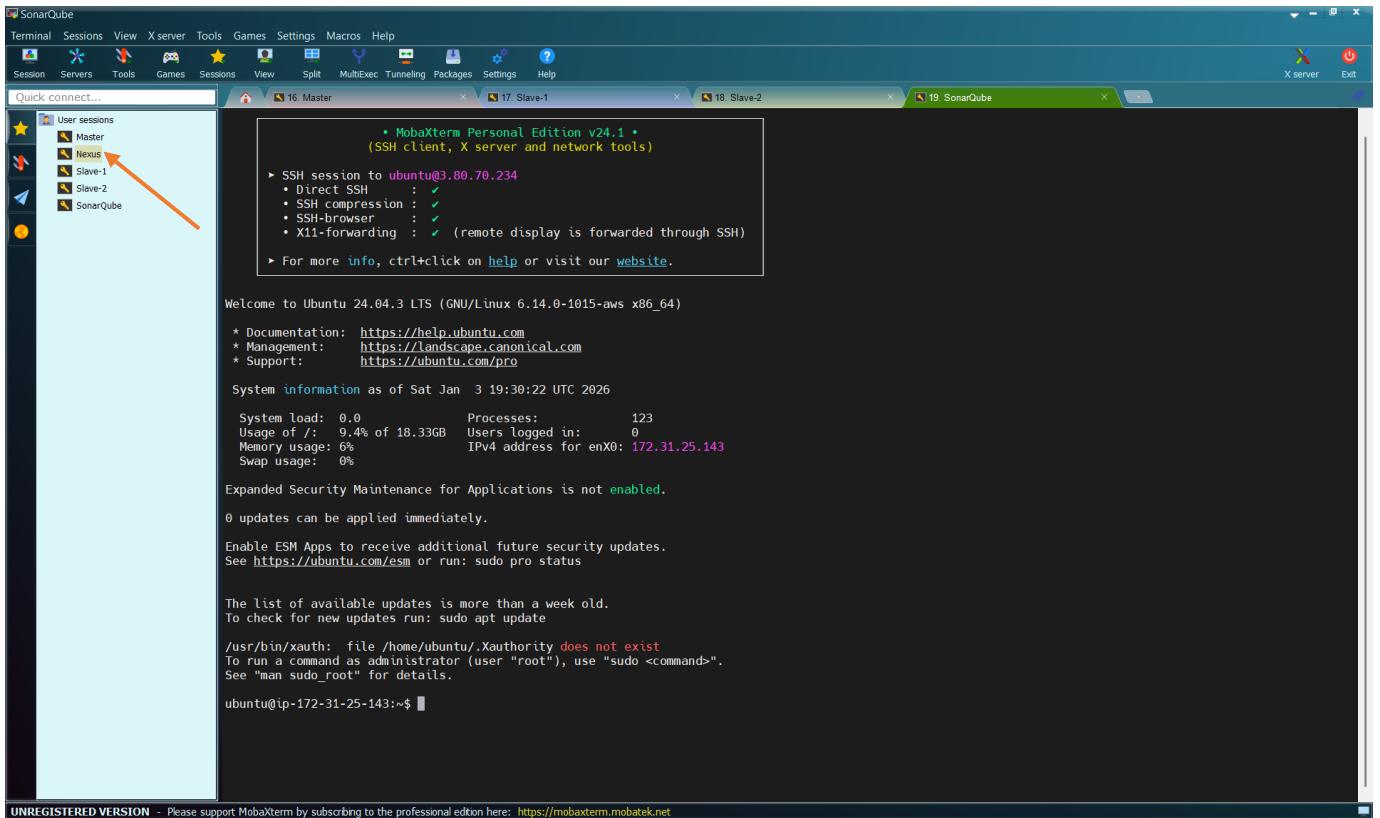
Let us select "Brown"



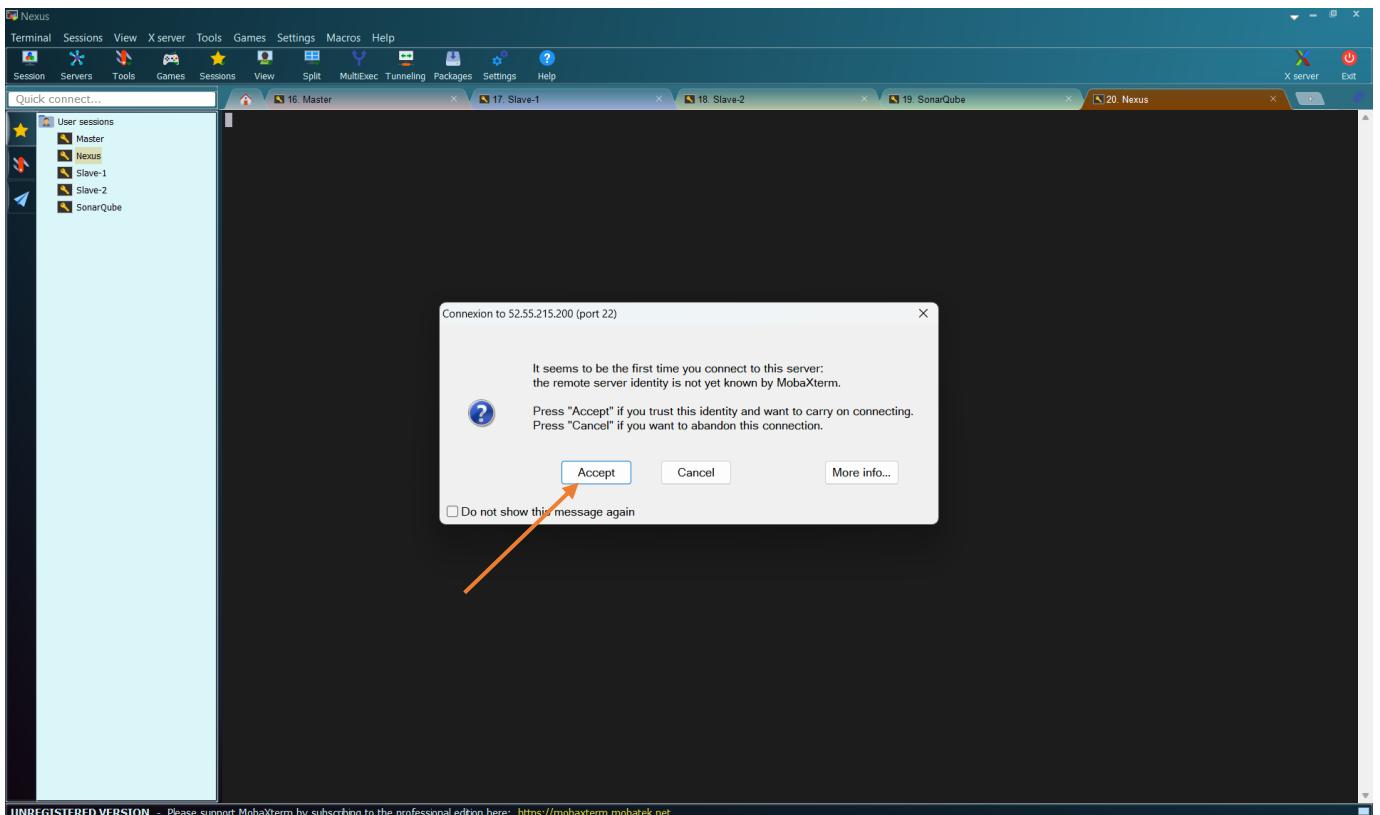
Click on “OK”



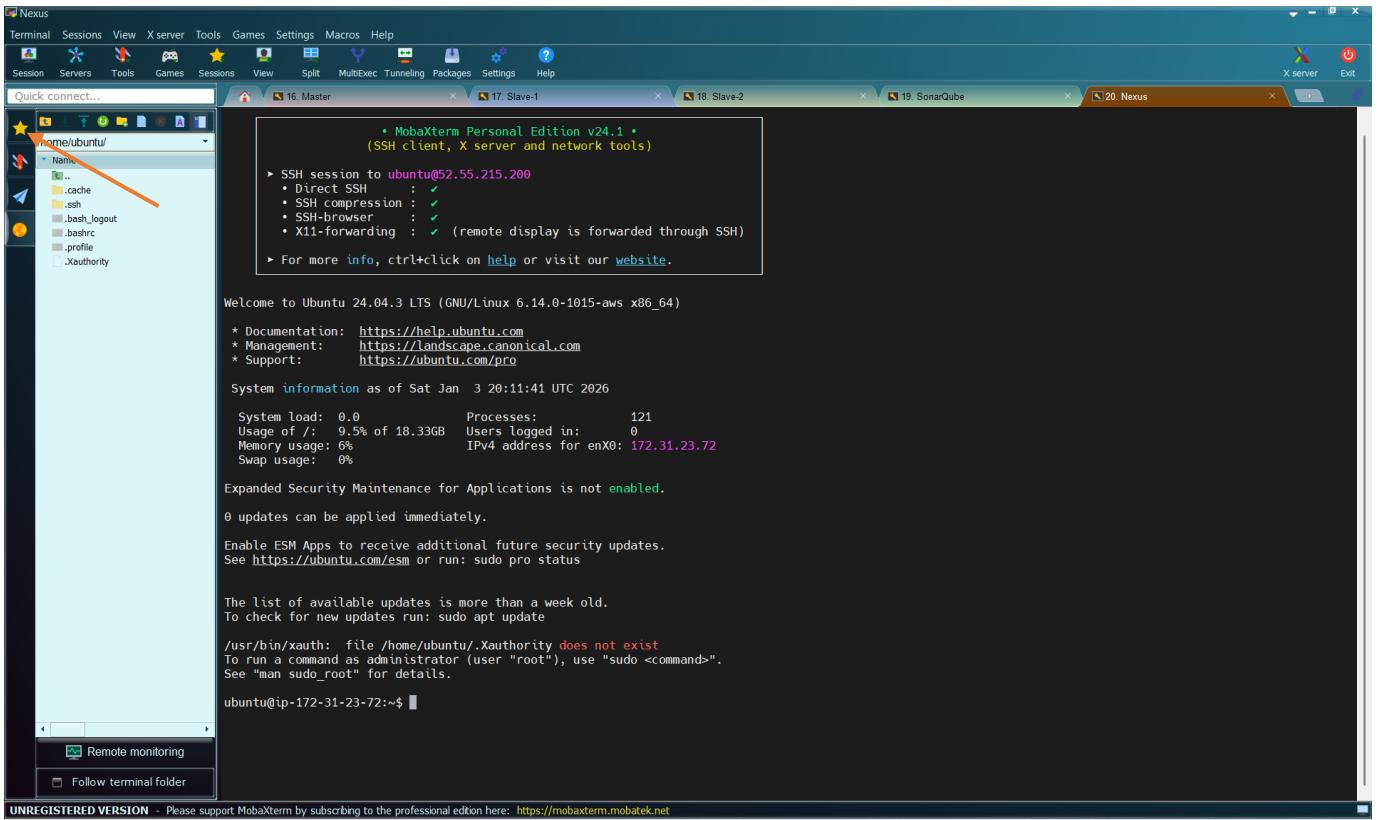
And click on “OK” again



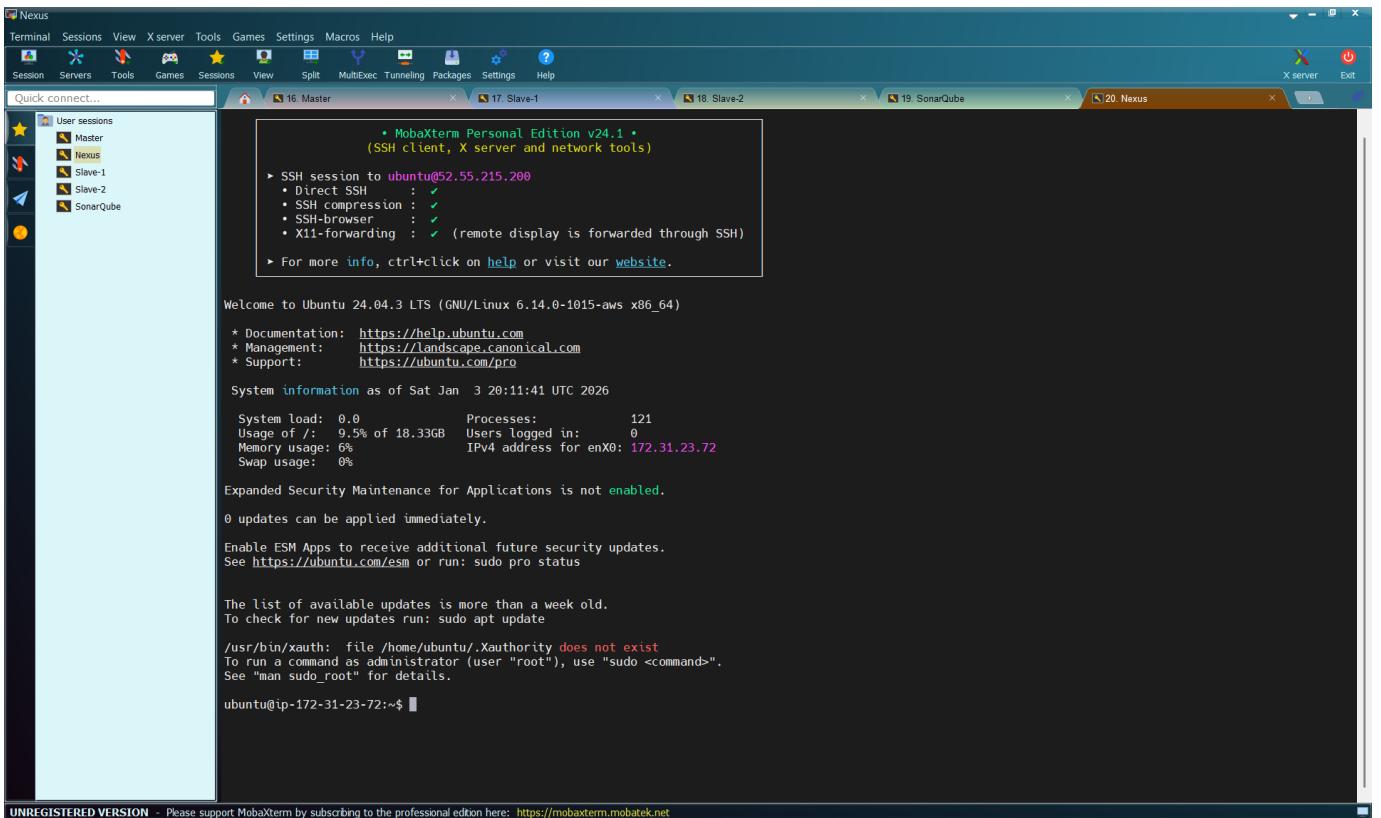
Then, double-click on the session “Nexus”



Click on “Accept”



You can see that the tab of “Nexus” is “Brown”. Click on the star

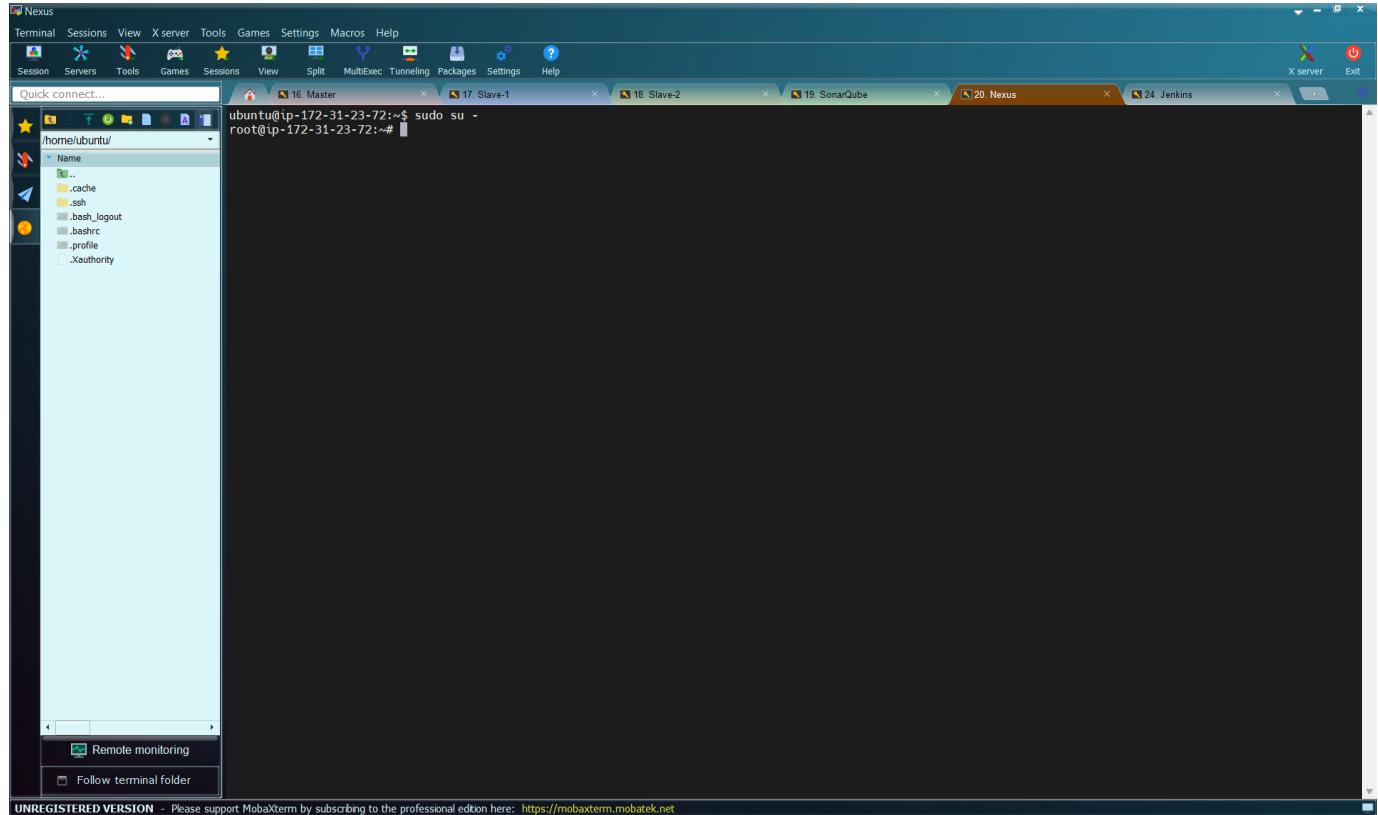


We are now connected to the Nexus server.

### 3.6.3 Configure Nexus Server

We will now configure the Nexus server. Let us first grant it root user privilege by running the command:

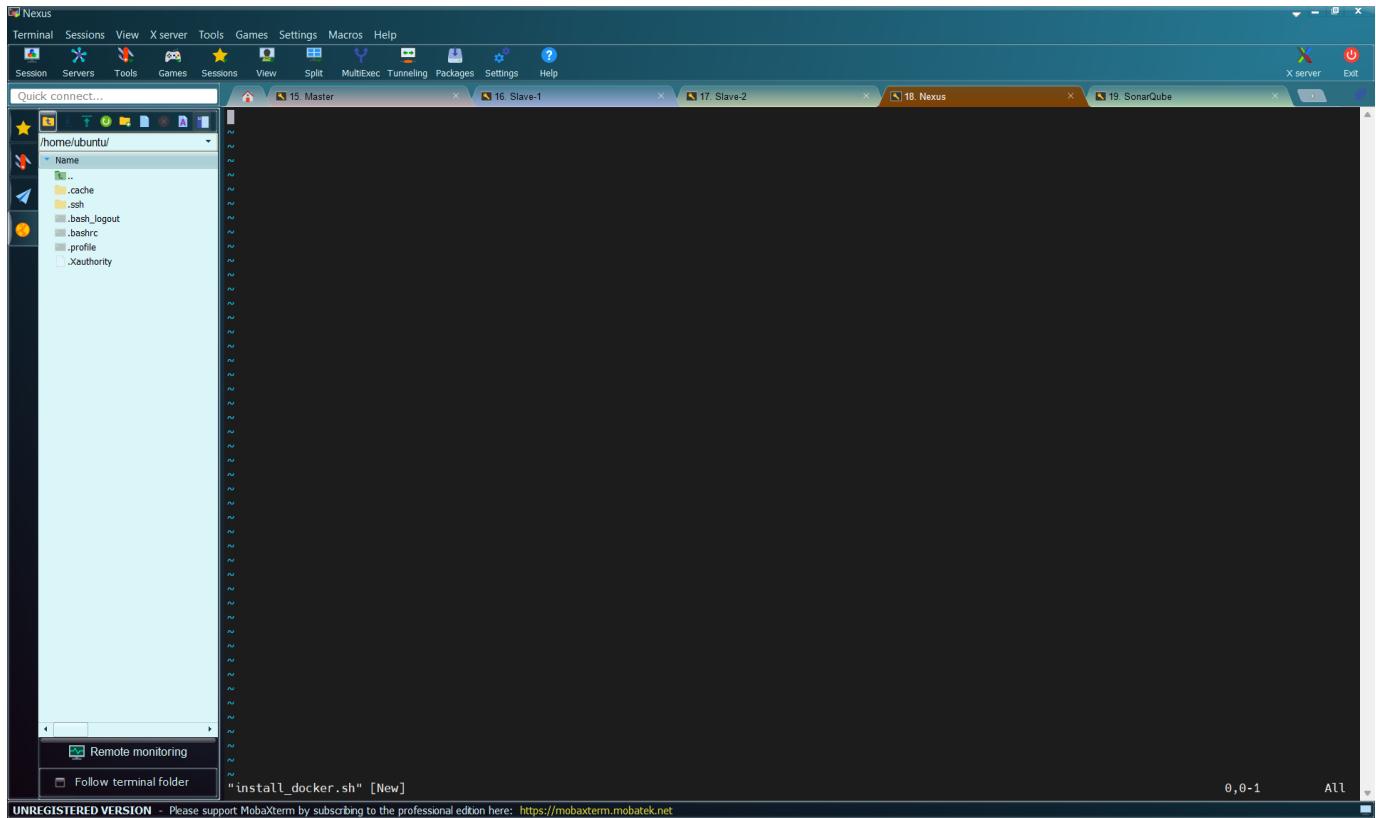
```
sudo su -
```



Next, we have to install Docker on the Nexus server. We will use a shell script called “docker-install.sh” that will first update and upgrade the packages before installing Docker.

First create the shell script using the command:

```
vi install_docker.sh
```



### Paste the code to install Docker Package

```
#!/bin/bash

# Update package manager repositories
sudo apt-get update

# Install necessary dependencies
sudo apt-get install -y ca-certificates curl

# Create directory for Docker GPG key
sudo install -m 0755 -d /etc/apt/keyrings

# Download Docker's GPG key
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

# Ensure proper permissions for the key
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add Docker repository to Apt sources
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Update package manager repositories
sudo apt-get update

sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```
#!/bin/bash
# Update package manager repositories
sudo apt-get update
# Install necessary dependencies
sudo apt-get install -y ca-certificates curl
# Create directory for Docker GPG key
sudo install -m 0755 -d /etc/apt/keyrings
# Download Docker's GPG key
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
# Ensure proper permissions for the key
sudo chmod a+r /etc/apt/keyrings/docker.asc
# Add Docker repository to Apt sources
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$(lsb_release -c -s) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
# Update package manager repositories
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

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

Save and Exit the file by pressing “**ESC**”, followed by “**:wq**” and then press “**Enter**”

```
root@ip-172-31-22-8:~# vi install_docker.sh
```

Make the script executable using the command:

```
chmod +x install_docker.sh
```

```

Nexus
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15 Master 16 Slave-1 17 Slave-2 18 Nexus 19 SonarCube X server Exit
root@ip-172-31-22-8:~# vi install_docker.sh
root@ip-172-31-22-8:~# chmod +x install_docker.sh
root@ip-172-31-22-8:~#

```

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

Then, run the file using the command:

```
./install_docker.sh
```

```

Nexus
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
15 Master 16 Slave-1 17 Slave-2 18 Nexus 19 SonarCube X server Exit
Selecting previously unselected package docker-ce.
Preparing to unpack .../2-docker-ce_5%3a29.1.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package pigz.
Preparing to unpack .../3-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../4-docker-buildx-plugin_0.30.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a29.1.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_5.0.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../7-libslirp0_4.7.0-0ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-0ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../8-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (2.2.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (2.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Setting up docker-ce-clients (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-0ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-22-8:~#

```

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

Docker has been installed but at the moment other users cannot access it. Only the root user can access it. In order to grant access to other users, we have to run the command:

```
sudo chmod 666 /var/run/docker.sock
```

The screenshot shows a MobaXterm interface with multiple windows open. The active terminal window displays the command `sudo chmod 666 /var/run/docker.sock` being run. The output of the command is shown below:

```
Preparing to unpack .../2-docker-ce_5%3a20.3.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package pigz ...
Preparing to unpack .../3-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package docker-buildx-plugin ...
Preparing to unpack .../4-docker-buildx-plugin_0.30.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras ...
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a29.1.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin ...
Preparing to unpack .../6-docker-compose-plugin_5.0.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libslirp0:amd64 ...
Preparing to unpack .../7-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns ...
Preparing to unpack .../8-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Setting up containerd (1.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Setting up docker-ce-cll (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

At the bottom of the terminal window, the command `root@ip-172-31-22-8:# sudo chmod 666 /var/run/docker.sock` is shown again, indicating it was run in the previous step.

Let us run the command to verify that Docker is running:

```
sudo systemctl status docker
```

The screenshot shows a MobaXterm interface with multiple windows open. The active terminal window displays the command `sudo systemctl status docker` being run. The output of the command is shown below:

```
Setting up docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Setting up docker-ce-cll (5:29.1.3-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:29.1.3-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:29.1.3-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

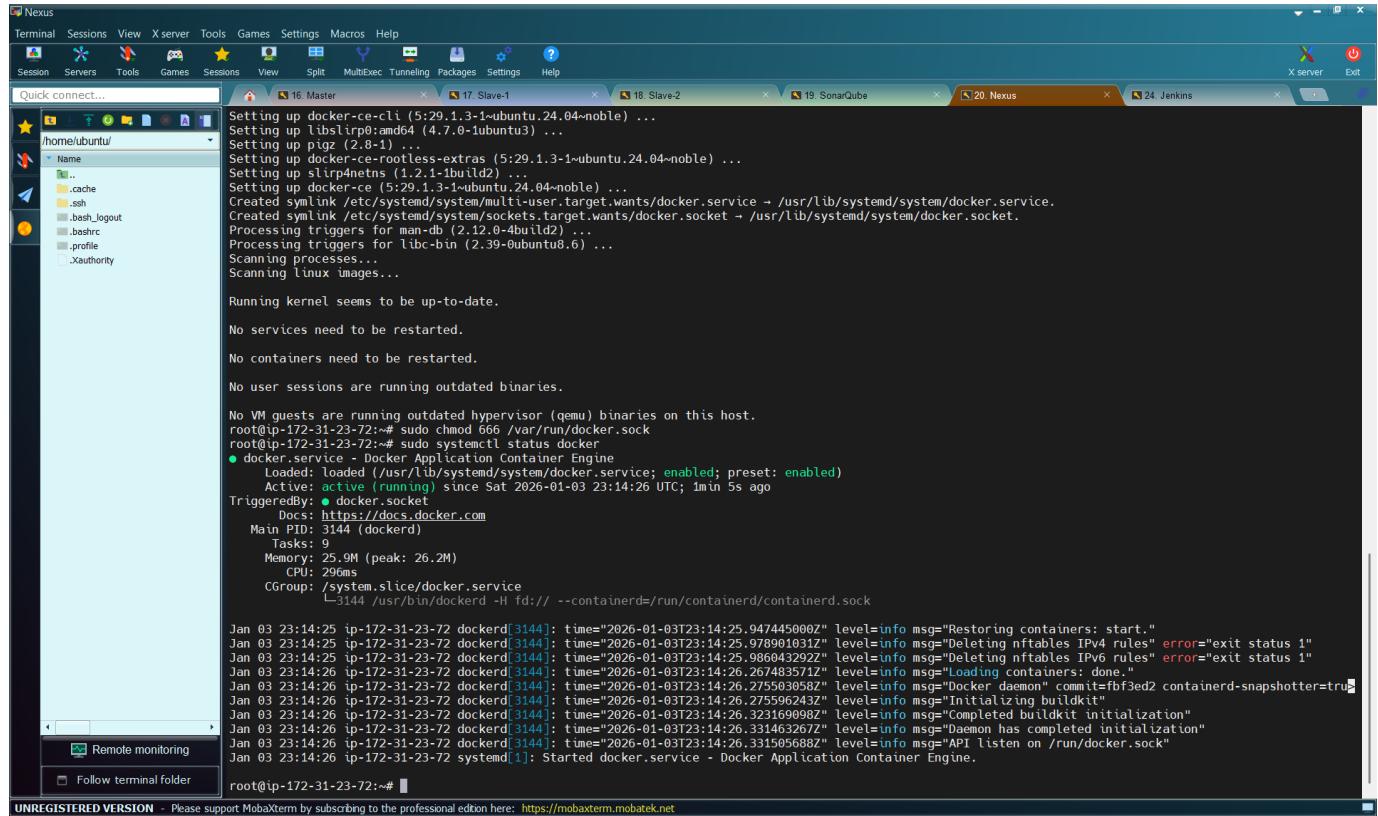
The output then continues with the results of the `systemctl status docker` command:

```
root@ip-172-31-22-8:# sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Sat 2026-01-03 23:14:26 UTC; 1min 5s ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
 Main PID: 3144 (dockerd)
   Tasks: 9
  Memory: 25.9M (peak: 26.2M)
    CPU: 296ms
   CGroup: /system.slice/docker.service
           └─3144 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jan 03 23:14:25 ip-172-31-22-7 dockerd[3144]: time="2026-01-03T23:14:25.947445000Z" level=info msg="Restoring containers: start."
Jan 03 23:14:25 ip-172-31-22-7 dockerd[3144]: time="2026-01-03T23:14:25.978901031Z" level=info msg="Deleting nftables IPv4 rules" error="exit status 1"
Jan 03 23:14:25 ip-172-31-22-7 dockerd[3144]: time="2026-01-03T23:14:25.986643292Z" level=info msg="Deleting nftables IPv6 rules" error="exit status 1"
Jan 03 23:14:26 ip-172-31-22-7 dockerd[3144]: time="2026-01-03T23:14:26.267483571Z" level=info msg="Loading containers: done."
Jan 03 23:14:26 ip-172-31-22-7 dockerd[3144]: time="2026-01-03T23:14:26.275503058Z" level=info msg="Docker daemon" commit=fbfbfed2 containerd-snapshotter=true
Jan 03 23:14:26 ip-172-31-22-7 dockerd[3144]: time="2026-01-03T23:14:26.275596243Z" level=info msg="Initializing buildkit"
Jan 03 23:14:26 ip-172-31-22-7 dockerd[3144]: time="2026-01-03T23:14:26.32316998Z" level=info msg="Completed buildkit initialization"
Jan 03 23:14:26 ip-172-31-22-7 dockerd[3144]: time="2026-01-03T23:14:26.331463267Z" level=info msg="Daemon has completed initialization"
Jan 03 23:14:26 ip-172-31-22-7 dockerd[3144]: time="2026-01-03T23:14:26.33150568Z" level=info msg="API listen on /run/docker.sock"
Jan 03 23:14:26 ip-172-31-22-7 systemd[1]: Started docker.service - Docker Application Container Engine.
[tues 1-22/22 (END)]
```

You can see that Docker is Active and running. Exit and go back to command mode by using

Ctrl+c



```
Setting up docker-ce-cli (5:29.1.3-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up slirpnetns (1.2.1-1build2) ...
Setting up docker-ce (5:29.1.3-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/socket.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-23-72:~# sudo chmod 666 /var/run/docker.sock
root@ip-172-31-23-72:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
  Active: active (running) since Sat 2026-01-03 23:14:26 UTC; 1min 5s ago
TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 3144 (dockerd)
     Tasks: 9
    Memory: 25.9M (peak: 26.2M)
      CPU: 296ms
     CGroup: /system.slice/docker.service
             └─3144 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jan 03 23:14:25 ip-172-31-23-72 dockerd[3144]: t[time="2026-01-03T23:14:25, 9474458006" level=info msg="Restoring containers: start"]
Jan 03 23:14:25 ip-172-31-23-72 dockerd[3144]: t[time="2026-01-03T23:14:25, 9789618317" level=info msg="Deleting nftables IP4 rules" error="exit status 1"]
Jan 03 23:14:25 ip-172-31-23-72 dockerd[3144]: t[time="2026-01-03T23:14:25, 9860432927" level=info msg="Deleting nftables IP6 rules" error="exit status 1"]
Jan 03 23:14:26 ip-172-31-23-72 dockerd[3144]: t[time="2026-01-03T23:14:26, 2674835712" level=info msg="Loading containers: done."]
Jan 03 23:14:26 ip-172-31-23-72 dockerd[3144]: t[time="2026-01-03T23:14:26, 2755630582" level=info msg="Docker daemon" commit=fbf3ed2 containerd-snapshotter=true]
Jan 03 23:14:26 ip-172-31-23-72 dockerd[3144]: t[time="2026-01-03T23:14:26, 2755624327" level=info msg="Initializing buildkit"]
Jan 03 23:14:26 ip-172-31-23-72 dockerd[3144]: t[time="2026-01-03T23:14:26, 3231699982" level=info msg="Completed buildkit initialization"]
Jan 03 23:14:26 ip-172-31-23-72 dockerd[3144]: t[time="2026-01-03T23:14:26, 3314632672" level=info msg="Daemon has completed initialization"]
Jan 03 23:14:26 ip-172-31-23-72 dockerd[3144]: t[time="2026-01-03T23:14:26, 3315056882" level=info msg="API listen on /run/docker.sock"]
Jan 03 23:14:26 ip-172-31-23-72 systemd[1]: Started docker.service - Docker Application Container Engine.

root@ip-172-31-23-72:~#
```

Let us create a Docker container running on the Nexus server. To create a Docker container running Nexus 3 and exposing it on port 8081, you can use the following command:

```
docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
```

This command does the following:

- **-d:** Detaches the container and runs it in the background.
- **--name nexus:** Specifies the name of the container as "nexus".
- **-p 8081:8081:** Maps port 8081 on the host to port 8081 on the container, allowing access to Nexus through port 8081.
- **sonatype/nexus3:latest:** Specifies the Docker image to use for the container, in this case, the latest version of Nexus 3 from the Sonatype repository.

After running this command, Nexus will be accessible on your host machine at

```
http://<Public IP address of Nexus Server>:8081
```

```
root@ip-172-31-22-8:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3:latest
7db948e960b2: Pull complete
c5bca7a30871: Pull complete
09cada718aca: Pull complete
ed8e6720939: Pull complete
46a9484471e5: Pull complete
25220a69af22: Pull complete
70f45f825dd3: Download complete
Digest: sha256:0ab4c84823747aaad31a1a35b4f646f28970e33b0b9da8014b19bcd5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
e9593c8babae3bf2415bf35e4f5115d82819e00ed14d925210ab5caf71f9a7ef
root@ip-172-31-22-8:~#
```

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

The container has been created.

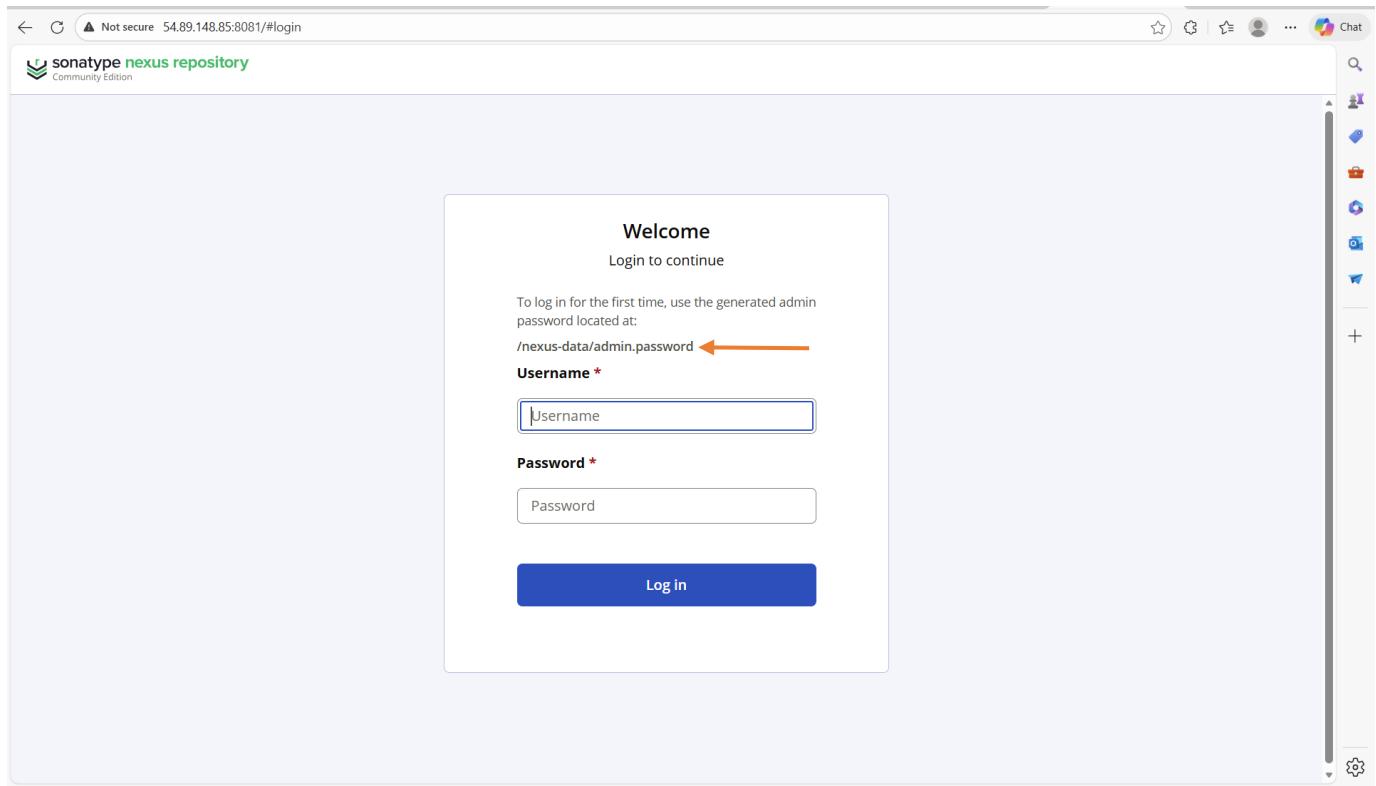
### 3.6.4 Accessing Nexus through Browser

Let us access Nexus through the browser by using the Public IP address of the Nexus server. This is done as follows:

`http://<Public IP address of Nexus Server>:8081`

That is

`http://54.89.148.85:8081`



Enter the default username is “**admin**”, but we have to get the password by accessing the path above. That is `/nexus-data/admin.password`

### Get Nexus initial password

Your provided commands are correct for accessing the Nexus password stored in the container. Here's a breakdown of the steps:

**Get Container ID:** You need to find out the ID of the Nexus container. You can do this by running this command on Nexus server:

```
docker ps
```

This command lists all running containers along with their IDs, among other information.

```

root@ip-172-31-22-8:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e960b2: Pull complete
c5bc7a30871: Pull complete
09ada718aca: Pull complete
ed8e6720939: Pull complete
46a9484471e5: Pull complete
25220a69af22: Pull complete
70fa5f025dd3: Download complete
Digest: sha256:0ab4c84823747aad31a1a35b4f646f28970e33b0b9da8014b19bcd5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
e9593c8abae3b4f24150f35e4f5115d82819e00ed14d925210ab5caf71f9a7ef
root@ip-172-31-22-8:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e9593c8abae3 sonatype/nexus3 "/opt/sonatype/nexus..." 5 minutes ago Up 5 minutes 0.0.0.0:8081->8081/tcp, [::]:8081->8081/tcp Nexus
root@ip-172-31-22-8:~#

```

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

We can see the Docker image is running with the name “**Nexus**”, with container ID “**f1d618e32968**”. We will use this in the next command to access the container’s Bash Shell.

**Access Container’s Bash Shell:** Once you have the container ID, you can execute the docker exec command to access the container’s bash shell:

```
docker exec -it <container_ID> /bin/bash
```

Replace <container\_ID> with the actual ID of the Nexus container.

```
docker exec -it e9593c8abae3 /bin/bash
```

```

root@ip-172-31-22-8:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e960b2: Pull complete
c5bc7a30871: Pull complete
99aca718acaa: Pull complete
ed9e6720939: Pull complete
46a9484471e5: Pull complete
25220a9af22: Pull complete
70fa5f8f25dd3: Download complete
Digest: sha256:0a84c84823747aad31a1a35b4f646f28970e33b0b9da8014b19bcd5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
e9593c8abae3b4f24150f35e4f5115d82819e0e0ed14d925210ab5caf71f9a7ef
root@ip-172-31-22-8:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e9593c8abae3 sonatype/nexus3 "/opt/sonatype/nexus..." 5 minutes ago Up 5 minutes 0.0.0.0:8081->8081/tcp, [::]:8081->8081/tcp Nexus
root@ip-172-31-22-8:~# ^C
root@ip-172-31-22-8:~# docker exec -it e9593c8abae3 /bin/bash
bash-5.1$ 

```

Then, run the command:

`ls`

```

root@ip-172-31-22-8:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e960b2: Pull complete
c5bc7a30871: Pull complete
99aca718acaa: Pull complete
ed9e6720939: Pull complete
46a9484471e5: Pull complete
25220a9af22: Pull complete
70fa5f8f25dd3: Download complete
Digest: sha256:0a84c84823747aad31a1a35b4f646f28970e33b0b9da8014b19bcd5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
e9593c8abae3b4f24150f35e4f5115d82819e0e0ed14d925210ab5caf71f9a7ef
root@ip-172-31-22-8:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e9593c8abae3 sonatype/nexus3 "/opt/sonatype/nexus..." 5 minutes ago Up 5 minutes 0.0.0.0:8081->8081/tcp, [::]:8081->8081/tcp Nexus
root@ip-172-31-22-8:~# ^C
root@ip-172-31-22-8:~# docker exec -it e9593c8abae3 /bin/bash
bash-5.1$ ls
nexus sonatype-work start-nexus-repository-manager.sh
bash-5.1$ 

```

**Navigate to Nexus Directory:** Inside the container's bash shell, navigate to the directory where Nexus stores its configuration:

```
cd sonatype-work/
```

```
root@ip-172-31-22-8:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e960b2: Pull complete
c5bca7a30871: Pull complete
09cad718aca: Pull complete
ed8e06720939: Pull complete
46a9484471e5: Pull complete
25220a69af22: Pull complete
70fa5f825dd3: Download complete
Digest: sha256:0ab4c84823747aa31a1a35b4f646f28970e33b0b9da8014b19bcd5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
e9593c8abae3: Pulling from sonatype/nexus3
" /opt/sonatype/nexus3" 5 minutes ago Up 5 minutes 0.0.0.0:8081->8081/tcp, [::]:8081->8081/tcp NAMES
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e9593c8abae3 sonatype/nexus3 "/opt/sonatype/nexus3" 5 minutes ago Up 5 minutes 0.0.0.0:8081->8081/tcp, [::]:8081->8081/tcp Nexus
root@ip-172-31-22-8:~# ^C
root@ip-172-31-22-8:~# docker exec -it e9593c8abae3 /bin/bash
bash-5.1$ ls
nexus sonatype-work start-nexus-repository-manager.sh
bash-5.1$ cd sonatype-work/
bash-5.1$ nexus sonatype-work start-nexus-repository-manager.sh
bash-5.1$ bash-5.1$
```

Then, run the command:

```
ls
```

```
root@ip-172-31-23-72:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e960b2: Pull complete
c5bca7a30871: Pull complete
46a9484471e5: Pull complete
25220a69af22: Pull complete
ed8e06720939: Pull complete
09cad718aca: Pull complete
70fa5f825dd3: Download complete
Digest: sha256:0ab4c84823747aa31a1a35b4f646f28970e33b0b9da8014b19bcd5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
f1d618e32968e8f04f213cd3fc170538e5d37a1c1ac68a33d5b1799e8e020b4
root@ip-172-31-23-72:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f1d618e32968 sonatype/nexus3 "/opt/sonatype/nexus3" 53 seconds ago Up 51 seconds 0.0.0.0:8081->8081/tcp, [::]:8081->8081/tcp Nexus
root@ip-172-31-23-72:~# ^C
root@ip-172-31-23-72:~# docker exec -it f1d618e32968 /bin/bash
bash-5.1$ ls
nexus sonatype-work start-nexus-repository-manager.sh
bash-5.1$ cd sonatype-work/
bash-5.1$ ls
nexus3
bash-5.1$
```

Then, navigate into the folder “**nexus3**” using the command:

```
cd nexus3/
```

```
root@ip-172-31-23-72:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e60b2: Pull complete
c5bc07a30871: Pull complete
46e9484471e5: Pull complete
25230a69af22: Pull complete
ed8e6720939: Pull complete
09cada718ac0: Pull complete
70fa5f825dd3: Download complete
Digest: sha256:0ab4c84823747aad31a1a35b4f646f28970e33b0b9da8014b19bcd5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
f1d618e329688ef04721f3cd3fc170538e5d37a1c1ac68a33d5b1799e8e02b4
root@ip-172-31-23-72:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
f1d618e32968     sonatype/nexus3   "/opt/sonatype/nexus..."   53 seconds ago   Up 51 seconds   0.0.0.0:8081->8081/tcp, :::8081->8081/tcp   Nexus
root@ip-172-31-23-72:~# ^C
root@ip-172-31-23-72:~# docker exec -it f1d618e32968 /bin/bash
bash-5.1$ ls
nexus  sonatype-work  start-nexus-repository-manager.sh
bash-5.1$ cd sonatype-work/
bash-5.1$ ls
nexus3
bash-5.1$ cd nexus3/
bash-5.1$
```

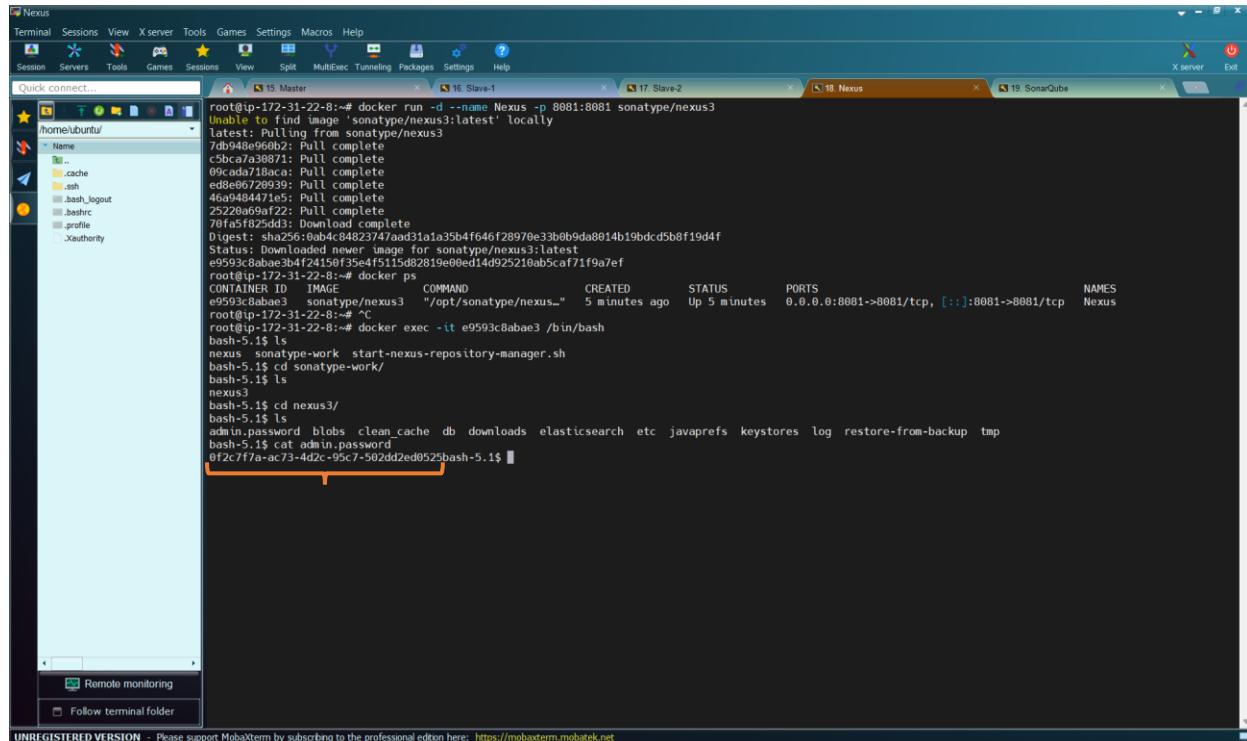
Then, run the command:

```
ls
```

```
root@ip-172-31-23-72:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e60b2: Pull complete
c5bc07a30871: Pull complete
46e9484471e5: Pull complete
25230a69af22: Pull complete
ed8e6720939: Pull complete
09cada718ac0: Pull complete
70fa5f825dd3: Download complete
Digest: sha256:0ab4c84823747aad31a1a35b4f646f28970e33b0b9da8014b19bcd5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
f1d618e329688ef04721f3cd3fc170538e5d37a1c1ac68a33d5b1799e8e02b4
root@ip-172-31-23-72:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
f1d618e32968     sonatype/nexus3   "/opt/sonatype/nexus..."   53 seconds ago   Up 51 seconds   0.0.0.0:8081->8081/tcp, :::8081->8081/tcp   Nexus
root@ip-172-31-23-72:~# ^C
root@ip-172-31-23-72:~# docker exec -it f1d618e32968 /bin/bash
bash-5.1$ ls
nexus  sonatype-work  start-nexus-repository-manager.sh
bash-5.1$ cd sonatype-work/
bash-5.1$ ls
nexus3
bash-5.1$ cd nexus3/
bash-5.1$ ls
admin.password  blobs  clean_cache  db  downloads  elasticsearch  etc  javaprefs  keystores  log  restore-from-backup  tmp
bash-5.1$
```

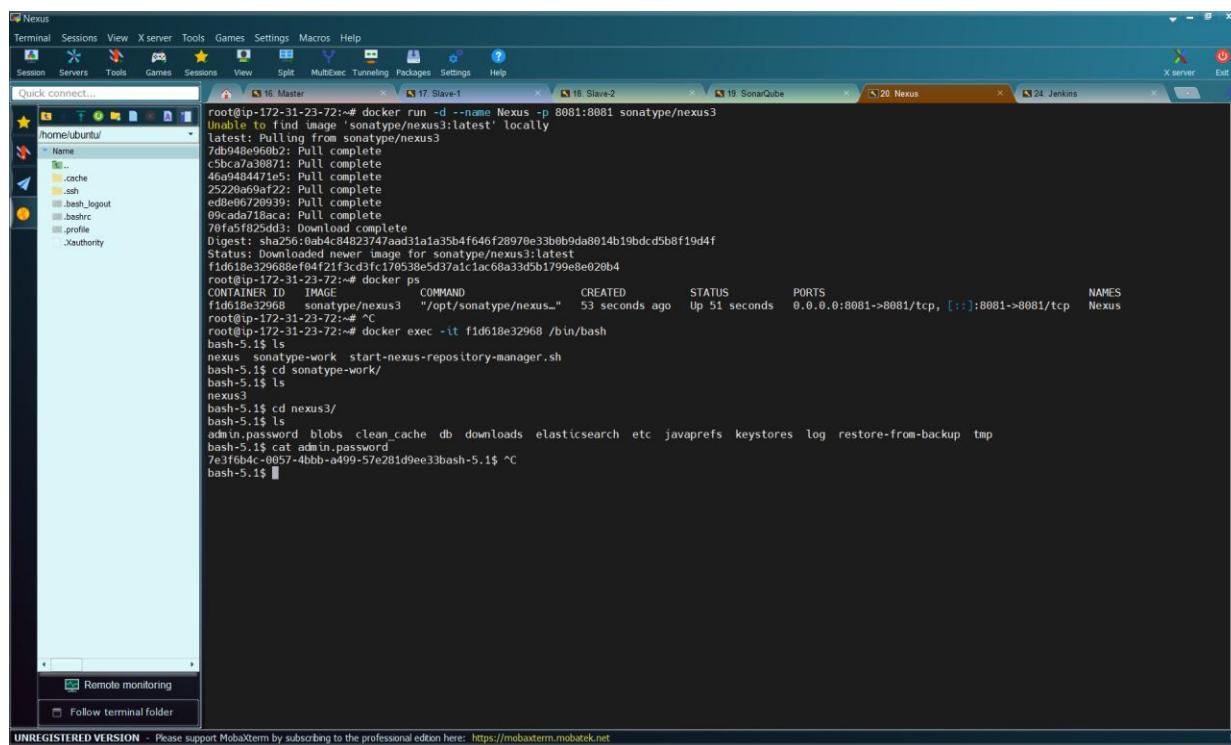
**View Admin Password:** Finally, you can view the admin password by displaying the contents of the admin.password file:

```
cat admin.password
```



```
root@ip-172-31-22-8:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e960b2: Pull complete
:c5bc7a30871: Pull complete
:09cad0718ac: Pull complete
:ed8e06720939: Pull complete
:46a9484471e5: Pull complete
:25220a69a22: Pull complete
70fa5f025dd3: Download complete
Digest: sha256:0a84c8482374aa31a1a35b4f646f28970e33b0b9da8014b19bdc5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
e9593c8abae3b4724150f35e4f515d82819e06ed14d925210ab5caf71f9a7ef
root@ip-172-31-22-8:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
root@ip-172-31-22-8:~# ^C
root@ip-172-31-22-8:~# docker exec -it e9593c8abae3 /bin/bash
bash-5.1$ ls
nexus sonatype-work start-nexus-repository-manager.sh
bash-5.1$ cd sonatype-work/
bash-5.1$ ls
nexus3
bash-5.1$ cd nexus3/
bash-5.1$ ls
admin.password blobs clean_cache db downloads elasticsearch etc javaprefs keystores log restore-from-backup tmp
bash-5.1$ cat admin.password
0f2c7f7a-ac73-4d2c-95c7-502dd2ed0525bash-5.1$
```

Copy the above part: 0f2c7f7a-ac73-4d2c-95c7-502dd2ed0525 by using “ctrl+c”. This is the password we will use on the Nexus browser.

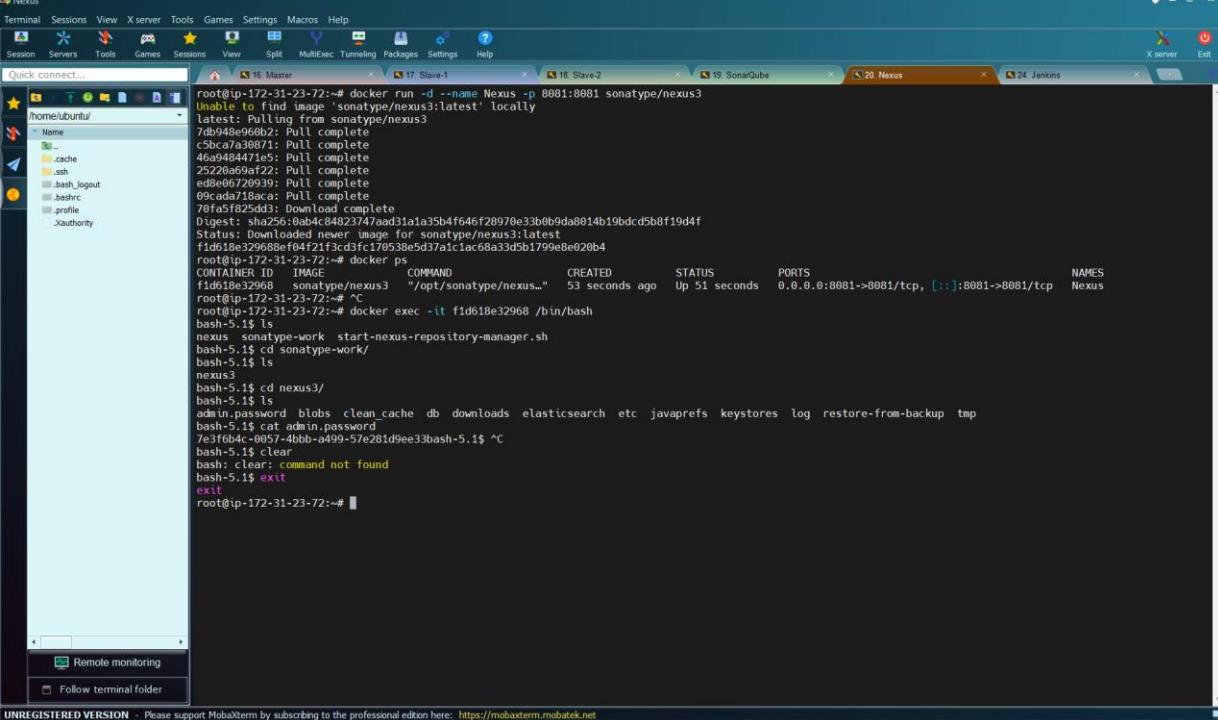


```
root@ip-172-31-23-72:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e960b2: Pull complete
:c5bc7a30871: Pull complete
:46a9484471e5: Pull complete
:25220a69a22: Pull complete
:ed8e06720939: Pull complete
:09cad0718ac: Pull complete
:7d618e3296: Download complete
Digest: sha256:0a84c8482374aa31a1a35b4f646f28970e33b0b9da8014b19bdc5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
f1d618e329688ef04f21f3cd3fc179538e5d37a1c1ac68a33d5b1799e8e020b4
root@ip-172-31-23-72:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f1d618e32968 sonatype/nexus3 "/opt/sonatype/nexus3" 53 seconds ago Up 51 seconds 0.0.0.0:8081->8081/tcp Nexus
root@ip-172-31-23-72:~# ^C
root@ip-172-31-23-72:~# docker exec -it f1d618e32968 /bin/bash
bash-5.1$ ls
nexus sonatype-work start-nexus-repository-manager.sh
bash-5.1$ cd sonatype-work/
bash-5.1$ ls
nexus3
bash-5.1$ cd nexus3/
bash-5.1$ ls
admin.password blobs clean_cache db downloads elasticsearch etc javaprefs keystores log restore-from-backup tmp
bash-5.1$ cat admin.password
7e3fb64c-0bb5-a499-57e281d9ee33bash-5.1^C
bash-5.1$
```

This process allows you to access the Nexus admin password stored within the container. Make sure to keep this password secure, as it grants administrative access to your Nexus instance.

**Exit the Container Shell:** Once you have retrieved the password, you can exit the container's bash shell:

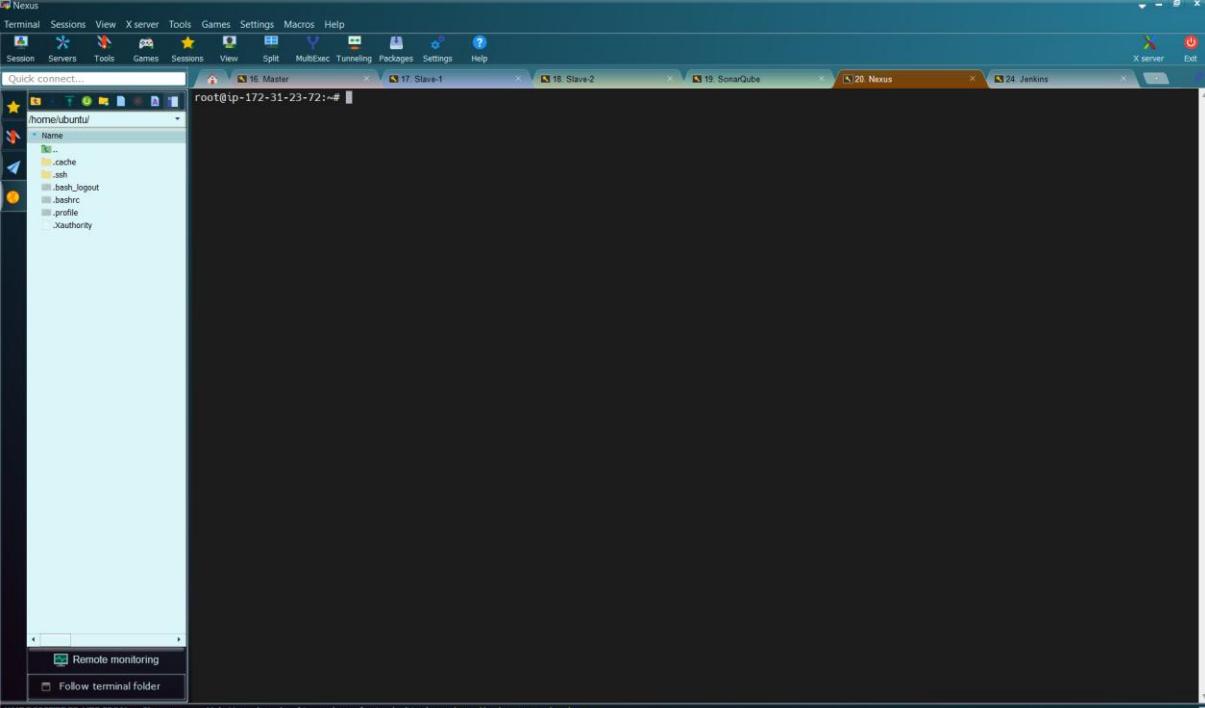
```
exit
```



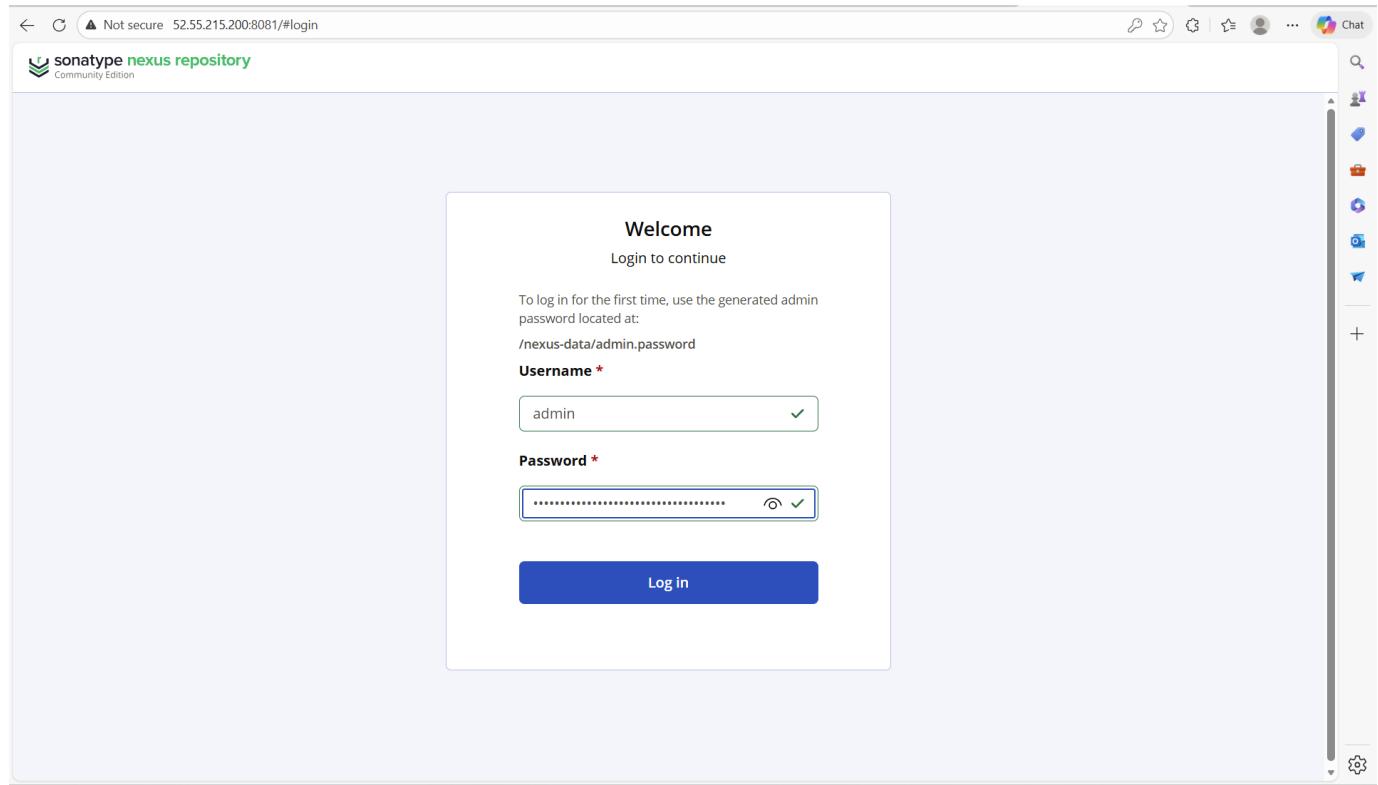
```
root@ip-172-31-23-72:~# docker run -d --name Nexus -p 8081:8081 sonatype/nexus3
Unable to find image 'sonatype/nexus3:latest' locally
latest: Pulling from sonatype/nexus3
7db948e960b2: Pull complete
c5bc07a3087a: Pull complete
f409a0a5545: Pull complete
25220a93f2: Pull complete
ed8e06720939: Pull complete
9cada718aca: Pull complete
70fa5f025dd3: Download complete
Digest: sha256:9ab4c8482374aad31a1a35bf646f28970e33bb0b9da0814b19bdc5b8f19d4f
Status: Downloaded newer image for sonatype/nexus3:latest
fid618e329688ef04721f3cd5fc170538e5d37a1c1ac68a33d5b1799e8e020b4
root@ip-172-31-23-72:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
fid618e329688ef04721f3cd5fc170538e5d37a1c1ac68a33d5b1799e8e020b4 Nexus
root@ip-172-31-23-72:~# ^C
root@ip-172-31-23-72:~# docker exec -it fid618e32968 /bin/bash
bash-5.1$ ls
nexus sonatype-work start-nexus-repository-manager.sh
bash-5.1$ cd sonatype-work/
bash-5.1$ ls
nexus3
bash-5.1$ cd nexus3/
bash-5.1$ ls
admin.password blobs clean.cache db downloads elasticsearch etc javaprefs keystores log restore-from-backup tmp
bash-5.1$ cat admin.password
7e1fd0557-40bb-a499-57e281d9ee33bash-5.1$ ^
bash-5.1$ clear
bash: clear: command not found
bash-5.1$ exit
root@ip-172-31-23-72:~#
```

Then clear the screen using the command:

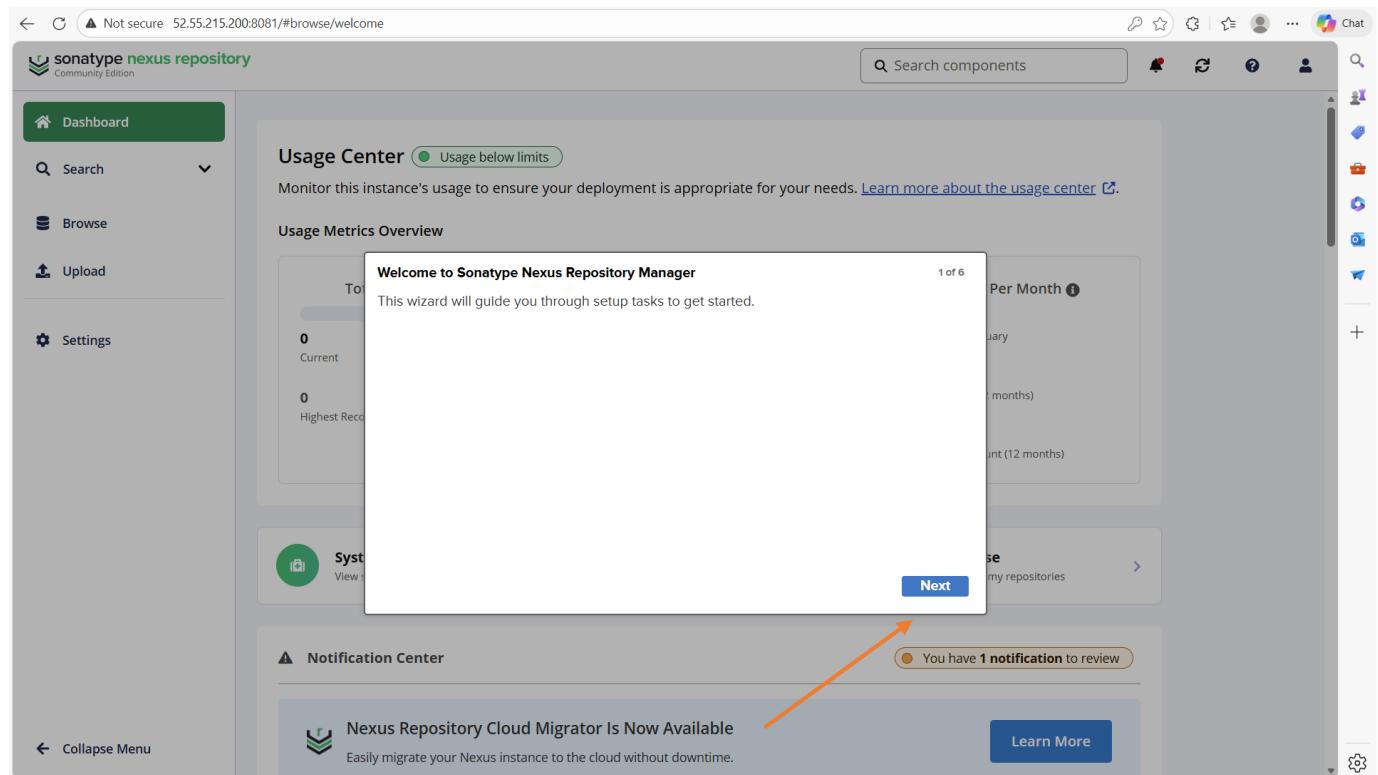
```
clear
```



Paste the copied password on the “Password” field on the Nexus browser



Click on “Log in”



Click on “Next”

The screenshot shows the Sonatype Nexus Repository interface. A modal window titled "Please choose a password for the admin user" is open. It contains two input fields: "New password:" and "Confirm password:", both of which have placeholder text consisting of four dots. Below the input fields are "Back" and "Next" buttons. To the right of the modal, there is a progress bar indicating "2 of 6". In the background, the "Usage Center" section is visible, showing a green status indicator "Usage below limits" and a message about monitoring usage. The left sidebar includes links for Dashboard, Search, Browse, Upload, and Settings.

You can now change the password. Enter the new password and confirm it.

The screenshot shows the same Sonatype Nexus Repository interface as the previous one, but with an orange arrow pointing to the "Next" button in the modal window. The modal window is identical to the one in the first screenshot, showing the "Please choose a password for the admin user" step. The background "Usage Center" section and the left sidebar are also visible.

Then, click on “Next”

Not secure 52.55.215.200:8081/#browse/welcome

sonatype nexus repository Community Edition

Dashboard Search Browse Upload Settings

Usage Center Usage below limits

Monitor this instance's usage to ensure your deployment is appropriate for your needs. [Learn more about the usage center](#).

Usage Metrics Overview

Discover Community Edition

Sonatype Nexus Repository Community Edition provides powerful tools to support your deployments.

[Learn more about the Community Edition](#).

**Benefits:** Support for all formats, OCI, PostgreSQL, Kubernetes, and smart usage tracking.

3 of 6 Per Month

0 Current 0 Highest Recount (12 months)

System View Back Next

Notification Center You have 1 notification to review

Nexus Repository Cloud Migrator Is Now Available Easily migrate your Nexus instance to the cloud without downtime. Learn More

Collapse Menu

Click on “Next”

Not secure 52.55.215.200:8081/#browse/welcome

sonatype nexus repository Community Edition

Dashboard Search Browse Upload Settings

Usage Center Usage below limits

Monitor this instance's usage to ensure your deployment is appropriate for your needs. [Learn more about the usage center](#).

Usage Metrics Overview

Agree End User License Agreement

END USER LICENSE AGREEMENT PLEASE READ THIS AGREEMENT CAREFULLY

This End User License Agreement (this "Agreement") is entered into by and between Sonatype, Inc. ("Sonatype") and you ("Licensee") and governs Licensee's use of and access to the Product to which Licensee has requested access. Sonatype and Licensee may be referred to individually in this Agreement as a "Party" or collectively as the "Parties."

**Licensee agrees that, unless Sonatype has provided its express written consent, Sonatype's competitors, including anyone acting on their behalf, are strictly prohibited from accessing the Product.**

By clicking 'Agree', you acknowledge that you have read and agree to the End User License Agreement.

Back Agree

Notification Center You have 1 notification to review

Nexus Repository Cloud Migrator Is Now Available Easily migrate your Nexus instance to the cloud without downtime. Learn More

Collapse Menu

Click on “Agree”

sonatype nexus repository  
Community Edition

Usage Center Usage below limits

Monitor this instance's usage to ensure your deployment is appropriate for your needs. [Learn more about the usage center](#).

Usage Metrics Overview

Configure Anonymous Access

Enable anonymous access means that by default, users can search, browse and download components from repositories without credentials. Please consider the security implications for your organization.

Disable anonymous access should be chosen with care, as it will require credentials for all users and/or build tools.

More information

Enable anonymous access  
 Disable anonymous access

Back Next

Per Month

January (1 month)  
February (1 month)  
March (1 month)

Use my repositories

Notification Center You have 1 notification to review

Nexus Repository Cloud Migrator Is Now Available

Easily migrate your Nexus instance to the cloud without downtime. [Learn More](#)

Check the box on “Enable anonymous access”

sonatype nexus repository  
Community Edition

Usage Center Usage below limits

Monitor this instance's usage to ensure your deployment is appropriate for your needs. [Learn more about the usage center](#).

Usage Metrics Overview

Configure Anonymous Access

Enable anonymous access means that by default, users can search, browse and download components from repositories without credentials. Please consider the security implications for your organization.

Disable anonymous access should be chosen with care, as it will require credentials for all users and/or build tools.

More information

Enable anonymous access  
 Disable anonymous access

Back Next

Per Month

January (1 month)  
February (1 month)  
March (1 month)

Use my repositories

Notification Center You have 1 notification to review

Nexus Repository Cloud Migrator Is Now Available

Easily migrate your Nexus instance to the cloud without downtime. [Learn More](#)

Click on “Next”

The screenshot shows the Sonatype Nexus Repository Community Edition dashboard. On the left, there's a sidebar with links for Dashboard, Search, Browse, Upload, and Settings. The main area is titled 'Usage Center' with a green status indicator 'Usage below limits'. It says 'Monitor this instance's usage to ensure your deployment is appropriate for your needs.' Below this is the 'Usage Metrics Overview' section. A central callout box is titled 'Setup Complete' with the message 'Your instance is now ready to use. Explore Sonatype Nexus Repository to unlock its full potential.' At the bottom right of this box is a blue 'Finish' button. An orange arrow points from the text above to this 'Finish' button.

Then, click on “Finish”

The screenshot shows the same dashboard as before, but the sidebar has changed. The 'Search' link is now grayed out, while 'Browse', 'Upload', and 'Settings' are in their original colors. An orange arrow points from the text above to the 'Browse' link in the sidebar.

Nexus has been set up. Click on browse

The screenshot shows the Sonatype Nexus Repository Manager interface. The left sidebar has a 'Browse' button highlighted in green. The main area is titled 'Browse' and lists several repositories:

Name	Type	Format	Status	URL	Health check	Firewall Re...
maven-central	proxy	maven2	Online - Ready to Connect		0 0	
maven-public	group	maven2	Online			
maven-releases	hosted	maven2	Online			
maven-snapshots	hosted	maven2	Online			
nuget-group	group	nuget	Online			
nuget-hosted	hosted	nuget	Online			
nuget.org-proxy	proxy	nuget	Online - Ready to Connect		0 0	

At the bottom left is a 'Collapse Menu' button.

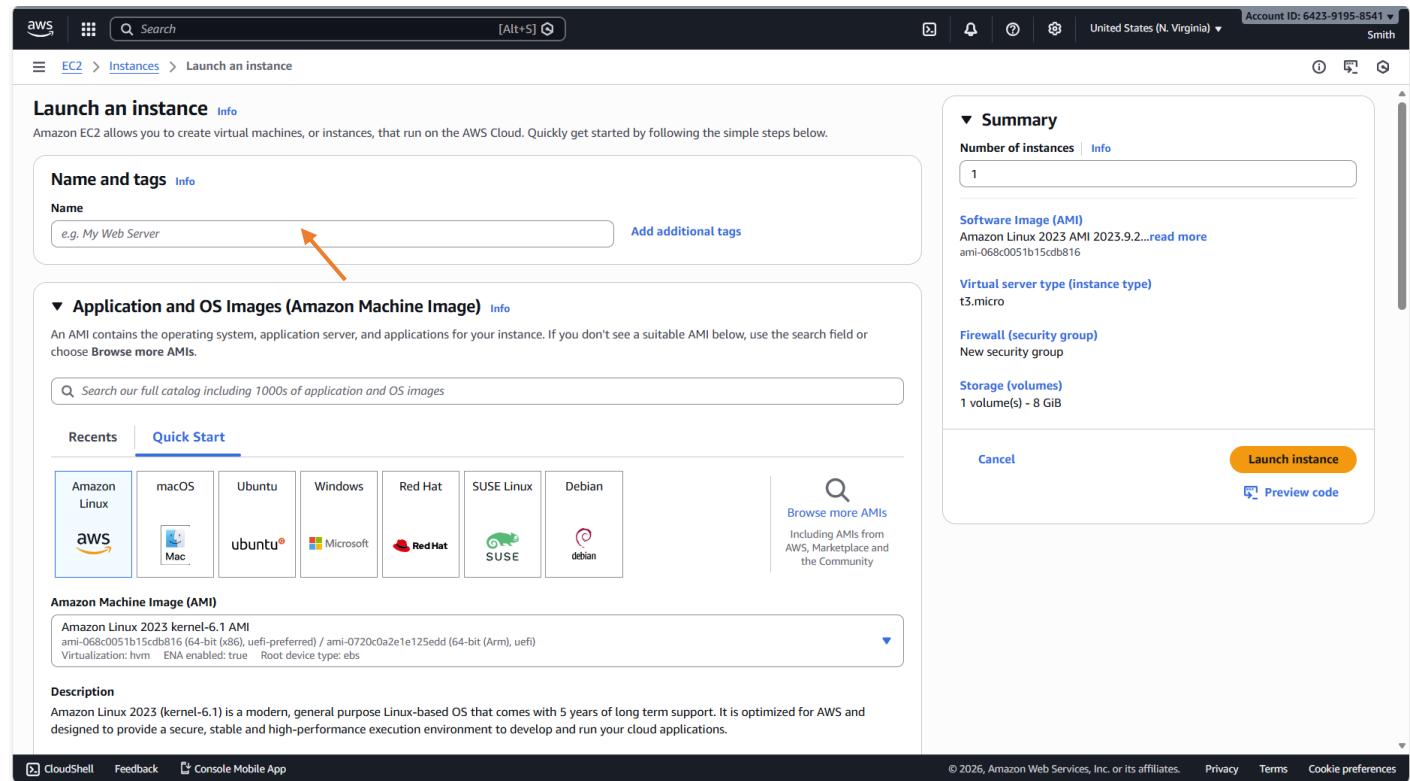
You can see different Nexus repositories where you can push different files if you want. We can now go to the terminal of the Nexus virtual machine and clear it

## 3.7 Create and Configure Virtual Machine for Jenkins

Let us create a virtual machine that will serve as our server for Jenkins.

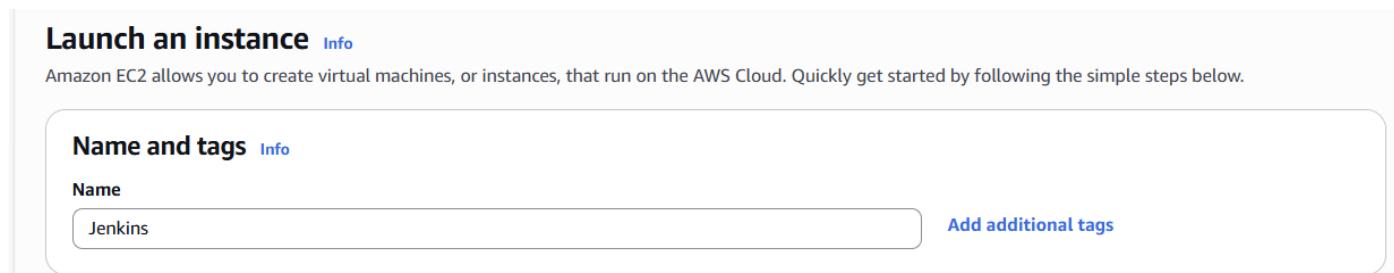
### 3.7.1 Create Virtual Machine for Jenkins

We will start by creating the virtual machine for Jenkins called “**Jenkins**”. Go AWS Management console.



The screenshot shows the AWS Management Console interface for launching a new EC2 instance. The top navigation bar includes the AWS logo, a search bar, and account information. The main content area is titled "Launch an instance". The first step, "Name and tags", has a red arrow pointing to the "Name" input field which contains "e.g. My Web Server". Below it is the "Application and OS Images (Amazon Machine Image)" section, which lists various operating systems like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. The third step, "Summary", shows the configuration for the launch: one instance, Amazon Linux 2023 AMI, t3.micro instance type, and a new security group. At the bottom right of the summary step is a prominent orange "Launch instance" button.

Let us give the virtual machine a name, we will call it “**Jenkins**”



The screenshot shows the "Launch an instance" wizard on the "Name and tags" step. The "Name" input field is populated with "Jenkins". To the right of the input field is a blue "Add additional tags" button. The rest of the wizard interface is visible above and below this step.

Then on “**Application and OS Images (Amazon Machine Image)**” and select “**Ubuntu**”

## ▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recents

Quick Start



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

### Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
ami-0ecb62995f68bb549 (64-bit (x86)) / ami-01b9f1e7dc427266e (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

#### Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x86)

AMI ID

ami-0ecb62995f68bb549

Publish Date

2025-10-22

Username

ubuntu

Verified provider

Scroll down to “Instance Type” and select “t2.large”

## ▼ Instance type [Info](#) | [Get advice](#)

### Instance type

t2.large

Family: t2 2 vCPU 8 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.1208 USD per Hour On-Demand RHEL base pricing: 0.1216 USD per Hour  
On-Demand SUSE base pricing: 0.1928 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0963 USD per Hour  
On-Demand Linux base pricing: 0.0928 USD per Hour

All generations

[Compare instance types](#)

**Additional costs apply for AMIs with pre-installed software**

Scroll down to “Key Pair” and select the key we created previously

## ▼ Instance type [Info](#) | [Get advice](#)

### Instance type

t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true  
On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour On-Demand Linux base pricing: 0.0464 USD per Hour  
On-Demand RHEL base pricing: 0.0752 USD per Hour On-Demand Windows base pricing: 0.0644 USD per Hour  
On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

[Compare instance types](#)

**Additional costs apply for AMIs with pre-installed software**

Scroll down to “Network Settings” and select the security group we created.

**▼ Network settings** [Info](#)

[Edit](#)

**Network** | [Info](#)  
vpc-0d74d3736a240e572

**Subnet** | [Info](#)  
No preference (Default subnet in any availability zone)

**Auto-assign public IP** | [Info](#)  
Enable

**Firewall (security groups)** | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group       Select existing security group

**Common security groups** | [Info](#)

Select security groups ▾

Primary-SG sg-002d4edfb66259799 [X](#)  
VPC: vpc-0d74d3736a240e572

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Scroll down to “Configure Storage” and make it “30GiB”

**▼ Configure storage** [Info](#)

Advanced

1x  GiB  Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

[Edit](#)

[Cancel](#) [Launch instance](#) [Preview code](#)

Click refresh to view backup information  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

**► Advanced details** [Info](#)

Then, click on “Launch Instance”

The screenshot shows the AWS EC2 Instances launch success page. At the top, there is a breadcrumb navigation: EC2 > Instances > Launch an instance. A red arrow points from the text "Click on 'Instances'" to the "Instances" link in the breadcrumb. Below the breadcrumb is a green success message: "Success Successfully initiated launch of instance (i-079228c84670de28)". There is a "Launch log" button below the message. A "Next Steps" section follows, containing eight cards: "Create billing usage alerts", "Connect to your instance", "Connect an RDS database", "Create EBS snapshot policy", "Manage detailed monitoring", "Create Load Balancer", "Create AWS budget", and "Manage CloudWatch alarms". Each card has a corresponding "Create" or "Learn more" button. At the bottom of the page are links for CloudShell, Feedback, and Console Mobile App, along with copyright information and links for Privacy, Terms, and Cookie preferences.

Click on “Instances”

The screenshot shows the AWS EC2 Instances list page. On the left is a navigation sidebar with sections: EC2 (selected), Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs). The main area displays a table titled "Instances (6) Info". The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone. One row shows an instance named "Jenkins" with the status "Running" and "Status check: 2/2 checks passed". Another row shows an instance named "Nexus" with the status "Running" and "Status check: 2/2 checks passed". A third row shows an instance named "Slave-1" with the status "Running" and "Status check: 2/2 checks passed". A fourth row shows an instance named "Slave-2" with the status "Running" and "Status check: 2/2 checks passed". A fifth row shows an instance named "Master" with the status "Running" and "Status check: 2/2 checks passed". A sixth row shows an instance named "SomarQube" with the status "Running" and "Status check: 2/2 checks passed". An arrow points from the text "We have created the instance and it is initializing, let us wait for it to pass the “2/2 check”" to the "Status check" column of the Jenkins instance, which currently shows "2/2 checks passed". The bottom of the page includes links for CloudShell, Feedback, and Console Mobile App, along with copyright information and links for Privacy, Terms, and Cookie preferences.

We have created the instance and it is initializing, let us wait for it to pass the “2/2 check”

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links for EC2 services like Dashboard, Global View, Events, Instances, Images, Elastic Block Store, Network & Security, and more. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0d72ba387d698a6b0	Running	t2.medium	2/2 checks passed...	<a href="#">View alarms +</a>	us-east-1c
Slave-1	i-0b025e08ef005624c	Running	t2.medium	2/2 checks passed...	<a href="#">View alarms +</a>	us-east-1c
Slave-2	i-0afb570889b99b7d9	Running	t2.medium	2/2 checks passed...	<a href="#">View alarms +</a>	us-east-1c
Jenkins	i-0792282c84670de28	Running	t2.large	2/2 checks passed...	<a href="#">View alarms +</a>	us-east-1c
SonarQube	i-072e7267a412f30e2	Running	t2.medium	2/2 checks passed...	<a href="#">View alarms +</a>	us-east-1c
Nexus	i-0a783cb1d7f52ae	Running	t2.medium	2/2 checks passed...	<a href="#">View alarms +</a>	us-east-1c

At the bottom of the table, a message says "Select an instance". A red arrow points from the "Launch instances" button at the top right towards the "Status check" column for the Master instance.

The virtual machine has passed the “**2/2 Check**”

### 3.7.2 SSH Connect to Virtual Machine for Jenkins

Let us create a duplicate of the session “**Master**”.

The screenshot shows the MobaXterm application interface. On the left, there's a sidebar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. The main area shows a list of sessions:

- Master
- Slave-1
- Slave-2
- SonarQube
- Nexus

The "Master" session is currently active, showing a terminal window with the following content:

```

MobaXterm Personal Edition v24.1
(SSH client, X server and network tools)

SSH session to ubuntu@52.55.215.200
  • Direct SSH : ✓
  • SSH compression : ✓
  • SSH-browser : ✓
  • X11-forwarding : ✓ (remote display is forwarded through SSH)

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

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Sat Jan 3 20:11:41 UTC 2024
  System load: 0.0          Processes: 121
  Usage of /: 9.5% of 18.33GB   Users logged in: 0
  Memory usage: 6%           IPv4 address for enx0: 172.31.23.72
  Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

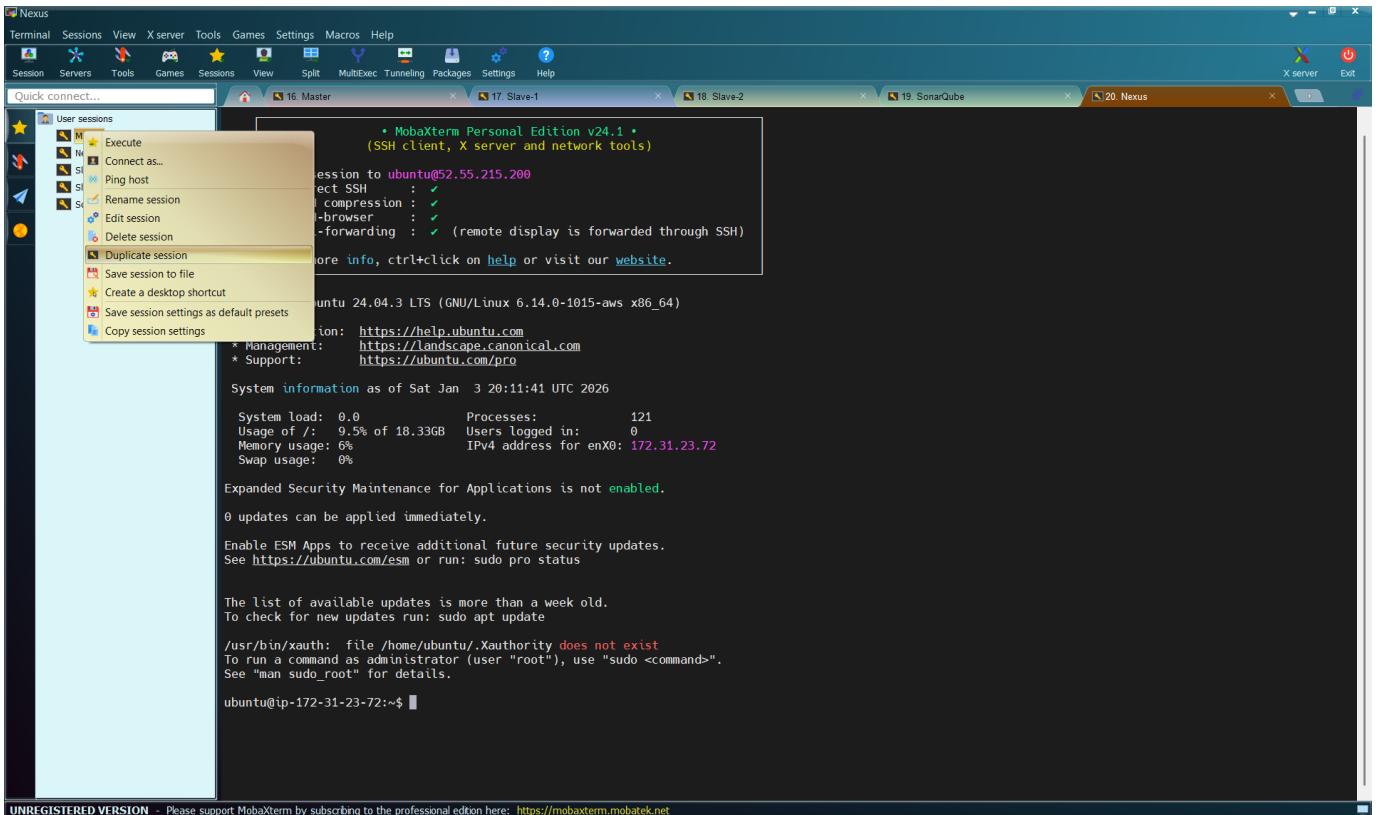
/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-23-72:~$ 

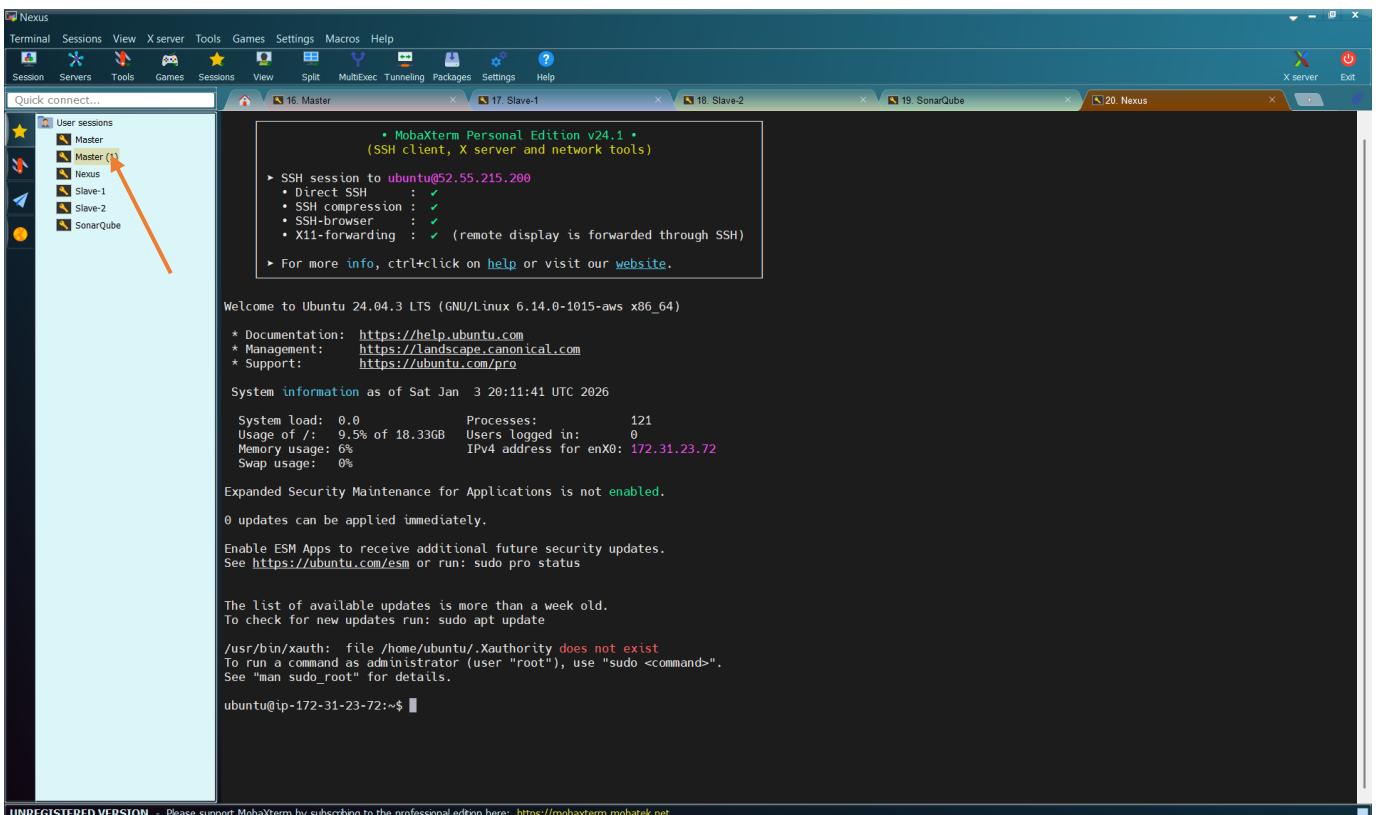
```

A red arrow points from the "Master" session name in the sidebar to the terminal window.

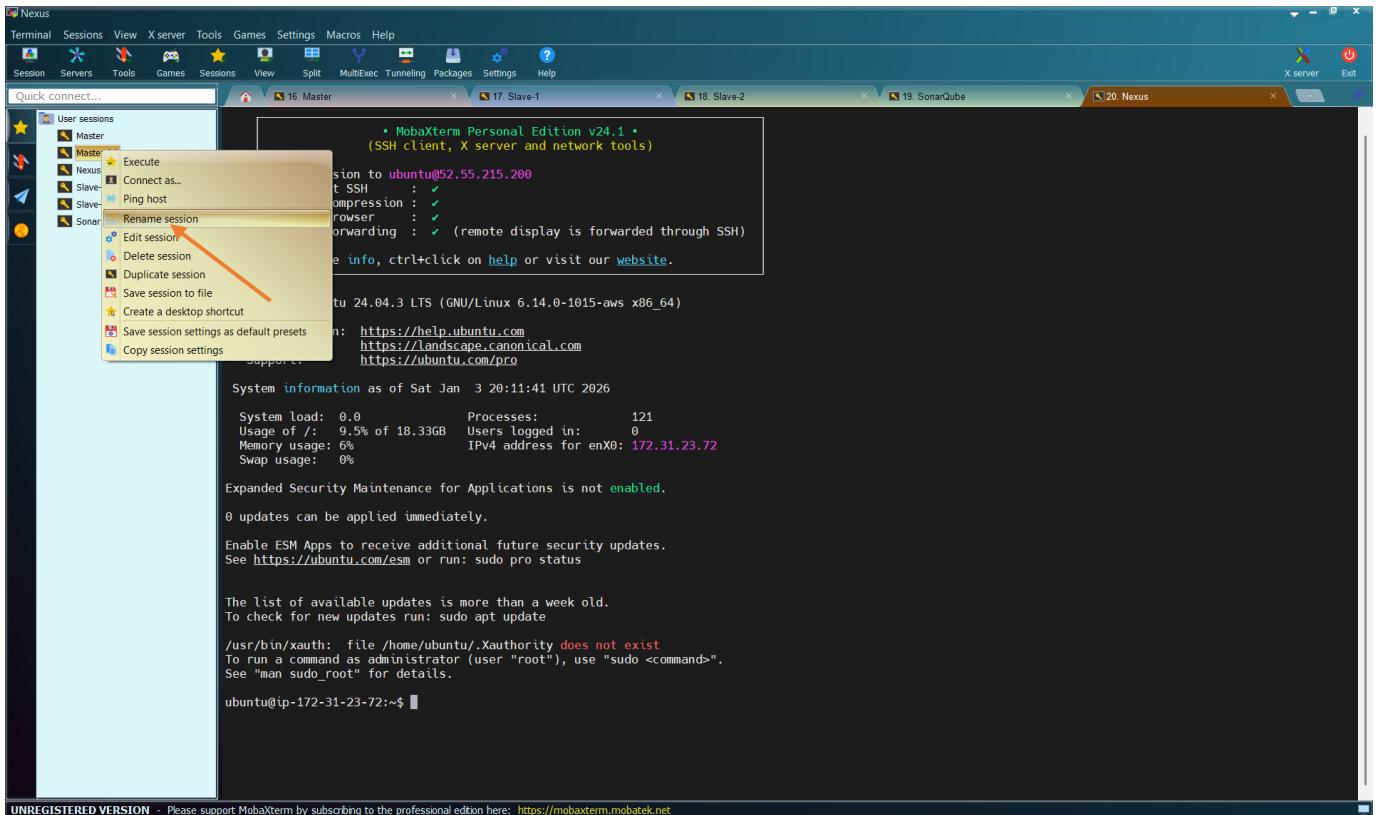
Right-click on the session name “**Master**”



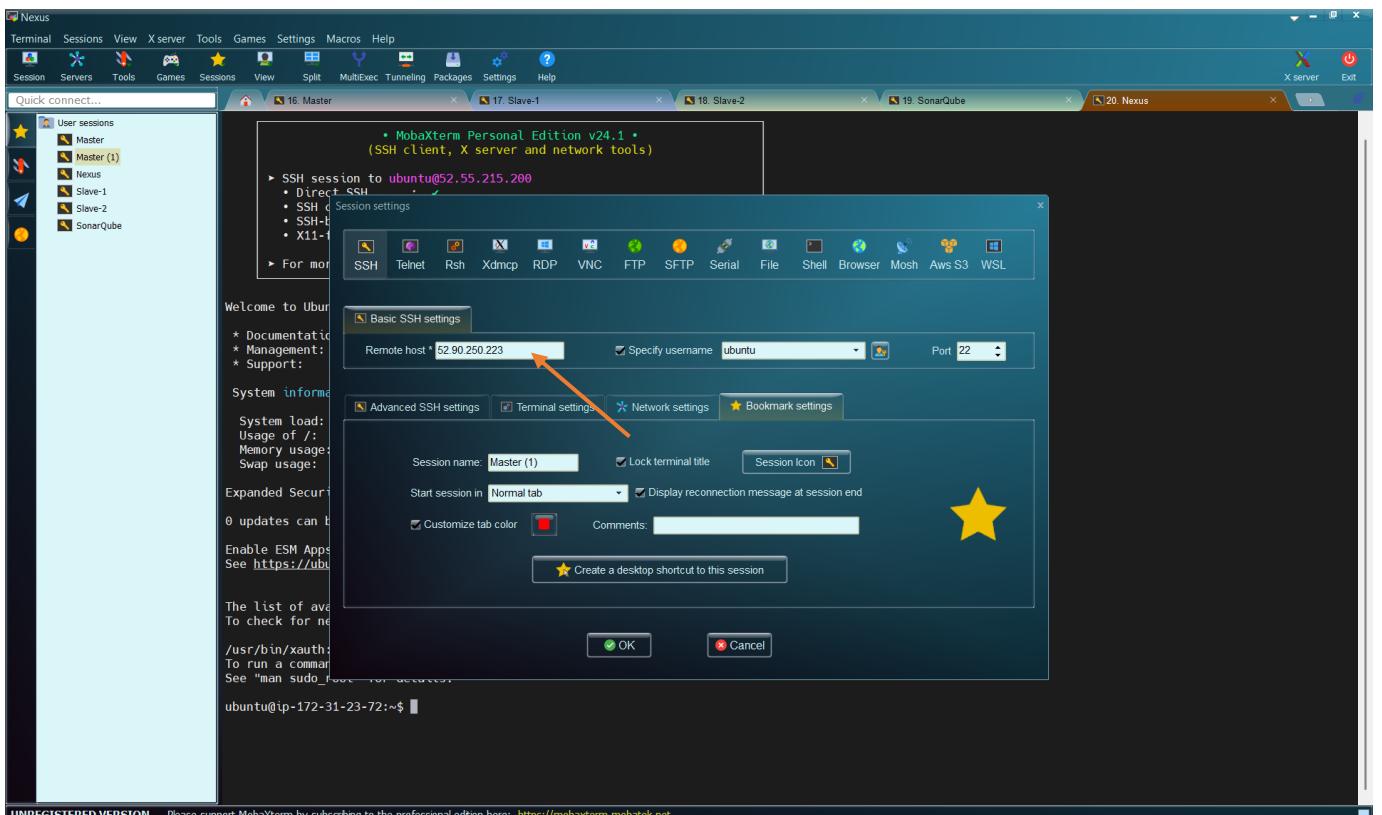
## Select “Duplicate Session”



We want to rename “Master(1)” to “Jenkins”. Right-click on “Master(1)”



## Select “Rename Session”



Copy the Public IP address of our “Nexus” virtual machine and paste here.

**Instances (1/6) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0d72ba387d698a6b0	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c
Slave-1	i-0b025e08ef003624c	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c
Slave-2	i-0afb570889b9b7d9	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c
SonarQube	i-072e7267a412f30e2	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c
<b>Jenkins</b>	<b>i-07b86791118f8bf95</b>	<b>Running</b>	<b>t2.large</b>	<b>2/2 checks passed</b>	<b>View alarms +</b>	<b>us-east-1c</b>
Nexus	i-0a783c8bc1d7f52ae	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c

**i-07b86791118f8bf95 (Jenkins)**

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

**Instance summary**

Instance ID: i-07b86791118f8bf95  
 Public IPv4 address: 35.172.225.123 [open address]  
 Instance state: Running  
 Private IP DNS name (IPv4 only): ip-172-31-18-218.ec2.internal  
 Instance type: t2.large

Private IPv4 addresses: 172.31.18.218  
 Public DNS: ec2-35-172-225-123.compute-1.amazonaws.com [open address]

Copy the Public IP: **35.172.225.123** and paste the MobaXterm

**New**

**Session** **Servers** **Tools** **Games** **Sessions** **View** **X server** **Tools** **Games** **Settings** **Macros** **Help**

**Quick connect...**

User sessions:

- Master
- Master (1)
- Nexus
- Slave-1
- Slave-2
- SonarQube

MobaXterm Personal Edition v24.1 (SSH client, X server and network tools)

SSH session to ubuntu@52.55.215.200

Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-72-generic x86\_64)

Session settings:

Basic SSH settings:

- Remote host: 13.218.172.184
- Specify username: ubuntu
- Port: 22

Advanced SSH settings:

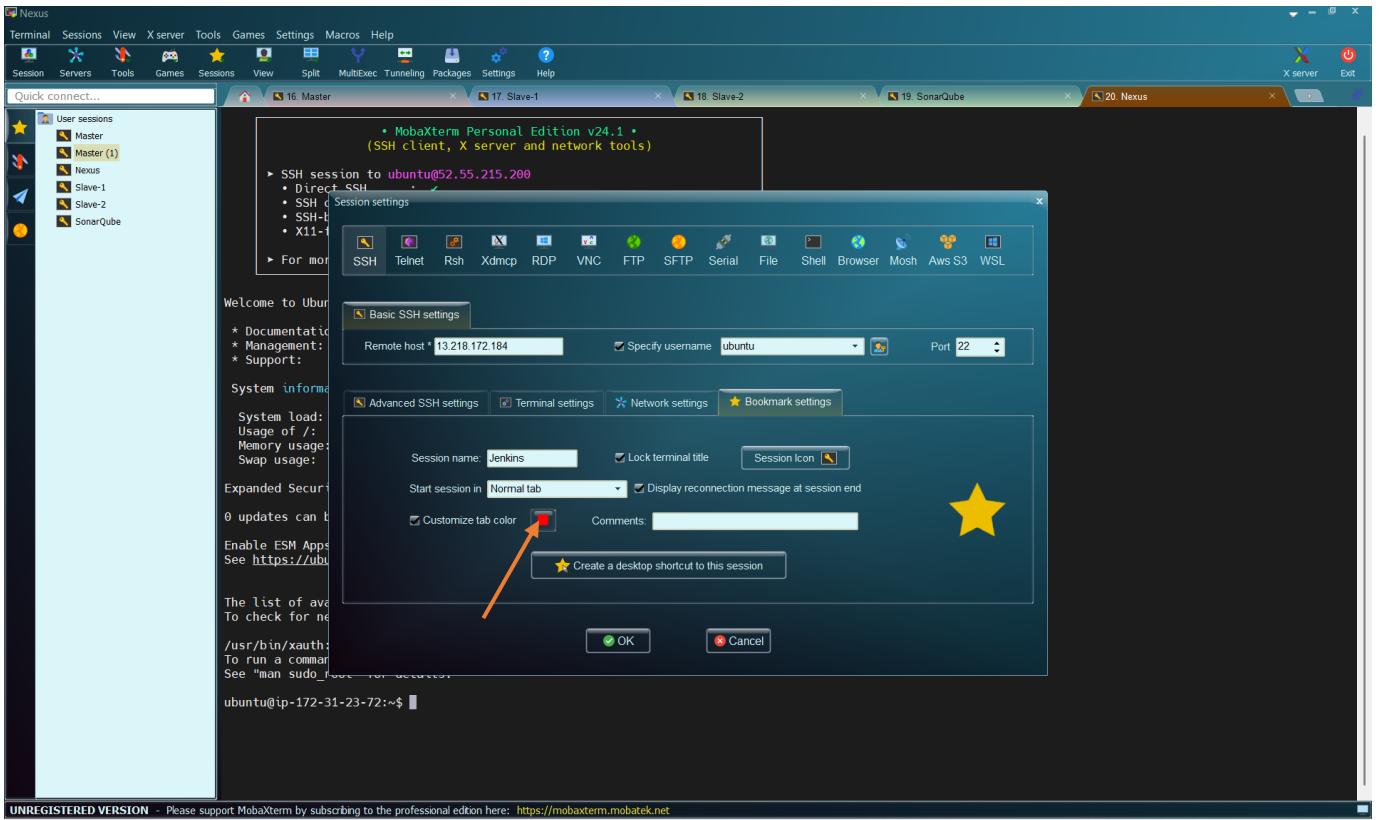
- Session name: Master (1)
- Start session in: Normal tab
- Comments:

OK Cancel

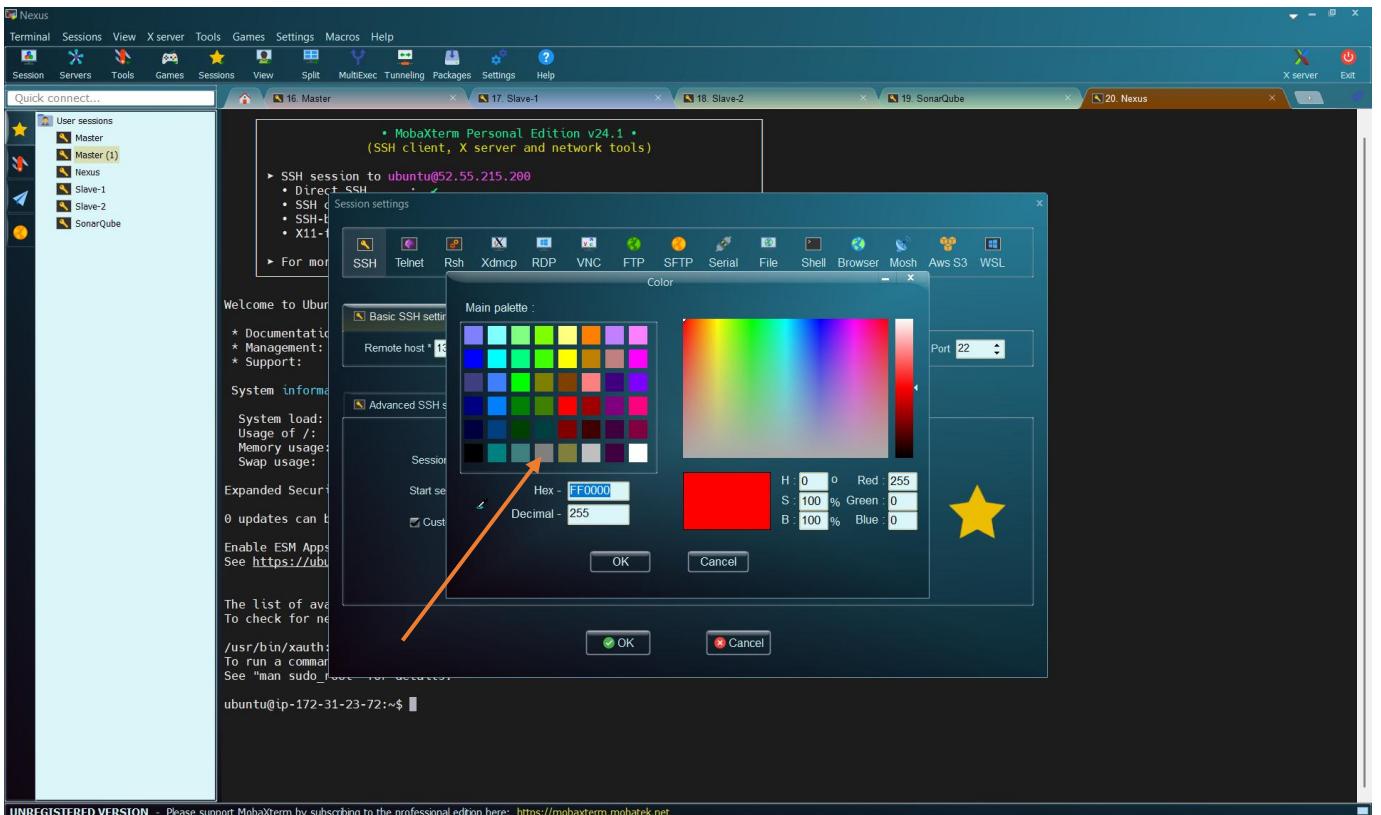
ubuntu@ip-172-31-23-72:~\$

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

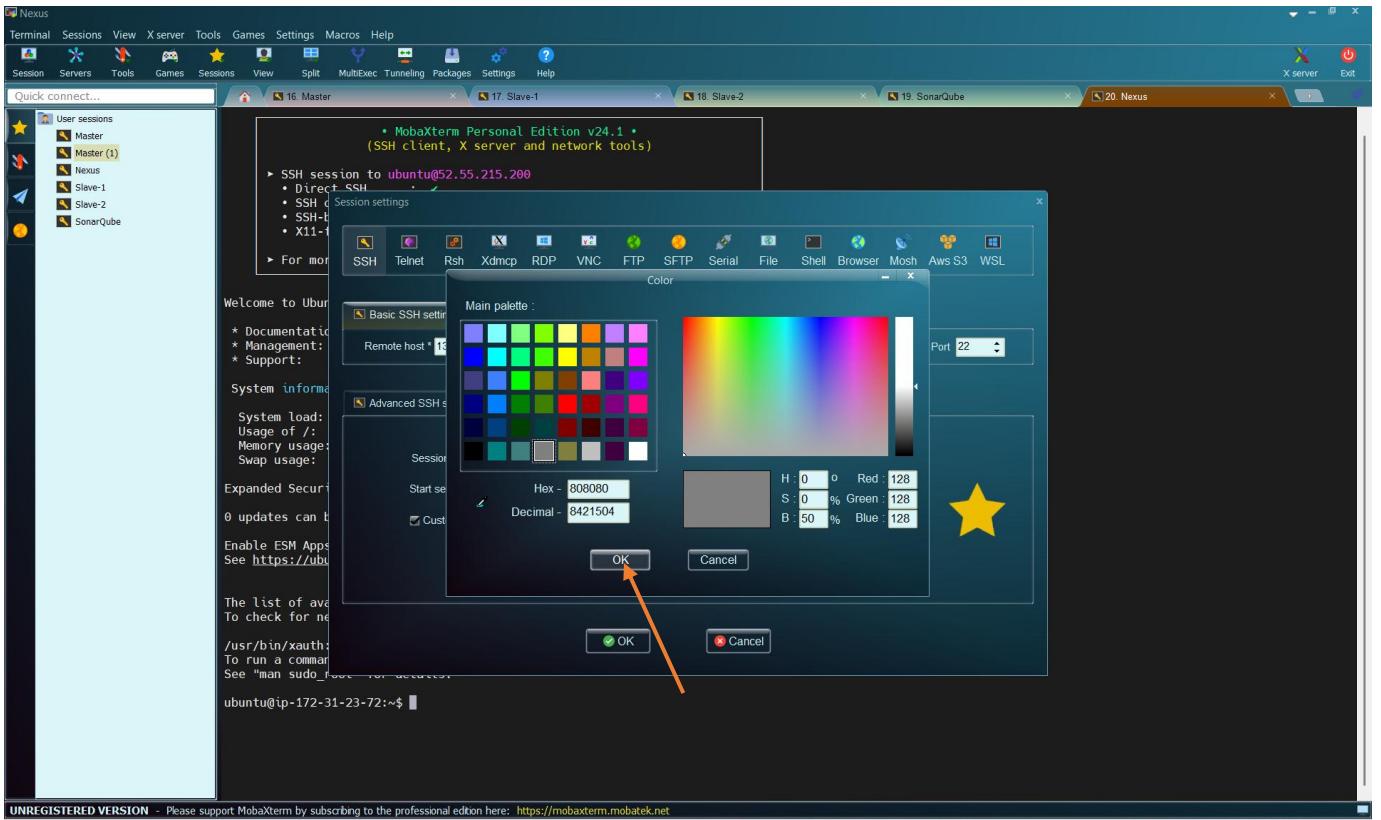
Then, change the name to “**Jenkins**”



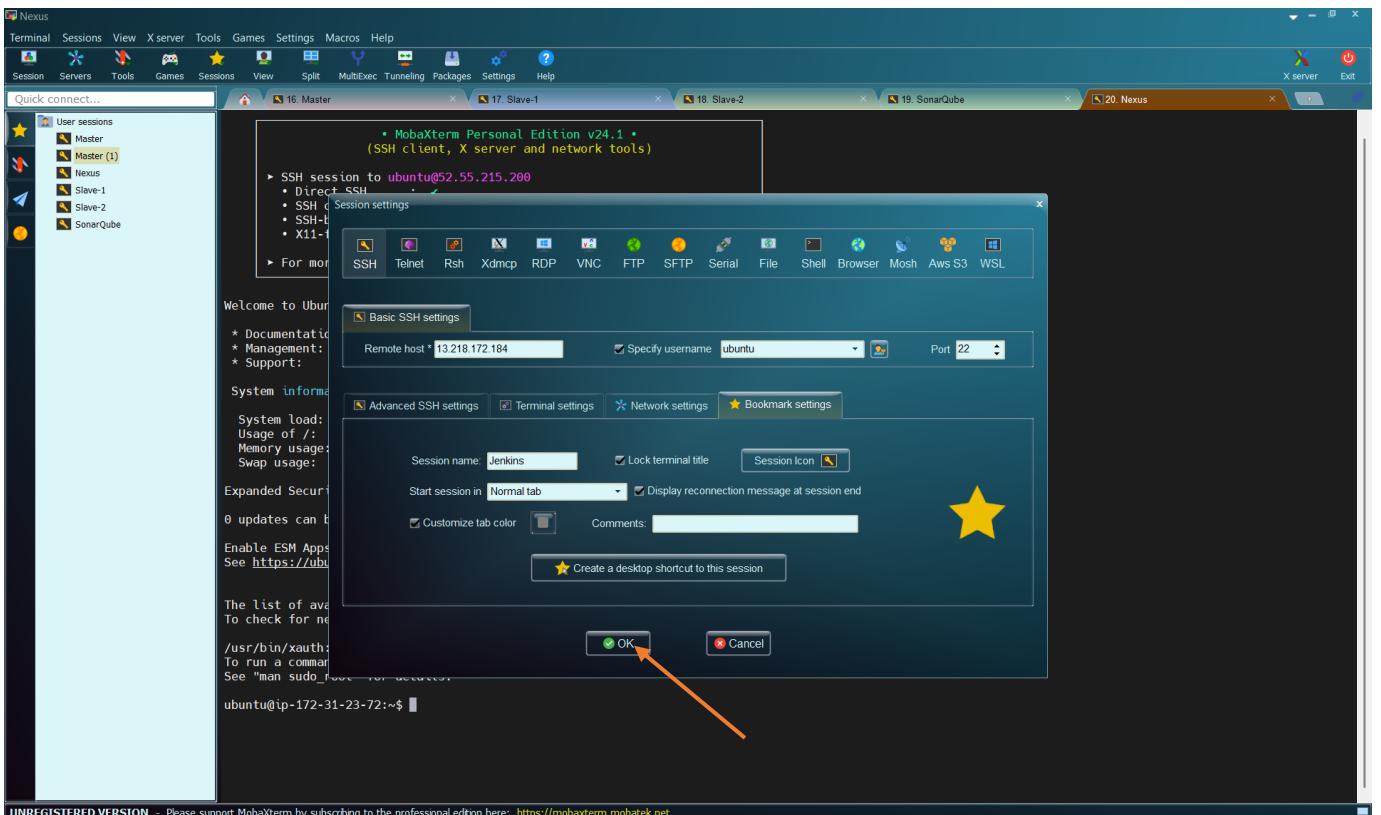
Then, let us change the color. Click on the RED box on “Customize tab color”



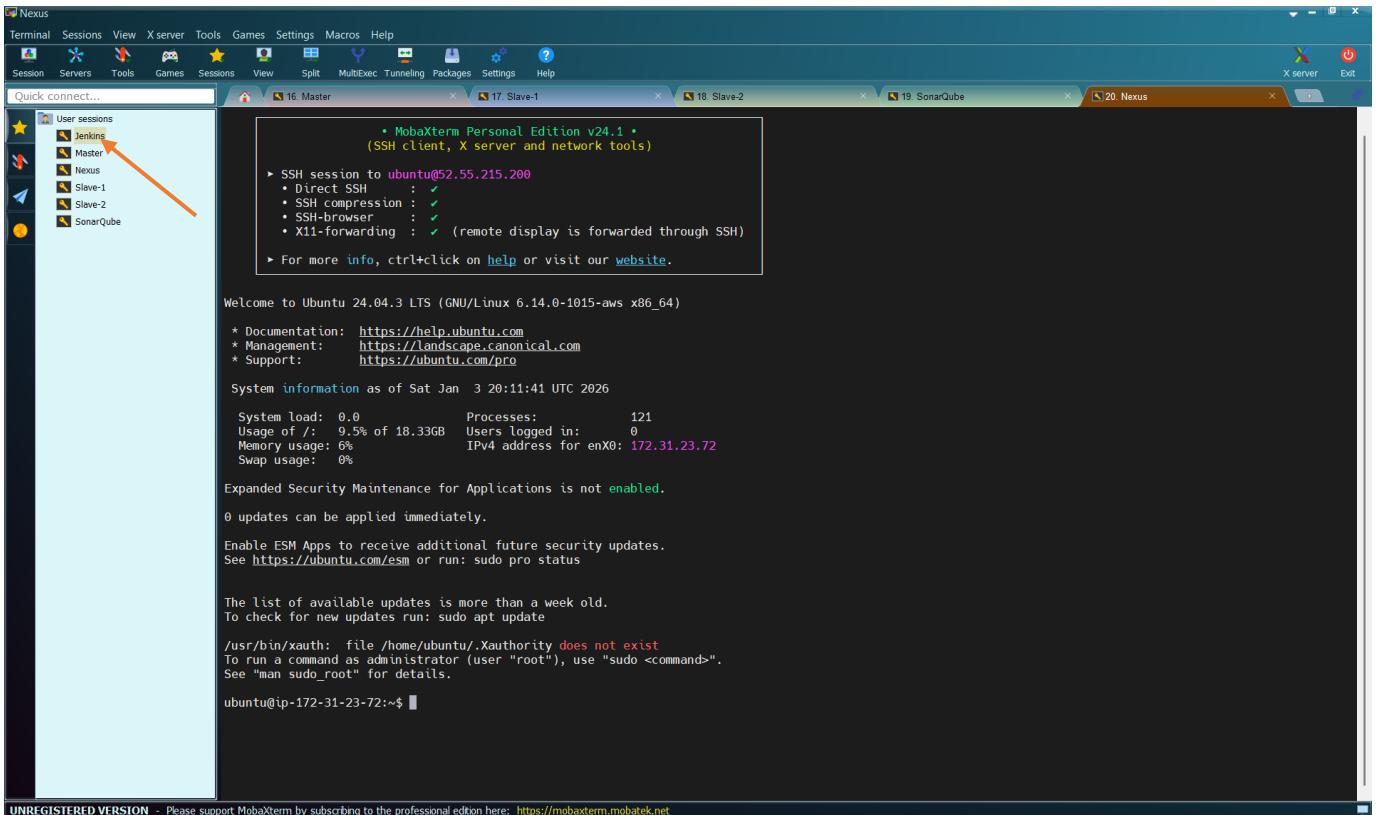
Let us select “Grey” box



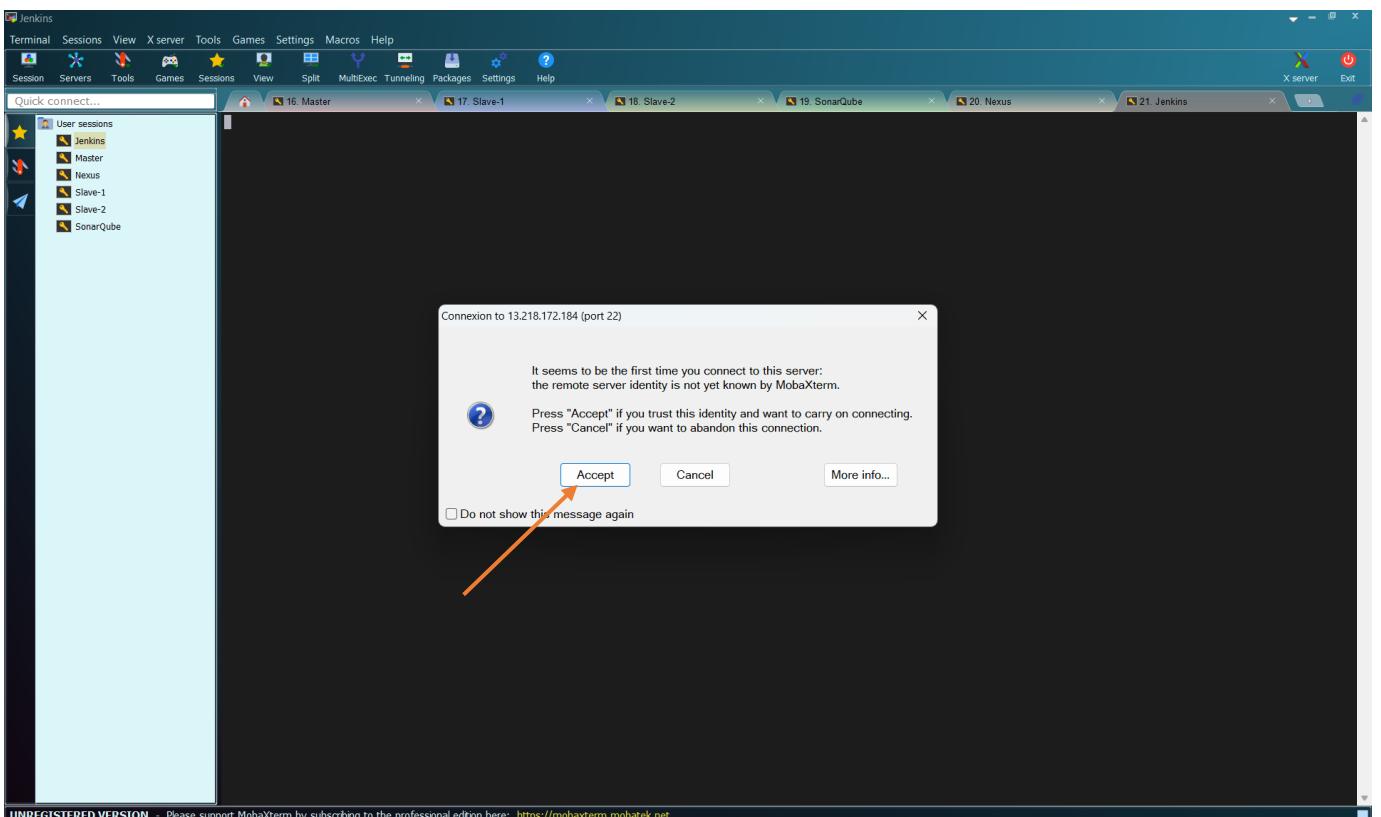
Click on “OK”



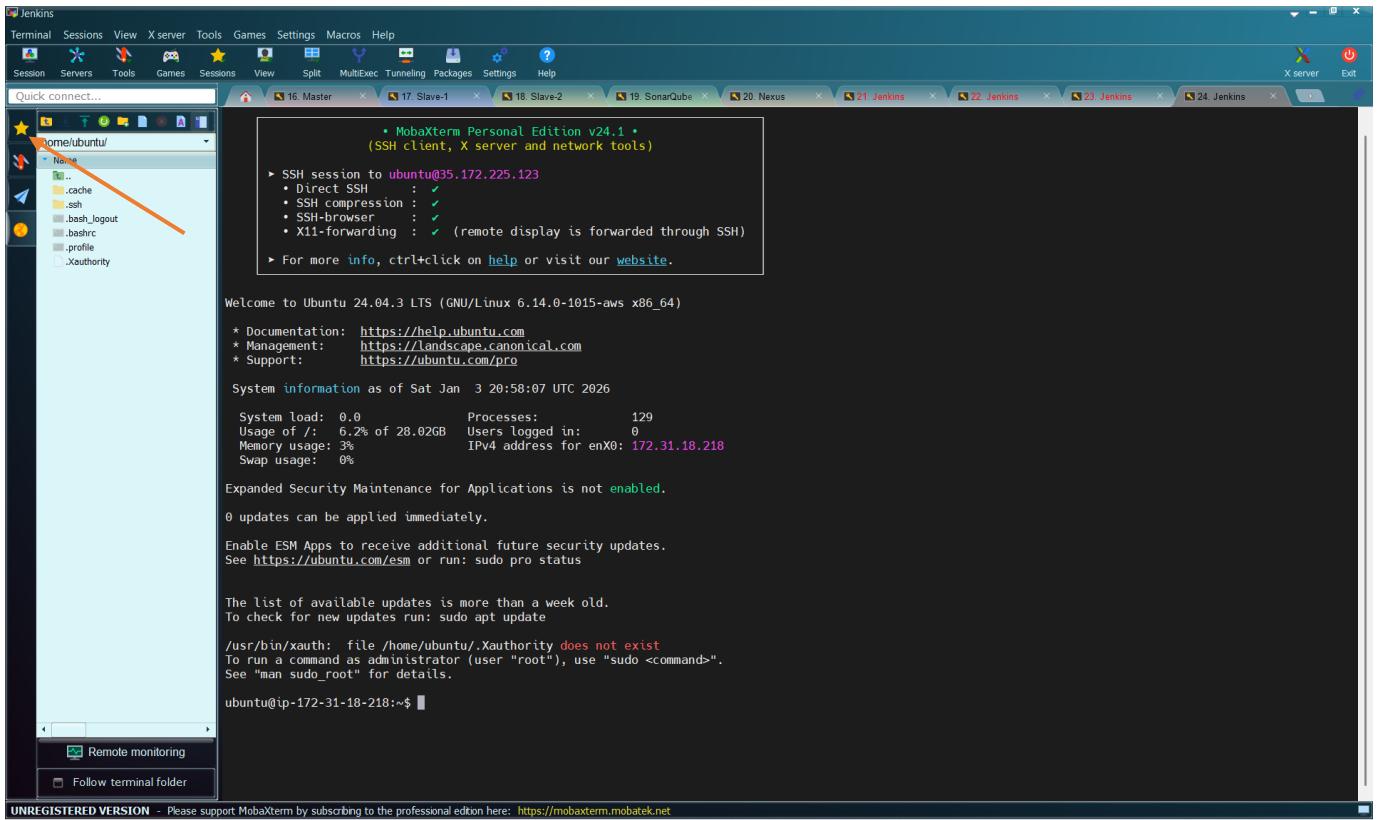
And click on “OK” again



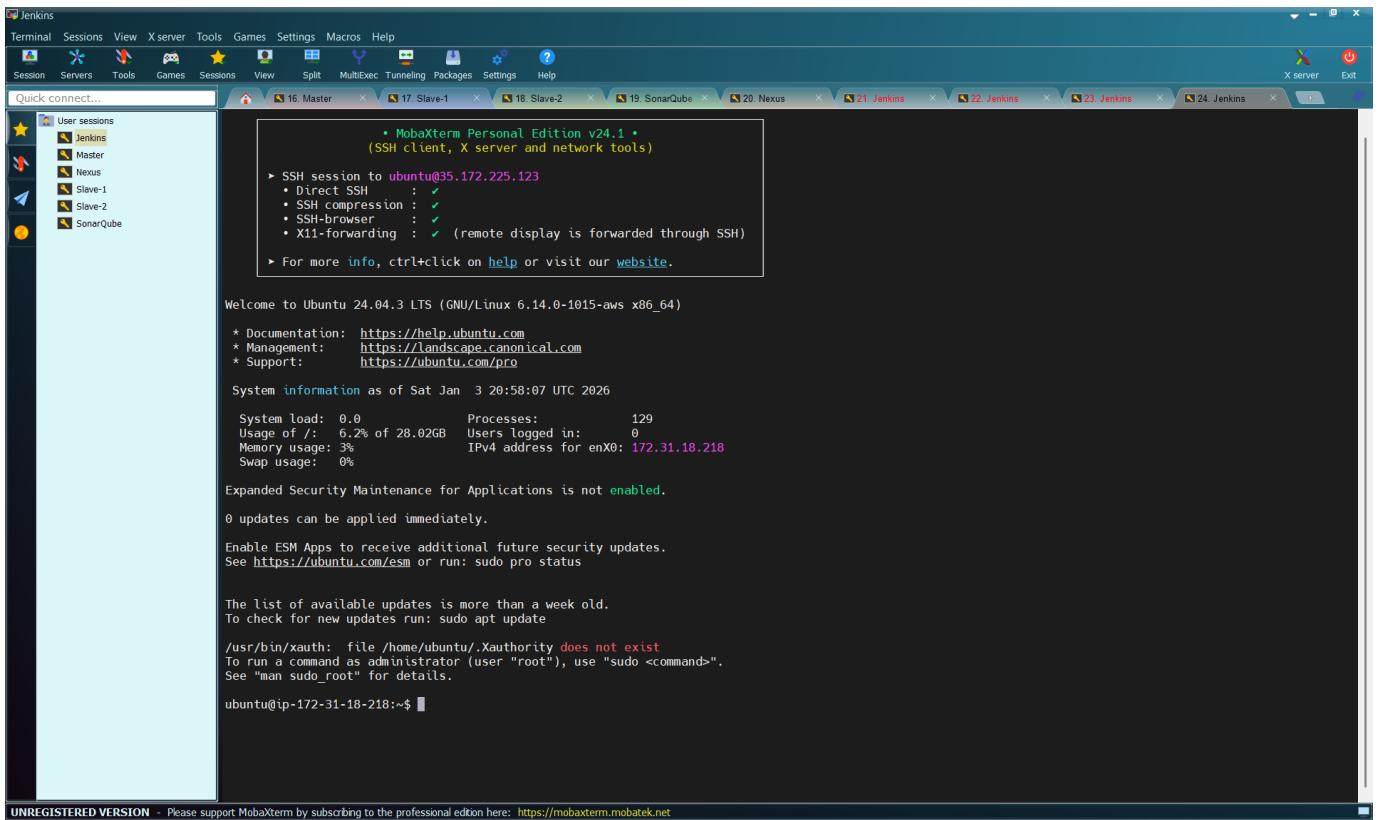
Then, double-click on the session “Jenkins”



Click on “Accept”

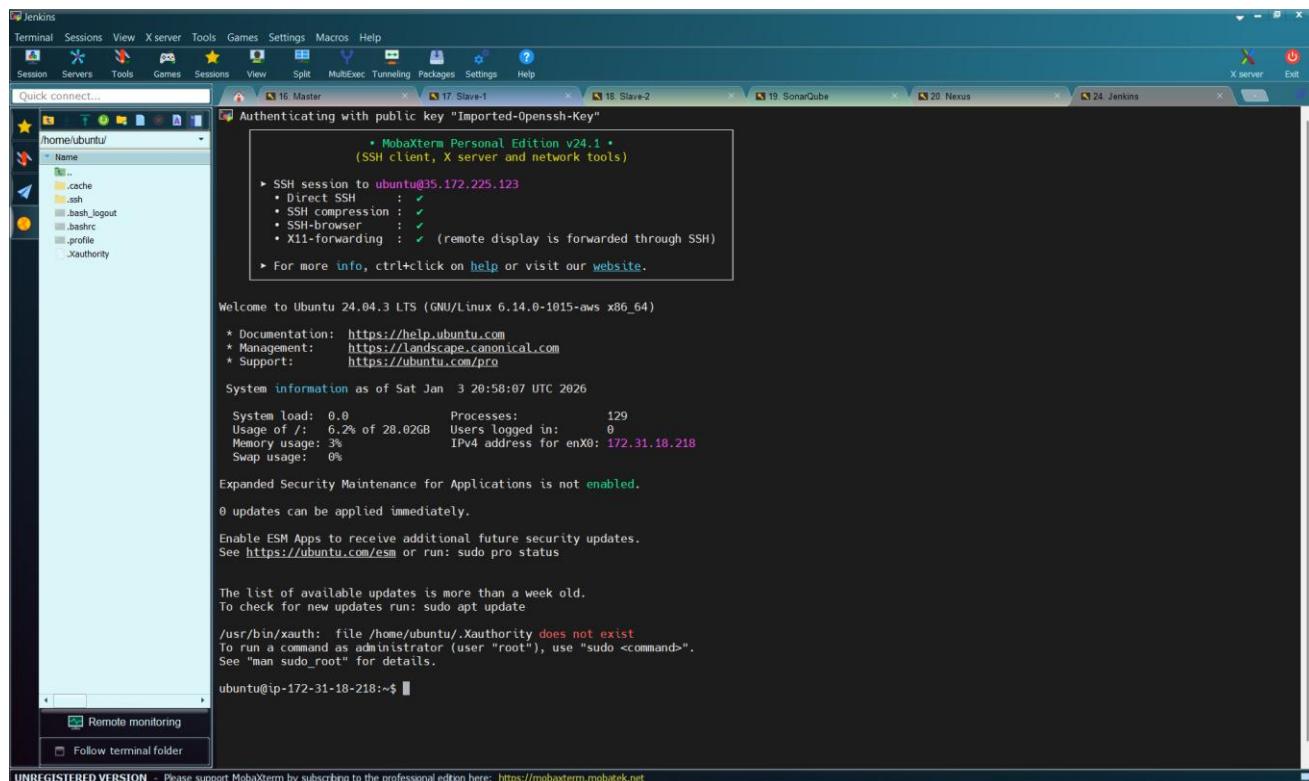


You can see that the tab of “Jenkins” is “Grey”. Click on the star



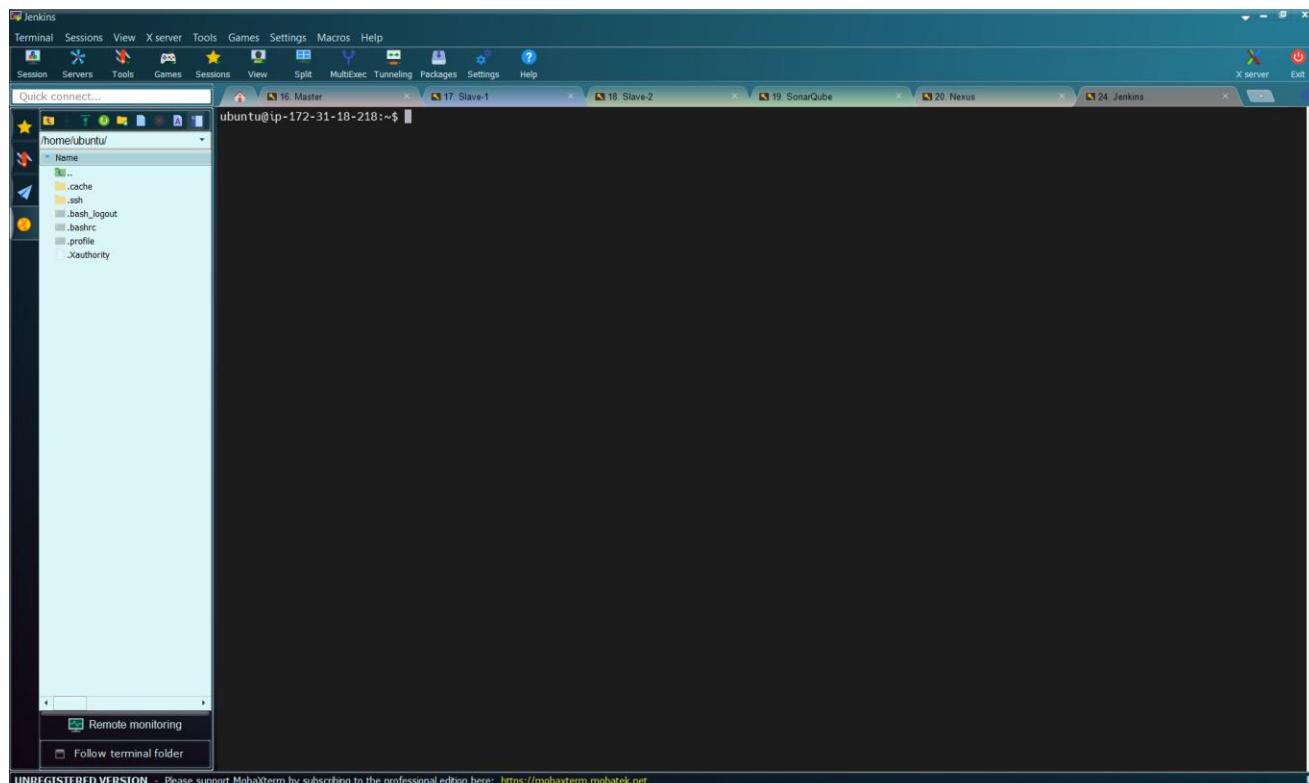
### 3.7.3 Configure Jenkins

Let us set up Jenkins.



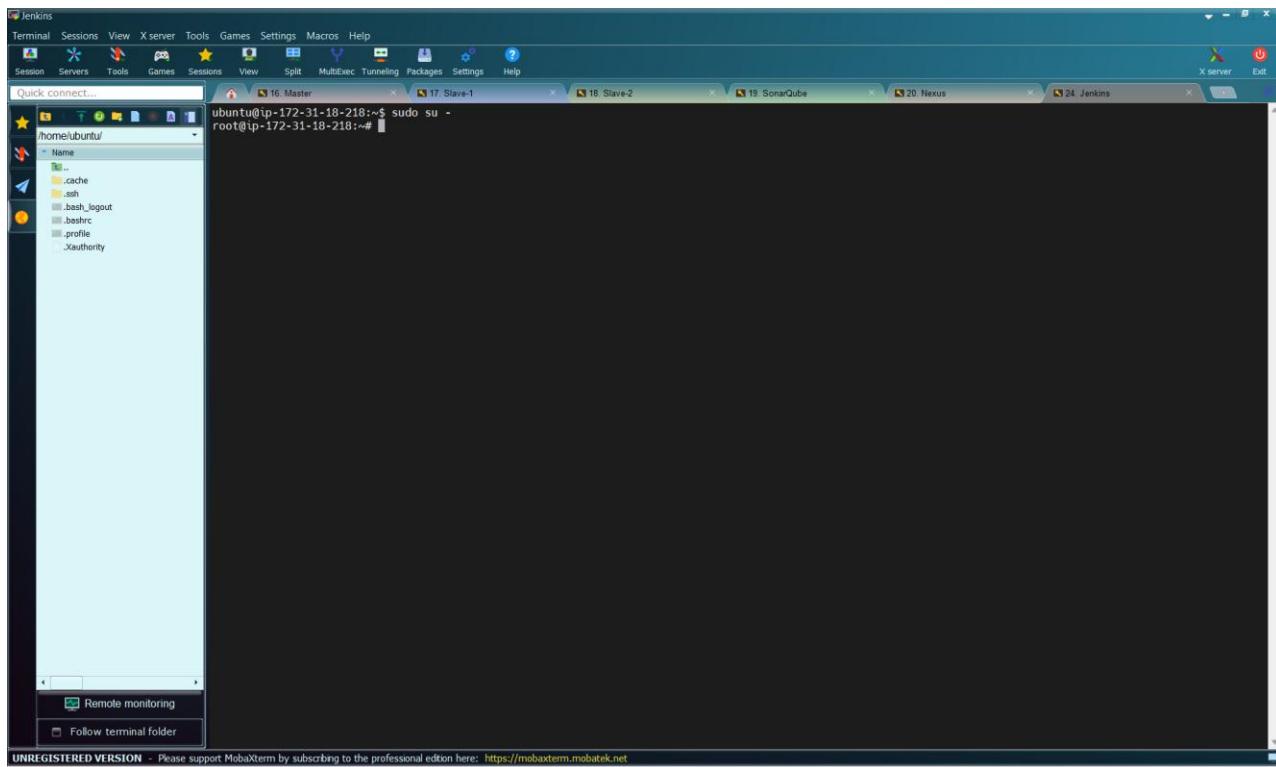
Run the command to clear the terminal:

```
clear
```



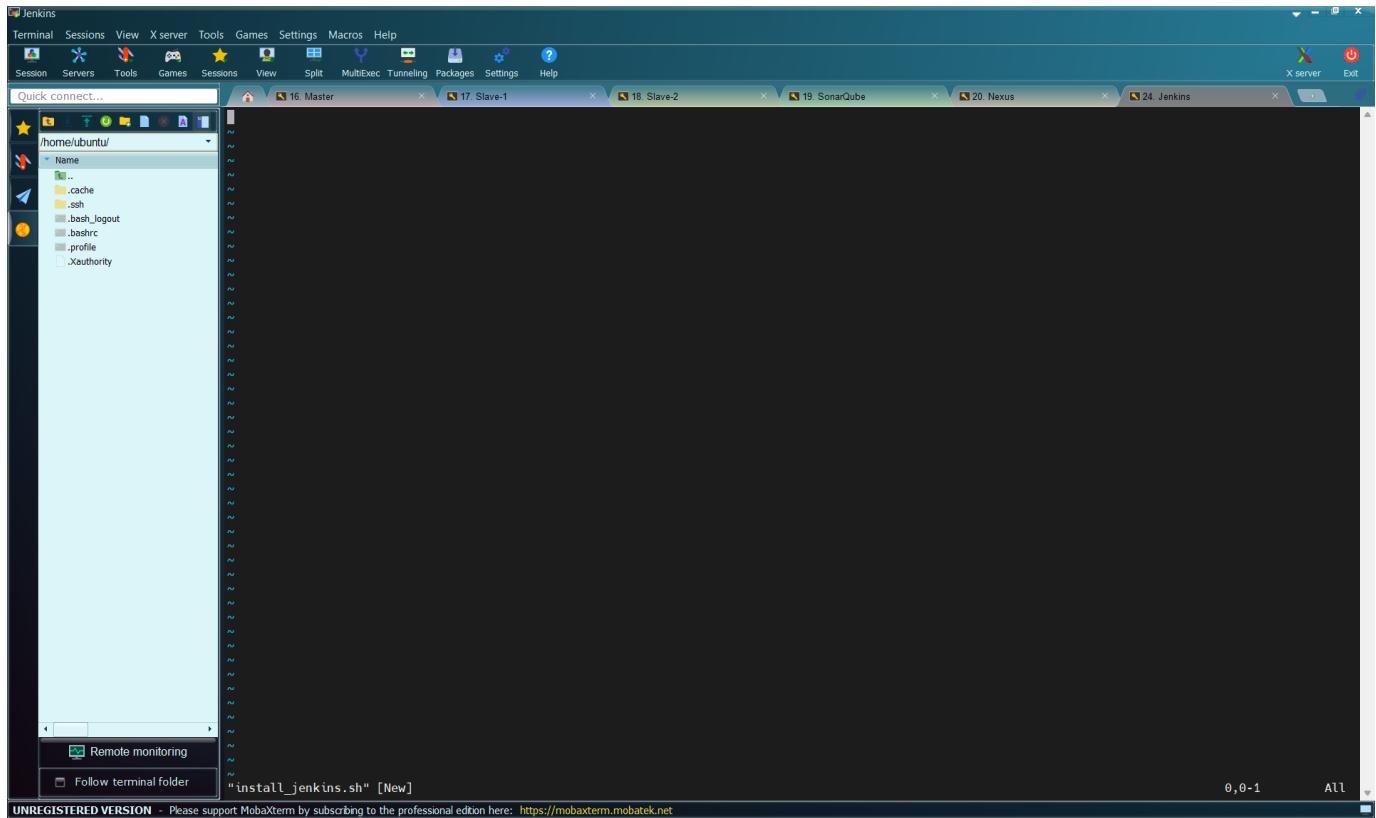
Give the Jenkins server the root user privilege by running the command:

```
sudo su -
```



The prerequisite for Jenkins to run, we need Java. We will install Java and Jenkins on the Jenkins server using a shell script. Let us create a script called "**install\_jenkins.sh**" by using the command:

```
vi install_jenkins.sh
```



Then, copy and paste the code below to install Jenkins here

```
#!/bin/bash

# Install OpenJDK 17 JRE Headless
sudo apt install openjdk-17-jre-headless -y

# Download Jenkins GPG key
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

# Add Jenkins repository to package manager sources
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null

# Update package manager repositories
sudo apt-get update

# Install Jenkins
sudo apt-get install jenkins -y
```

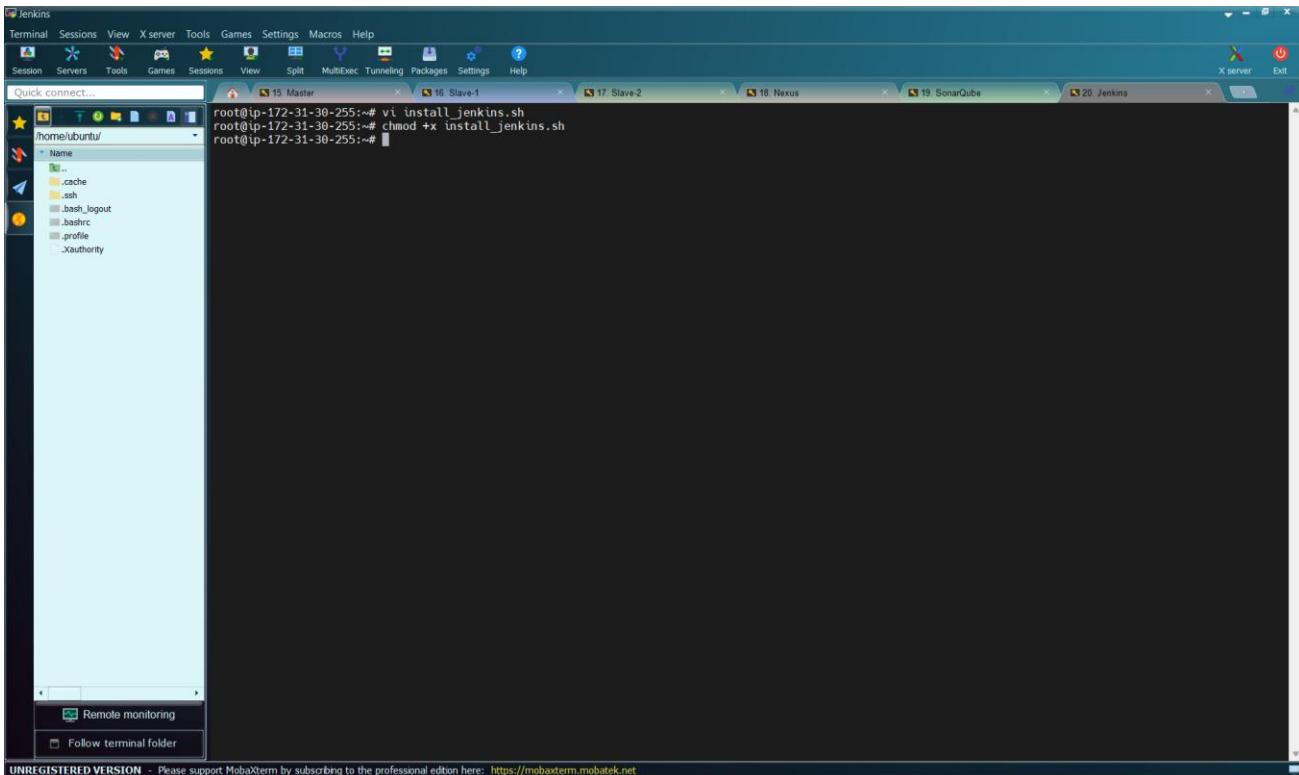
```
#!/bin/bash
# Install OpenJDK 17 JRE Headless
sudo apt install openjdk-17-jre-headless -y
# Download Jenkins GPG key
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
# Add Jenkins repository to package manager sources
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
# Update package manager repositories
sudo apt-get update
# Install Jenkins
sudo apt-get install jenkins -y
```

Save and exit the file by pressing “**ESC**” followed by (**:wq**) and press “**Enter**”

```
root@ip-172-31-30-255:~# vi install_jenkins.sh
```

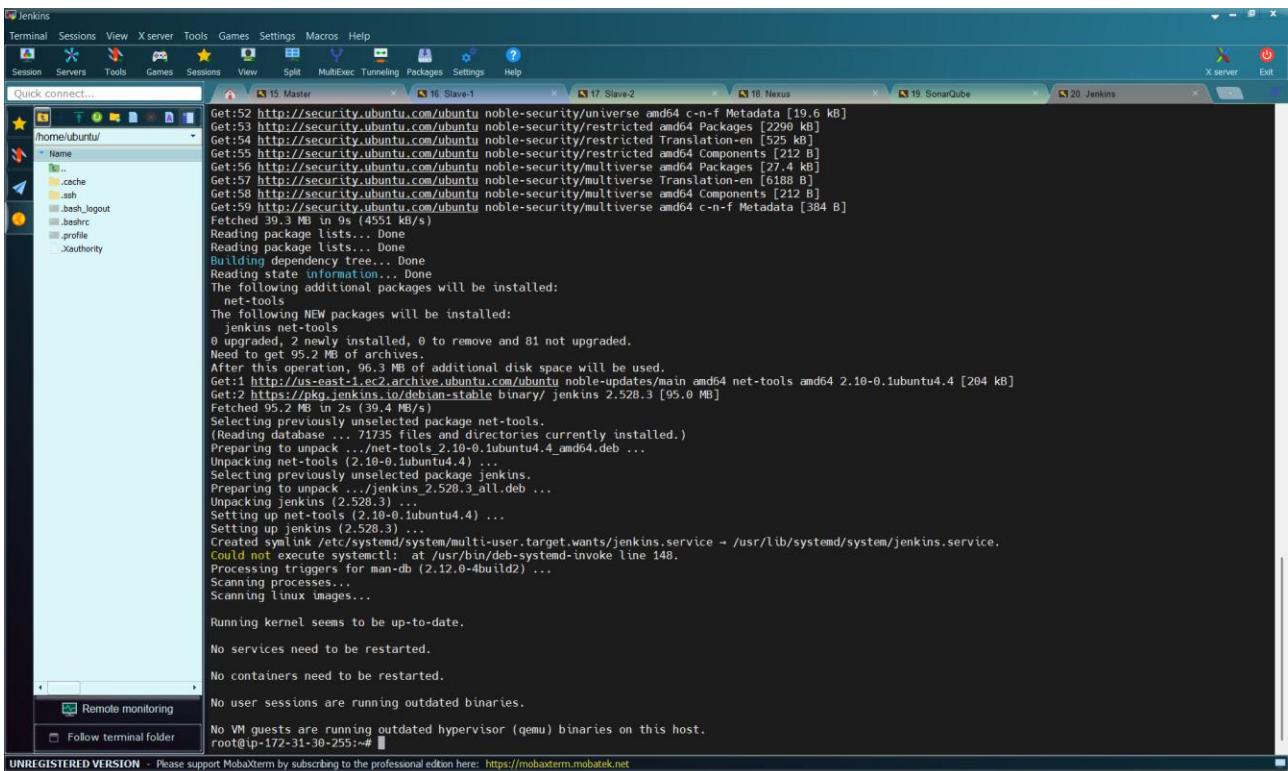
Make the file executable using the command:

```
chmod +x install_jenkins.sh
```



Then, run the script using the command:

```
./install_jenkins.sh
```

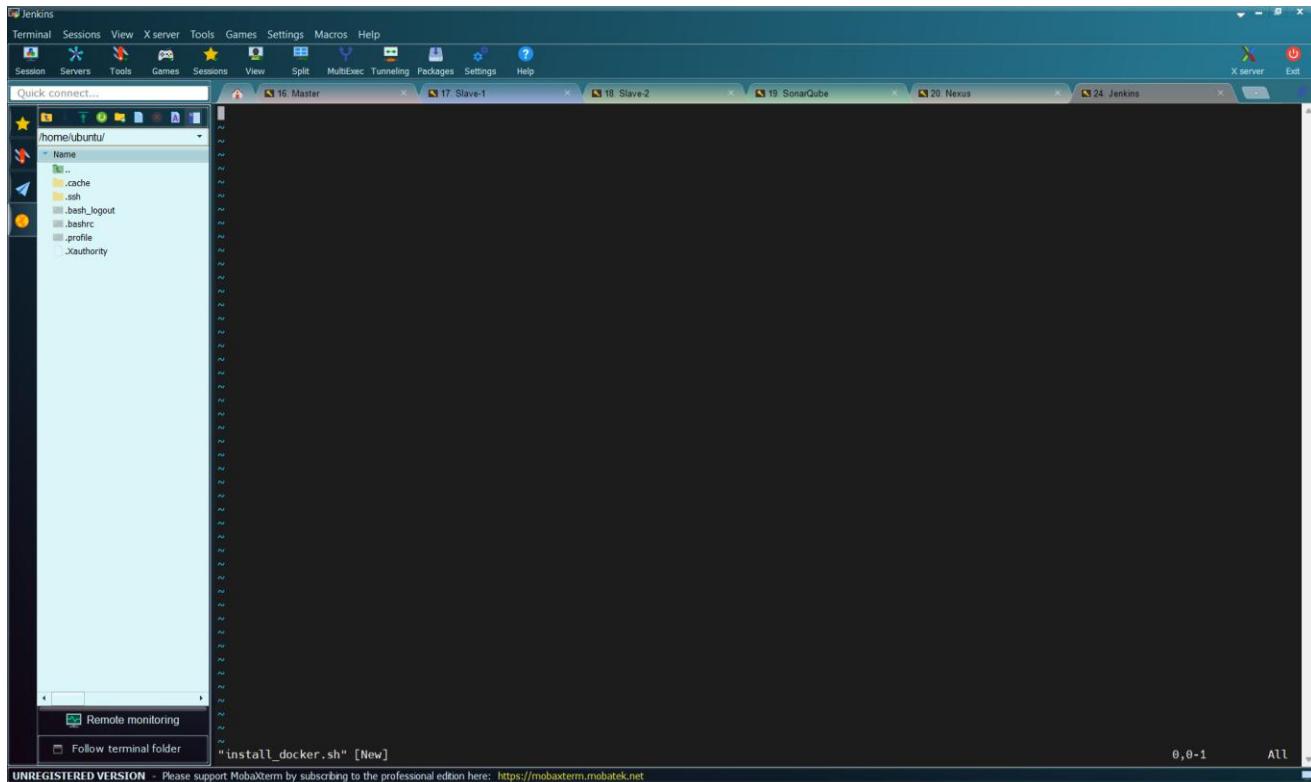


We have installed Jenkins on the Jenkins server.

## Install Docker on Jenkins Server

Let us now install Docker. We will also use a script to install Docker. We will call the script “**install\_docker.sh**”. Let us create the file using the command:

```
vi install_docker.sh
```



Copy and paste the below code to install Docker here

```
#!/bin/bash

# Update package manager repositories
sudo apt-get update

# Install necessary dependencies
sudo apt-get install -y ca-certificates curl

# Create directory for Docker GPG key
sudo install -m 0755 -d /etc/apt/keyrings

# Download Docker's GPG key
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

# Ensure proper permissions for the key
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add Docker repository to Apt sources
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Update package manager repositories
sudo apt-get update

sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```

#!/bin/bash

# Update package manager repositories
sudo apt-get update

# Install necessary dependencies
sudo apt-get install -y ca-certificates curl

# Create directory for Docker GPG key
sudo mkdir -m 0755 -d /etc/apt/keyrings

# Download Docker's GPG key
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor > /etc/apt/keyrings/docker.asc

# Ensure proper permissions for the key
sudo chmod afr /etc/apt/keyrings/docker.asc

# Add Docker repository to Apt sources
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$ . /etc/os-release && echo \"$VERSION_CODENAME\" stable" | \ 
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Update package manager repositories
sudo apt-get update

sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

```

The terminal window shows the command history and output of the Docker installation script. The script installs Docker CE, CLI, containerd.io, buildx-plugin, and compose-plugin. The terminal window has tabs for Jenkins, Slave-2, Slave-1, SonarQube, Nexus, and Jenkins.

Save and exit the file by pressing “**ESC**” followed by (**:wq**) and press “**Enter**”

```

root@ip-172-31-30-255:~# vi install_docker.sh
root@ip-172-31-30-255:~#

```

The terminal window shows the command 'vi install\_docker.sh' being run. The status bar indicates 'UNREGISTERED VERSION'. The terminal window has tabs for Jenkins, Slave-2, Slave-1, SonarQube, Nexus, and Jenkins.

Then, make it executable using the command:

```
chmod +x install_docker.sh
```

```

root@ip-172-31-30-255:~# vi install_docker.sh
root@ip-172-31-30-255:~# chmod +x install_docker.sh
root@ip-172-31-30-255:~#

```

Then, run the script using this command:

```
./install_docker.sh
```

```

Selecting previously unselected package docker-ce.
Preparing to unpack .../2-docker-ce_5%3a29.1.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package pigz.
Preparing to unpack .../3-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../4-docker-buildx-plugin_0.30.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a29.1.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_5.0.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../7-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../8-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (2.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Setting up docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu0.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-30-255:~#

```

Docker has been installed, let us run the command to give access to other users:

```
sudo chmod 666 /var/run/docker.sock
```

```

Preparing to unpack .../2-docker-ce_5%3a29.1.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package pigz.
Preparing to unpack .../3-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../4-docker-buildx-plugin_0.30.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a29.1.4-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_5.0.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../7-libslirp0_4.7.0~ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0~ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../8-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.30.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (2.1.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service -> /usr/lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (5.0.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0~ubuntu3) ...
Setting up libslirp0:amd64 (4.7.0~ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:29.1.4-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service -> /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket -> /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4ubuntu0.6) ...
Processing triggers for libc-bin (2.39.0-ubuntu0.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-30-255:~# sudo chmod 666 /var/run/docker.sock
root@ip-172-31-30-255:~#

```

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

Let us run the command to verify that Docker is running:

```
sudo systemctl status docker
```

```

root@ip-172-31-30-255:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Sun 2026-01-11 18:53:30 UTC; 1min 5s ago
     TriggeredBy: docker@socket
       Docs: https://docs.docker.com
 Main PID: 3283 (dockerd)
   Tasks: 9
  Memory: 26.0M (peak: 26.5M)
    CPU: 293ms
   CGroup: /system.slice/docker.service
           └─3283 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jan 11 18:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T18:53:30.261595952Z" level=info msg="Restoring containers: start."
Jan 11 18:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T18:53:30.293615990Z" level=info msg="Deleting nftables IPv4 rules" error="exit status 1"
Jan 11 18:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T18:53:30.298705440Z" level=info msg="Deleting nftables IPv6 rules" error="exit status 1"
Jan 11 18:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T18:53:30.588626159Z" level=info msg="Docker daemon" commit=08440b6 containerd-snapshotter=true
Jan 11 18:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T18:53:30.597943028Z" level=info msg="Loading containers: done."
Jan 11 18:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T18:53:30.598859381Z" level=info msg="Initializing buildkit"
Jan 11 18:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T18:53:30.633451447Z" level=info msg="Completed buildkit initialization"
Jan 11 18:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T18:53:30.637245678Z" level=info msg="Daemon has completed initialization"
Jan 11 18:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T18:53:30.637392218Z" level=info msg="API listen on /run/docker.sock"
Jan 11 18:53:30 ip-172-31-30-255 systemd[1]: Started docker.service - Docker Application Container Engine.
[lines 1-22/22 (END)]

```

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

Then, exit to command mode by running the command:

```
Ctrl+c
```

```

root@ip-172-31-30-255:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Sun 2026-01-11 10:53:30 UTC; 1min 5s ago
     Docs: https://docs.docker.com
      Main PID: 3283 (dockerd)
        Tasks: 9
       Memory: 26.0M (peak: 26.5M)
          CPU: 293ms
        CGroup: /system.slice/docker.service
                  └─3283 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Jan 11 10:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T10:53:30.261595052Z" level=info msg="Restoring containers: start."
Jan 11 10:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T10:53:30.293615990Z" level=info msg="Deleting nftables IPv4 rules" error="exit status 1"
Jan 11 10:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T10:53:30.298705440Z" level=info msg="Deleting nftables IPv6 rules" error="exit status 1"
Jan 11 10:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T10:53:30.579430280Z" level=info msg="Loading containers: done."
Jan 11 10:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T10:53:30.588626159Z" level=info msg="Docker daemon" commit=08440b6 containerd-snapshotter=tr
Jan 11 10:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T10:53:30.588850381Z" level=info msg="Initializing buildkit"
Jan 11 10:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T10:53:30.633451447Z" level=info msg="Completed buildkit initialization"
Jan 11 10:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T10:53:30.637245678Z" level=info msg="Daemon has completed initialization"
Jan 11 10:53:30 ip-172-31-30-255 dockerd[3283]: time="2026-01-11T10:53:30.637292218Z" level=info msg="API listen on /run/docker.sock"
Jan 11 10:53:30 ip-172-31-30-255 systemd[1]: Started docker.service - Docker Application Container Engine.

root@ip-172-31-30-255:~#

```

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

We have installed Jenkins and Docker.

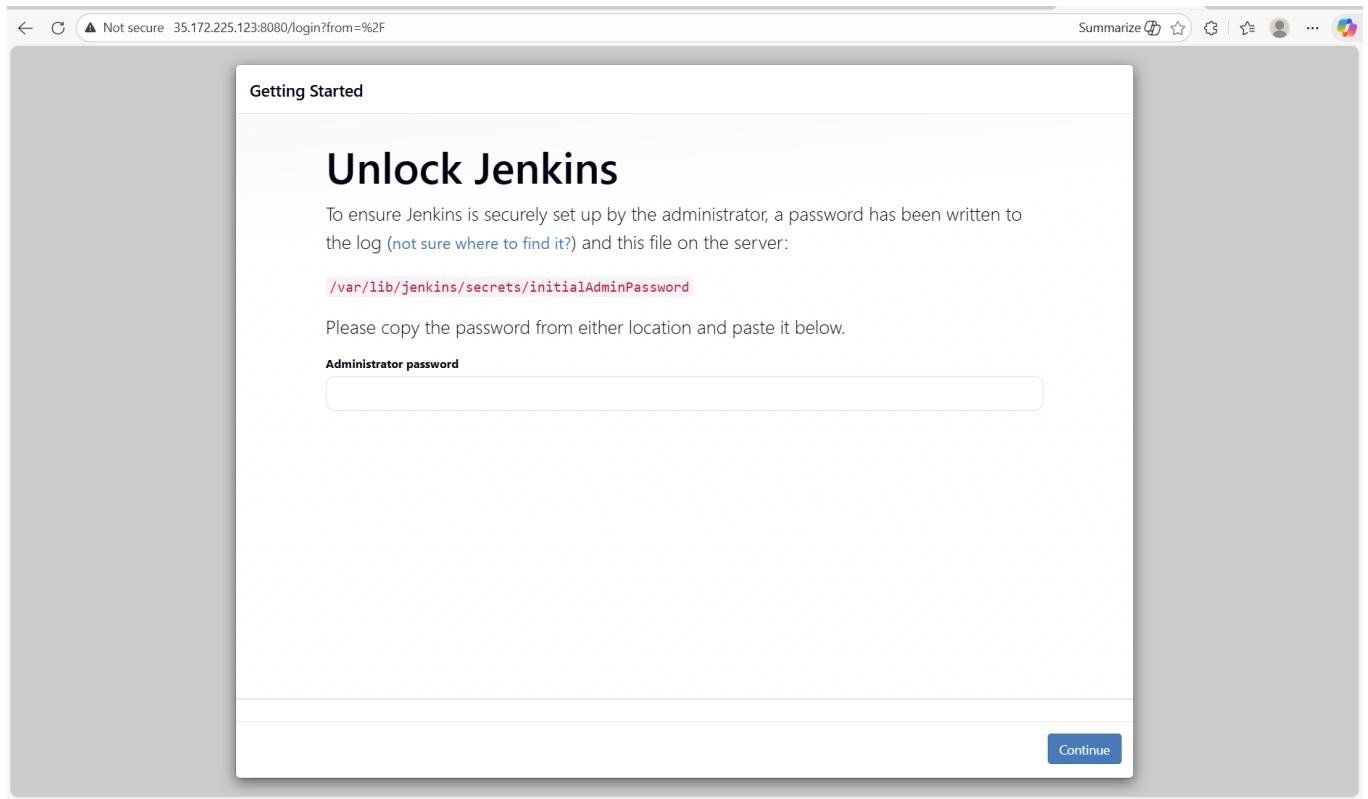
### 3.7.4 Accessing Jenkins through Browser

Let us access Jenkins through the browser by using the Public IP address of the Jenkins server. This is done as follows:

`http://<Public IP of Jenkins Server>:8080`

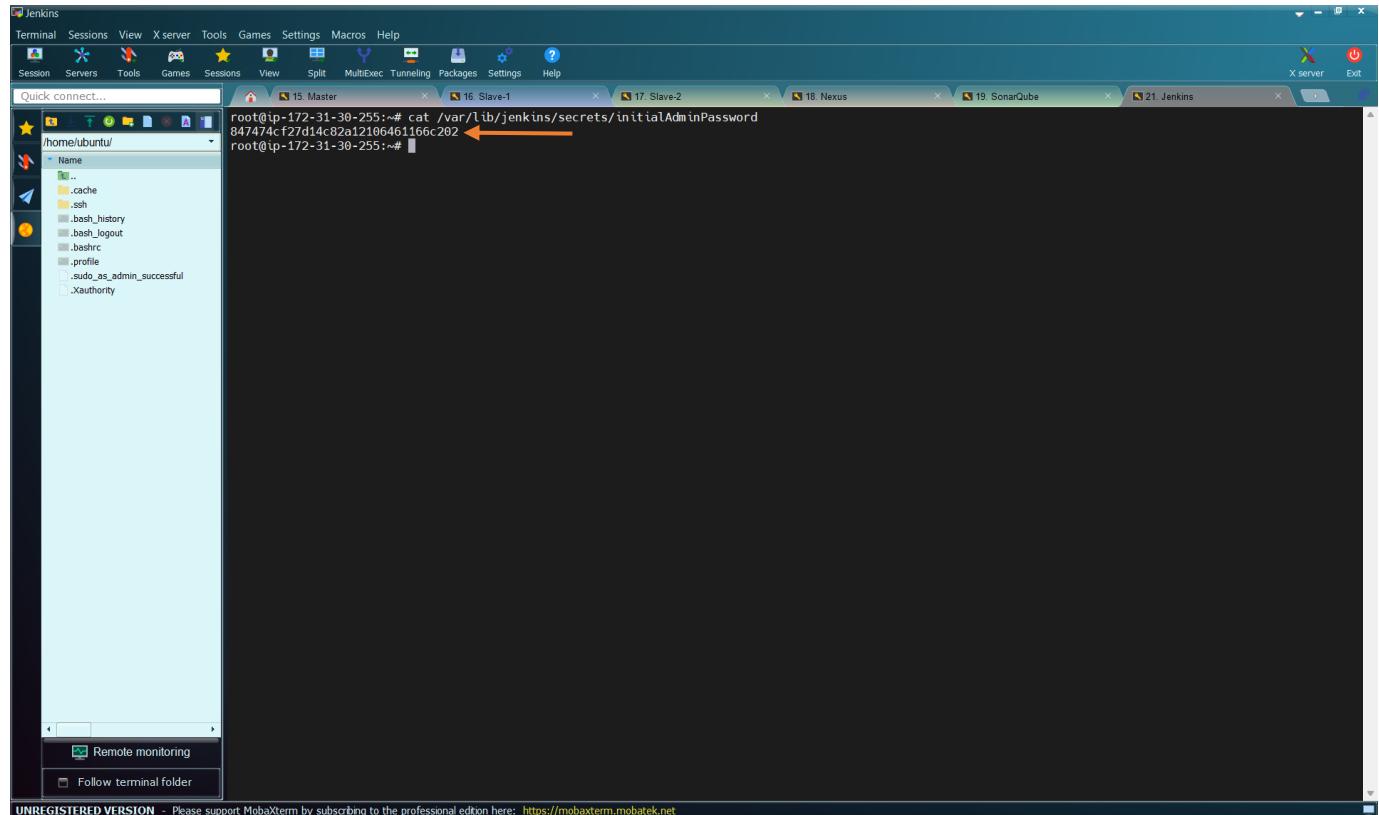
That is

`http://54.226.220.79:8080`



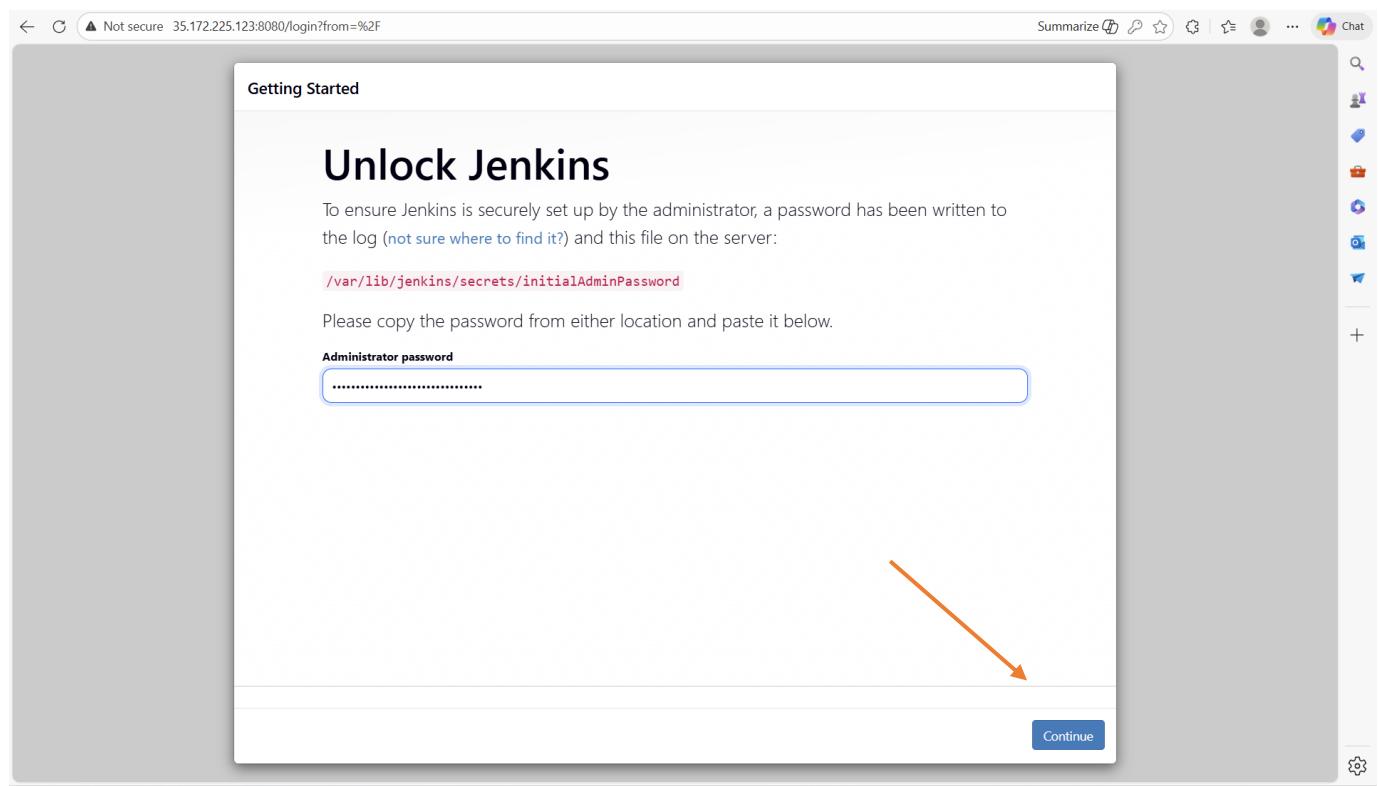
We have to get the Administrator password. To do this we run this command on Jenkins server terminal:

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

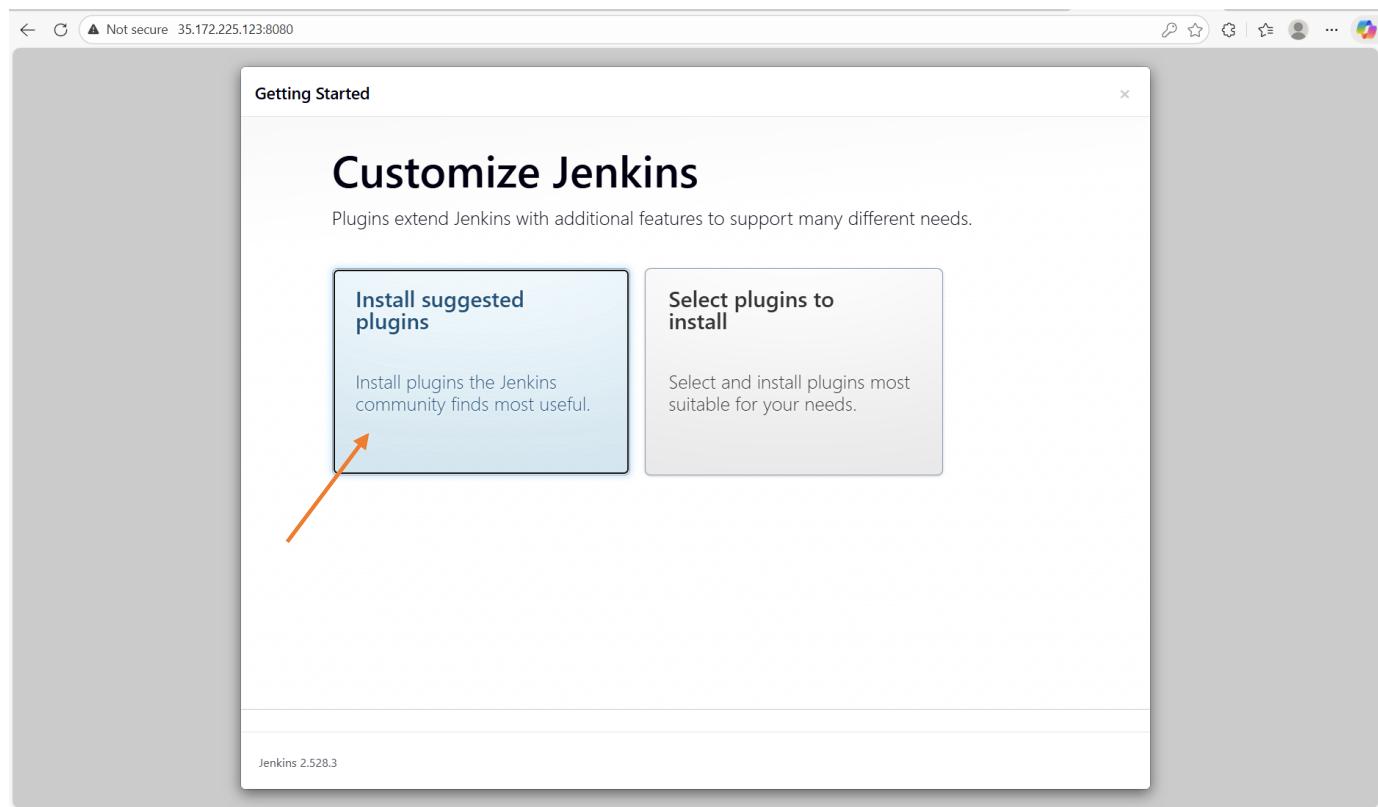


Copy this password and paste on the Jenkins browser

847474cf27d14c82a12106461166c202



Click on “Continue”



Click on “Install Suggested Plugins”

Not secure 35.172.225.123:8080

## Create First Admin User

Username

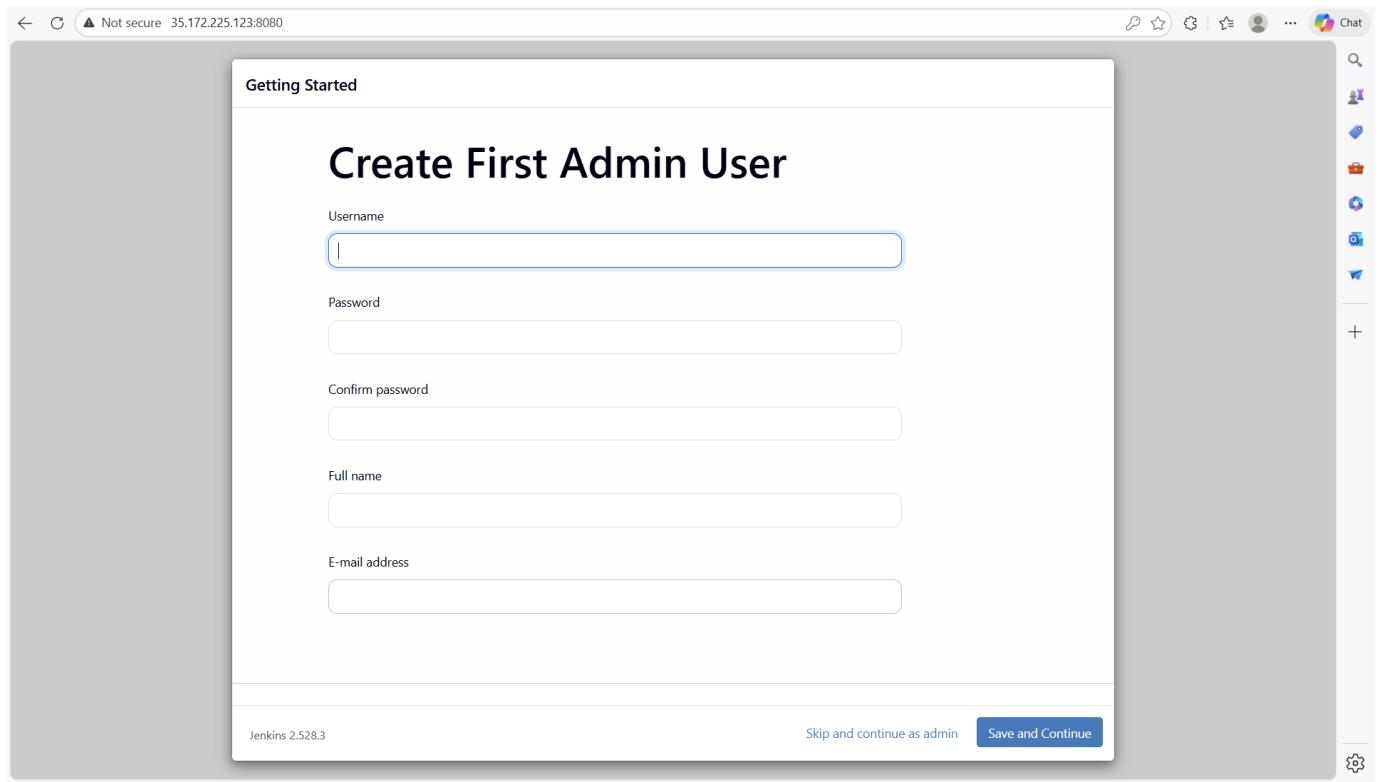
Password

Confirm password

Full name

E-mail address

Jenkins 2.528.3 [Skip and continue as admin](#) [Save and Continue](#)



Complete the above information. My username will be “**ebotsmith**”

Not secure 35.172.225.123:8080

## Create First Admin User

Username

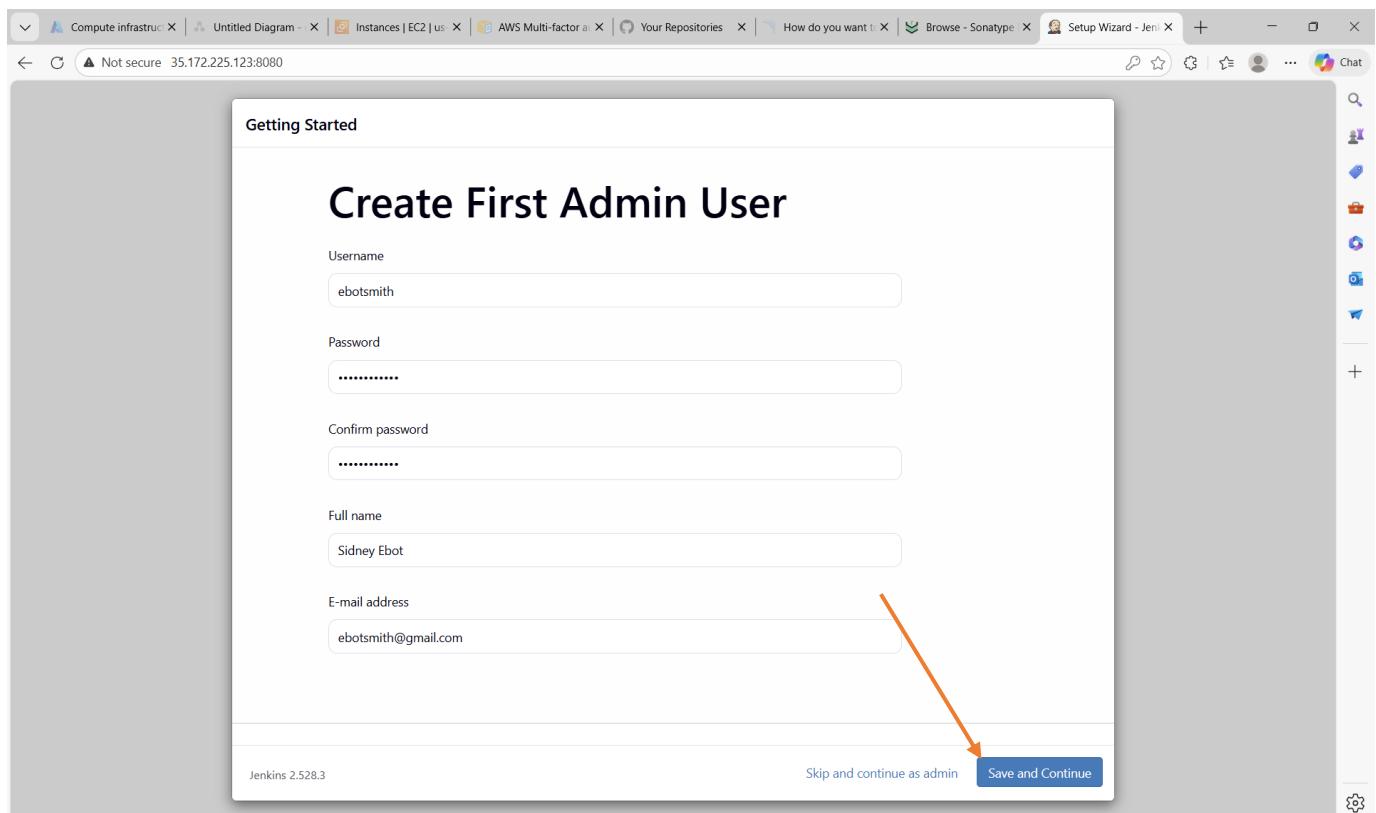
Password

Confirm password

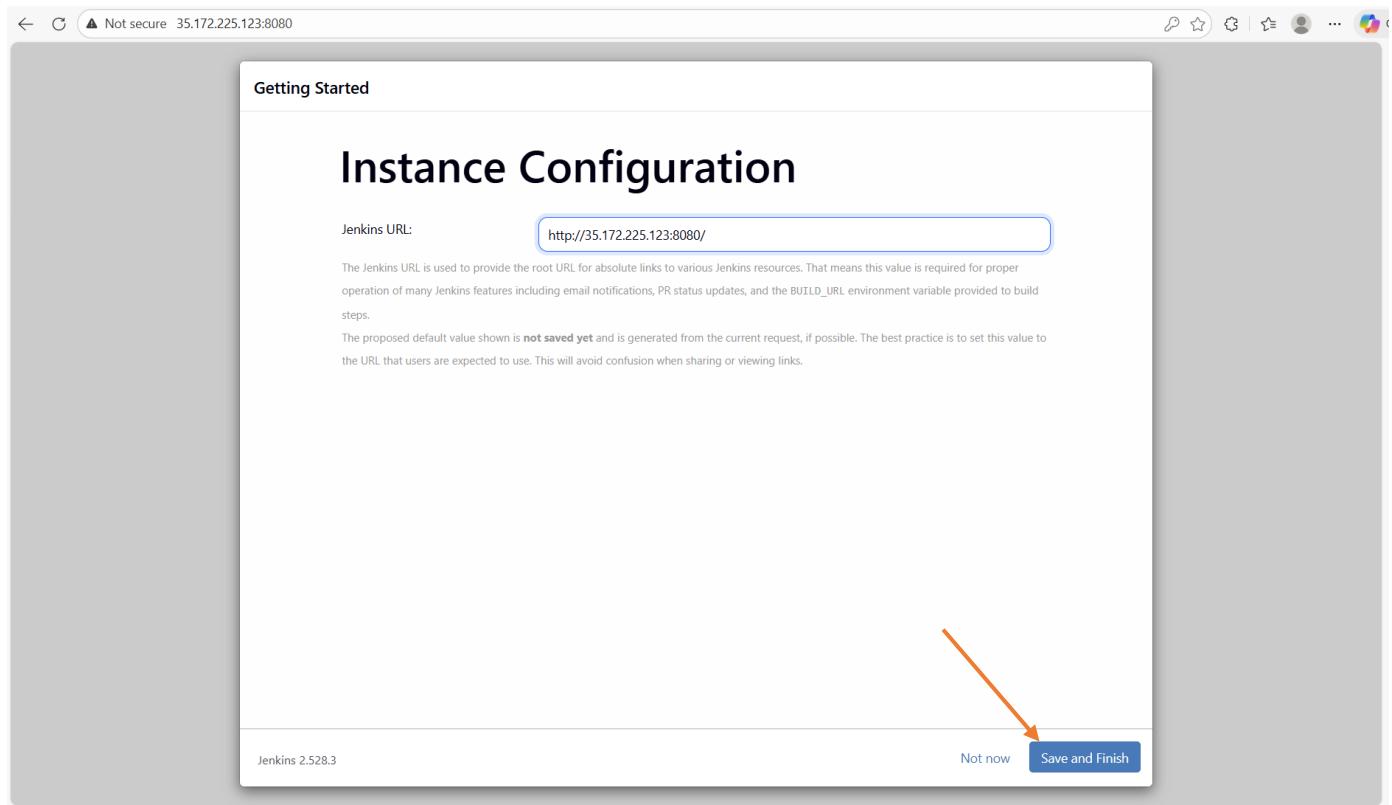
Full name

E-mail address

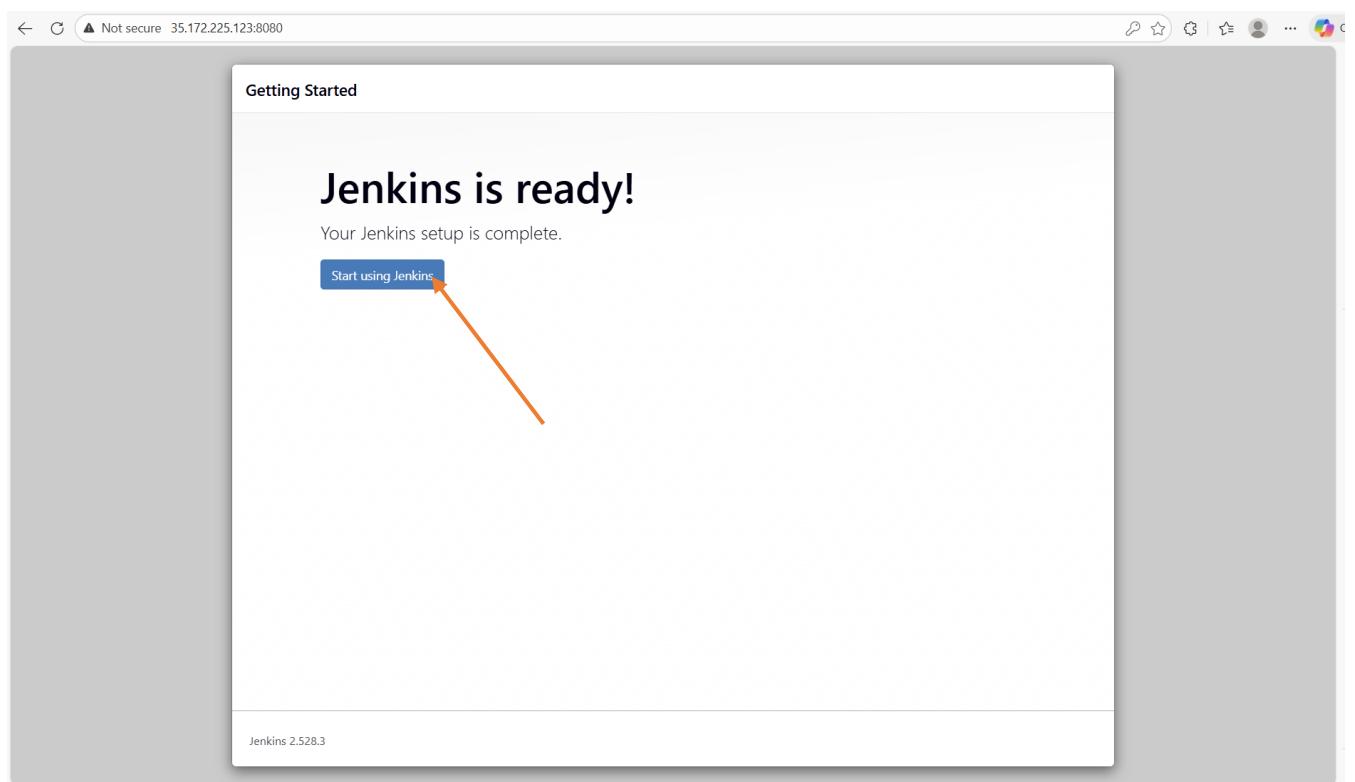
Jenkins 2.528.3 [Skip and continue as admin](#) [Save and Continue](#)



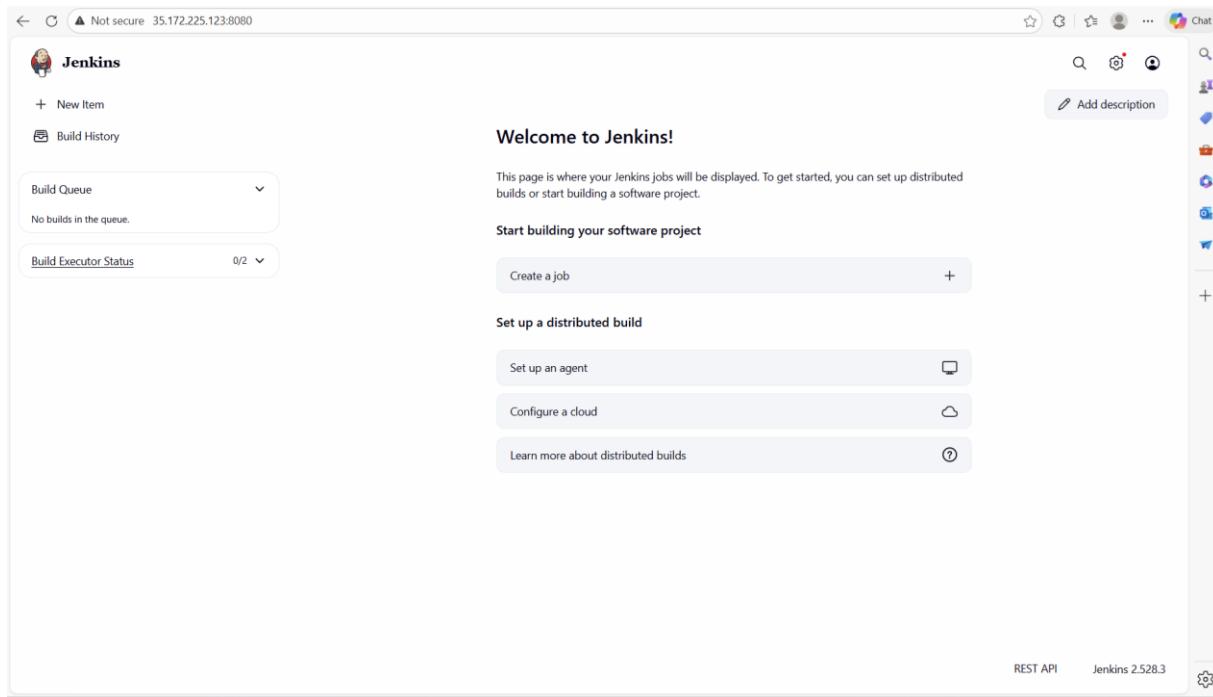
Click on “**Save and Continue**”



Click on “Save and Finish”



Then, click on “Start using Jenkins”

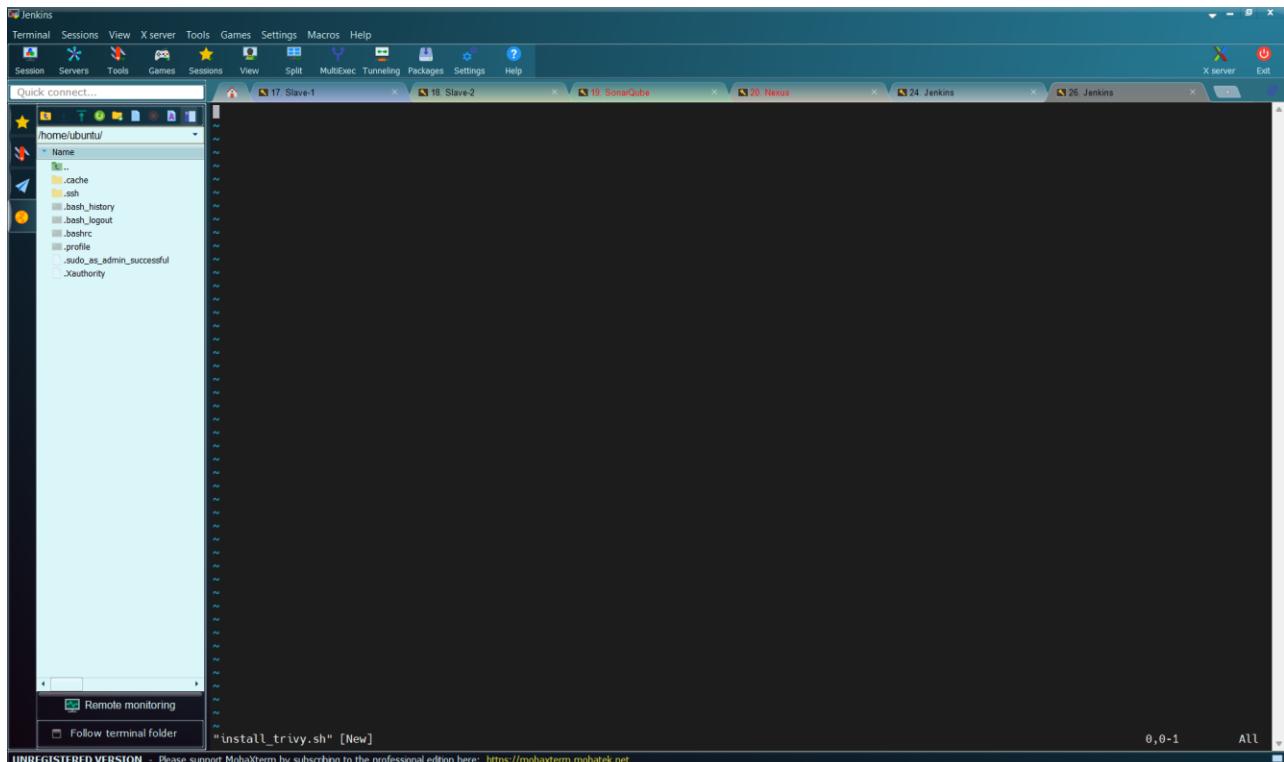


Jenkins has been set up. We have completed phase one.

### 3.7.5 Install Trivy on Jenkins Server

We have to install Trivy on Jenkins server. We will do this by using a script called “`install_trivy.sh`”. Let us create the file using the command:

```
vi install_trivy.sh
```



Copy and paste the code below that will install Trivy

```
#!/bin/bash

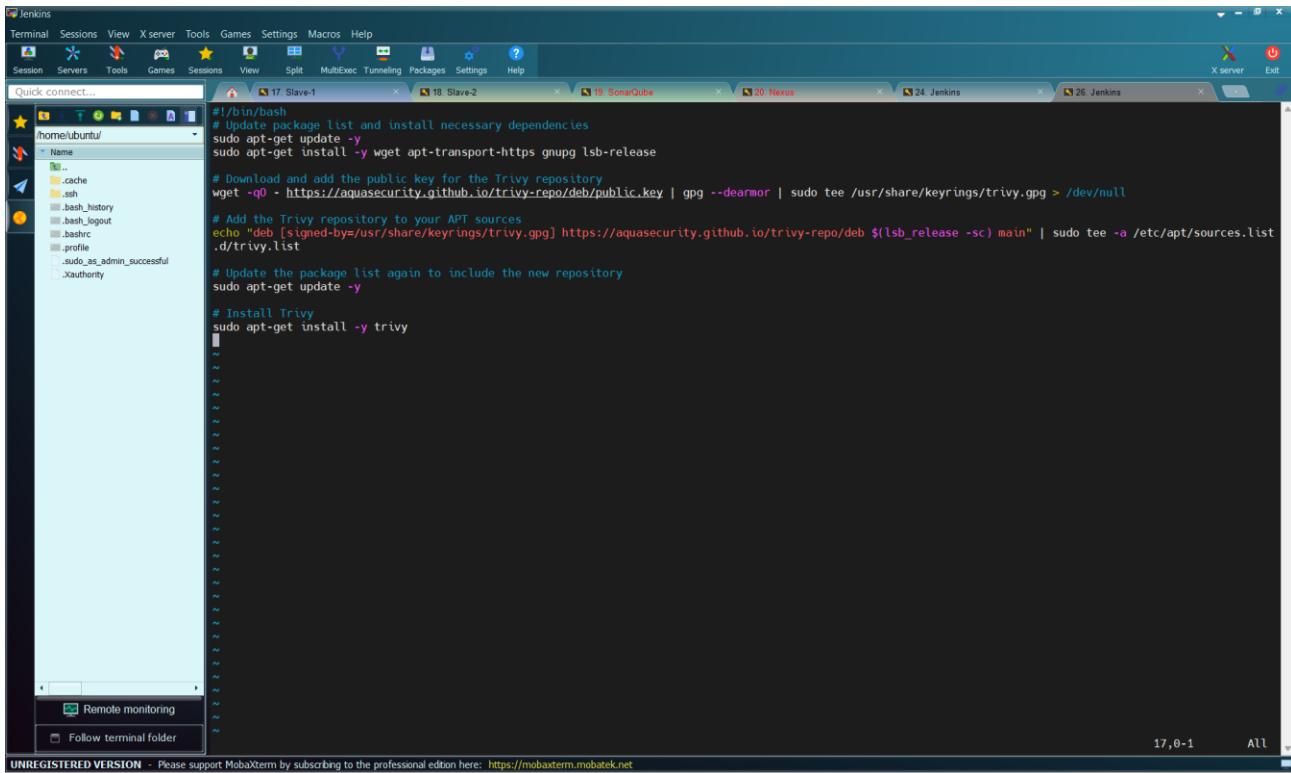
# Update package list and install necessary dependencies
sudo apt-get update -y
sudo apt-get install -y wget apt-transport-https gnupg lsb-release

# Download and add the public key for the Trivy repository
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null

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

# Update the package list again to include the new repository
sudo apt-get update -y

# Install Trivy
sudo apt-get install -y trivy
```



The screenshot shows a MobaXterm window titled 'Jenkins' with multiple tabs open. The active tab displays the Trivy installation script. The script includes comments explaining each step: updating dependencies, downloading the Trivy GPG key, adding the Trivy repository to /etc/apt/sources.list.d/trivy.list, updating the package list again, and finally installing Trivy. The terminal window also shows the file structure of the user's home directory (~/.cache, .ssh, .bash\_history, .bash\_logout, .bashrc, .profile, .sudo\_as\_admin\_successful, .Xauthority).

Save and Exit the script by pressing “**ESC**” followed by typing :**wq** and press “**Enter**”

```
root@ip-172-31-18-218:~# vi install_trivy.sh
```

Then, change the permission to executable file

```
chmod +x install_trivy.sh
```

```
root@ip-172-31-18-218:~# vi install_trivy.sh
root@ip-172-31-18-218:~# chmod +x install_trivy.sh
root@ip-172-31-18-218:~#
```

Then, execute the script using the command:

```
./install_trivy.sh
```

```

Jenkins
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
17. Slave-1 18. Slave-2 19. SonarQube 20. Nexus 24. Jenkins 26. Jenkins X server Exit

/home/ubuntu/
Name
.. .
.. cache
.. ssh
.. bash_history
.. bashrc
.. profile
.. sudo_is_admin_successful
.. xauthority

Get:7 https://aquasecurity.github.io/trivy-repo/deb noble InRelease [3061 B]
Hit:8 http://security.ubuntu.com/ubuntu noble=security InRelease
Fetched 3426 B in 1s (5788 B/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  trivy
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 47.5 MB of archives.
After this operation, 159 MB of additional disk space will be used.
Get:1 https://aquasecurity.github.io/trivy-repo/deb noble/main amd64 Packages [368 B]
Fetched 47.5 MB in 0s (97.5 MB/s)
Selecting previously unselected package trivy.
(Reading database ... 105012 files and directories currently installed.)
Preparing to unpack .../trivy_0.68.2_amd64.deb ...
Unpacking trivy (0.68.2) ...
Setting up trivy (0.68.2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.14.0-1015-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.14.0-1018-aws.
Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.
Restarting services...

Service restarts being deferred:
/etc/needrestart/restart.d/dhclient.service
systemctl restart getty@.service
systemctl restart networkd-dispatcher.service
systemctl restart serial-getty@ttyS0.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-18-218:~# 

```

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

Then, verify of Trivy has been installed using the command:

`trivy --version`

```

Master
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
16. Master 17. Slave-1 18. Slave-2 19. SonarQube 20. Nexus 24. Jenkins X server Exit

User sessions
Jenkins
Master
Nexus
Slave-1
Slave-2
SonarQube

Hit:8 https://prod-cdn.packages.k8s.io/repos/llvm/kubernetes/core/stable/v1.29/deb InRelease
Hit:9 http://security.ubuntu.com/ubuntu noble=security InRelease
Get:10 https://aquasecurity.github.io/trivy-repo/deb noble/main amd64 Packages [368 B]
Reading package lists... Done
E: The repository 'https://aquasecurity.github.io/trivy-repo/deb Release' does not have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8)' manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  trivy
0 upgraded, 1 newly installed, 0 to remove and 49 not upgraded.
Need to get 47.5 MB of archives.
After this operation, 159 MB of additional disk space will be used.
Get:1 https://aquasecurity.github.io/trivy-repo/deb noble/main amd64 trivy amd64 0.68.2 [47.5 MB]
Fetched 47.5 MB in 0s (97.5 MB/s)
Selecting previously unselected package trivy.
(Reading database ... 103545 files and directories currently installed.)
Preparing to unpack .../trivy_0.68.2_amd64.deb ...
Unpacking trivy (0.68.2) ...
Setting up trivy (0.68.2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.14.0-1015-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.14.0-1018-aws.
Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.
Restarting services...

Service restarts being deferred:
systemctl restart networkd-dispatcher.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-22-65:~# trivy --version
Version: 0.68.2
root@ip-172-31-22-65:~# 

```

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

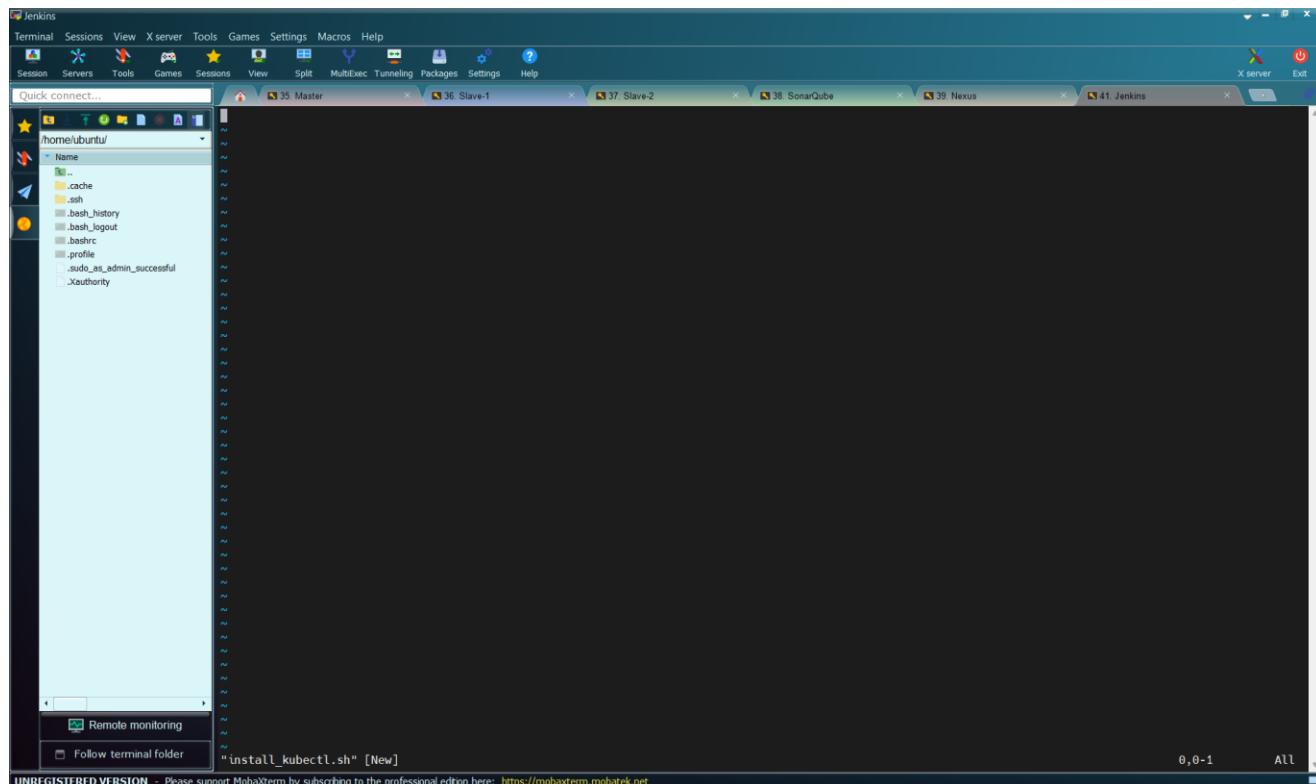
We have installed Trivy on the Jenkins server.

### 3.7.6 Install kubectl on Jenkins Server

We will now install kubectl on the Jenkins server. To do this, we will also create a shell script called “**kubectl\_install.sh**”.

Let us create the file using the command:

```
vi kubectl_install.sh
```



Then, paste the code below in the file

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl  
chmod +x ./kubectl  
sudo mv ./kubectl /usr/local/bin  
kubectl version --short --client
```

The screenshot shows a terminal window in MobaXterm with several tabs open at the top: Jenkins, X server, Sessions, View, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, Help, and X server. The main terminal window displays the following command sequence:

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin
kubectl version --short --client
```

Then save and exit the file.

The screenshot shows a terminal window in MobaXterm with several tabs open at the top: Jenkins, X server, Sessions, View, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, Help, and X server. The main terminal window displays the following command:

```
root@ip-172-31-23-59:~# vi kubectl_install.sh
```

Then, let us make the file executable by using the command:

```
chmod +x kubectl_install.sh
```

```
root@ip-172-31-23-59:~# vi kubectl_install.sh
root@ip-172-31-23-59:~# chmod +x kubectl_install.sh
```

Then, run the file using the command:

```
./kubectl_install.sh
```

```
root@ip-172-31-23-59:~# ./kubectl_install.sh
root@ip-172-31-23-59:~# ./kubectl_install.sh
          % Total    % Received % Xferd  Average Speed   Time   Time     Current
               Dload  Upload Total   Spent   Left Speed
100  57.4M  100  57.4M  0     0  7972k      0:00:07  0:00:07  --:--:-- 9536k
Client Version: v1.19.6-eks-49a6c0
root@ip-172-31-23-59:~#
```

Kubectl has been installed in the Jenkins server.

## 3.8 Create and Configure Virtual Machine for Monitoring

Let us create a virtual machine that will serve as our server for Monitoring.

### 3.8.1 Create Virtual Machine for Monitor

We will start by creating the virtual machine for Monitoring called “**Monitor**”. Go AWS Management console.

The screenshot shows the AWS Management Console EC2 service. The URL is [aws.amazon.com/ec2/instances/launch](#). The page title is "Launch an instance". The "Name and tags" section has a text input field containing "e.g. My Web Server" with an orange arrow pointing to it. Below this is the "Application and OS Images (Amazon Machine Image)" section, which includes a search bar and a grid of operating system icons: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. To the right is a "Summary" panel showing 1 instance, the selected AMI (Amazon Linux 2023 AMI 2023.9.2), the instance type (t3.micro), and the security group (New security group). At the bottom are "Cancel", "Launch instance", and "Preview code" buttons.

Let us give the virtual machine a name, we will call it “**Monitor**”

The screenshot shows the "Launch an instance" wizard. The URL is [aws.amazon.com/ec2/instances/launch](#). The "Name and tags" section has a "Name" input field containing "Monitor". The "Description" section below it provides details about the selected AMI (Amazon Linux 2023 kernel-6.1 AMI). The "Application and OS Images (Amazon Machine Image)" section is visible at the bottom.

Then on “**Application and OS Images (Amazon Machine Image)**” and select “**Ubuntu**”

## ▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recents

Quick Start



Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

### Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type  
ami-0ecb62995f68bb549 (64-bit (x86)) / ami-01b9f1e7dc427266e (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

#### Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x86) ▾

AMI ID

ami-0ecb62995f68bb549

Publish Date

2025-10-22

Username

ubuntu

Verified provider

Scroll down to “Instance Type” and select “t2.large”

## ▼ Instance type [Info](#) | [Get advice](#)

### Instance type

t2.large  
Family: t2 2 vCPU 8 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.1208 USD per Hour On-Demand RHEL base pricing: 0.1216 USD per Hour  
On-Demand SUSE base pricing: 0.1928 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0963 USD per Hour  
On-Demand Linux base pricing: 0.0928 USD per Hour

All generations

[Compare instance types](#)

**Additional costs apply for AMIs with pre-installed software**

Scroll down to “Key Pair” and select the key we created previously

## ▼ Instance type [Info](#) | [Get advice](#)

### Instance type

t2.medium  
Family: t2 2 vCPU 4 GiB Memory Current generation: true  
On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour On-Demand Linux base pricing: 0.0464 USD per Hour  
On-Demand RHEL base pricing: 0.0752 USD per Hour On-Demand Windows base pricing: 0.0644 USD per Hour  
On-Demand SUSE base pricing: 0.1464 USD per Hour

All generations

[Compare instance types](#)

**Additional costs apply for AMIs with pre-installed software**

Scroll down to “Network Settings” and select the security group we created.

**▼ Network settings** [Info](#)

[Edit](#)

**Network** | [Info](#)  
vpc-0d74d3736a240e572

**Subnet** | [Info](#)  
No preference (Default subnet in any availability zone)

**Auto-assign public IP** | [Info](#)  
Enable

**Firewall (security groups)** | [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group       Select existing security group

**Common security groups** | [Info](#)

Select security groups ▾

Primary-SG sg-002d4edfb66259799 [X](#)  
VPC: vpc-0d74d3736a240e572

[Compare security group rules](#)

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Scroll down to “Configure Storage” and make it “30GiB”

**▼ Configure storage** [Info](#)

Advanced

1x  GiB  Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

[Edit](#)

[Cancel](#) [Launch instance](#) [Preview code](#)

Click refresh to view backup information  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

**► Advanced details** [Info](#)

Then, click on “Launch Instance”

A screenshot of the AWS EC2 Instances launch success page. At the top, there's a green success banner stating "Successfully initiated launch of instance (i-0f054e188e84cebb4)". Below it, a red arrow points from the text "Click on 'Instances'" to the "Instances" link in the breadcrumb navigation ("EC2 > Instances > Launch an instance"). The main content area shows various "Next Steps" options like "Create billing usage alerts", "Connect to your instance", "Connect an RDS database", etc., each with a "Learn more" link. At the bottom, there are links for CloudShell, Feedback, and Console Mobile App.

Click on “Instances”

A screenshot of the AWS EC2 Instances list page. The left sidebar shows navigation links for EC2 (Dashboard, Global View, Events, Instances, Images, Elastic Block Store, Network & Security), AMIs, and Capacity Manager. The main content area displays a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One instance, "Monitor" (ID: i-0f054e188e84cebb4), is shown as "Initializing". A red arrow points from the text "We have created the instance and it is initializing, let us wait for it to pass the “2/2 check”" to the "Monitor" row. The bottom of the page includes links for CloudShell, Feedback, and Console Mobile App.

We have created the instance and it is initializing, let us wait for it to pass the “2/2 check”

The screenshot shows the AWS EC2 Instances page with seven instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0ef4c1febc0e19e91	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Slave-1	i-0582e758212bd7f3d	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Slave-2	i-0a3f8f290a4942336	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Monitor	i-0f054e188e84cebb4	Running	t2.large	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
SonarQube	i-0d35e1225e07f37be	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Nexus	i-0a271aa7284fa9682	Running	t2.medium	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c
Jenkins	i-00ef31279720e00a2	Running	t2.large	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c

The “2/2 check” has passed.

### 3.8.2 SSH Connect to Virtual Machine for Jenkins

Let us create a duplicate of the session “Master”.

The screenshot shows the MobaXterm interface with multiple sessions open:

- Session 16: Master (Active)
- Session 17: Slave-1
- Session 18: SonarQube
- Session 19: Nexus
- Session 20: X server

The 'Master' session terminal window displays the following content:

```

Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Sat Jan 3 20:11:41 UTC 2026

System Load: 0.0 Processes: 121
Usage of /: 9.5% of 18.33GB Users Logged in: 0
Memory usage: 6% IPv4 address for enX0: 172.31.23.72
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-23-72:~$ 

```

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

Right-click on the session name “Master”

Nexus

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultExec Tunneling Packages Settings Help

Quick connect...

User sessions

- M Execute
- N Connect as...
- S Ping host
- S Rename session
- E Edit session
- D Delete session
- Duplicate session**
- S Save session to file
- C Create a desktop shortcut
- S Save session settings as default presets
- C Copy session settings

• MobaXterm Personal Edition v24.1 •  
 (SSH client, X server and network tools)

Session to **ubuntu@52.55.215.200**

- Protocol: SSH
- Compression: ✓
- X-Browser: ✓
- Forwarding: ✓ (remote display is forwarded through SSH)

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

Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86\_64)

Session information as of Sat Jan 3 20:11:41 UTC 2026

System load: 0.0 Processes: 121  
 Usage of /: 9.5% of 18.33GB Users logged in: 0  
 Memory usage: 6% IPv4 address for enX0: 172.31.23.72  
 Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.  
 0 updates can be applied immediately.  
 Enable ESM Apps to receive additional future security updates.  
 See <https://ubuntu.com/esm> or run: sudo pro status

The list of available updates is more than a week old.  
 To check for new updates run: sudo apt update

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist  
 To run a command as administrator (user "root"), use "sudo <command>".  
 See "man sudo\_root" for details.

ubuntu@ip-172-31-23-72:~\$ █

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

## Select “Duplicate Session”

Master

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultExec Tunneling Packages Settings Help

Quick connect...

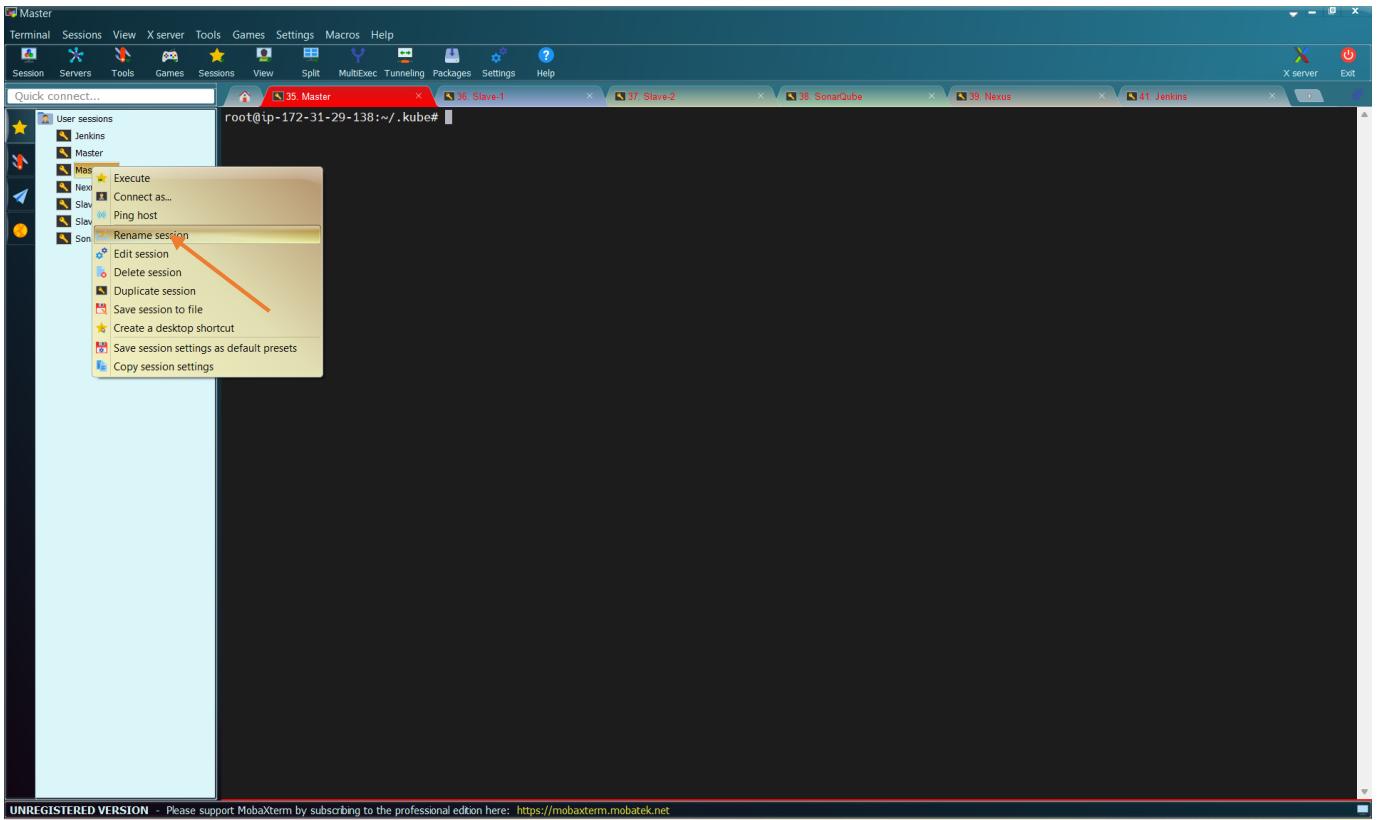
User sessions

- Jenkins
- Master
- Master (1)**
- Nexus
- Slave-1
- Slave-2
- SonarQube

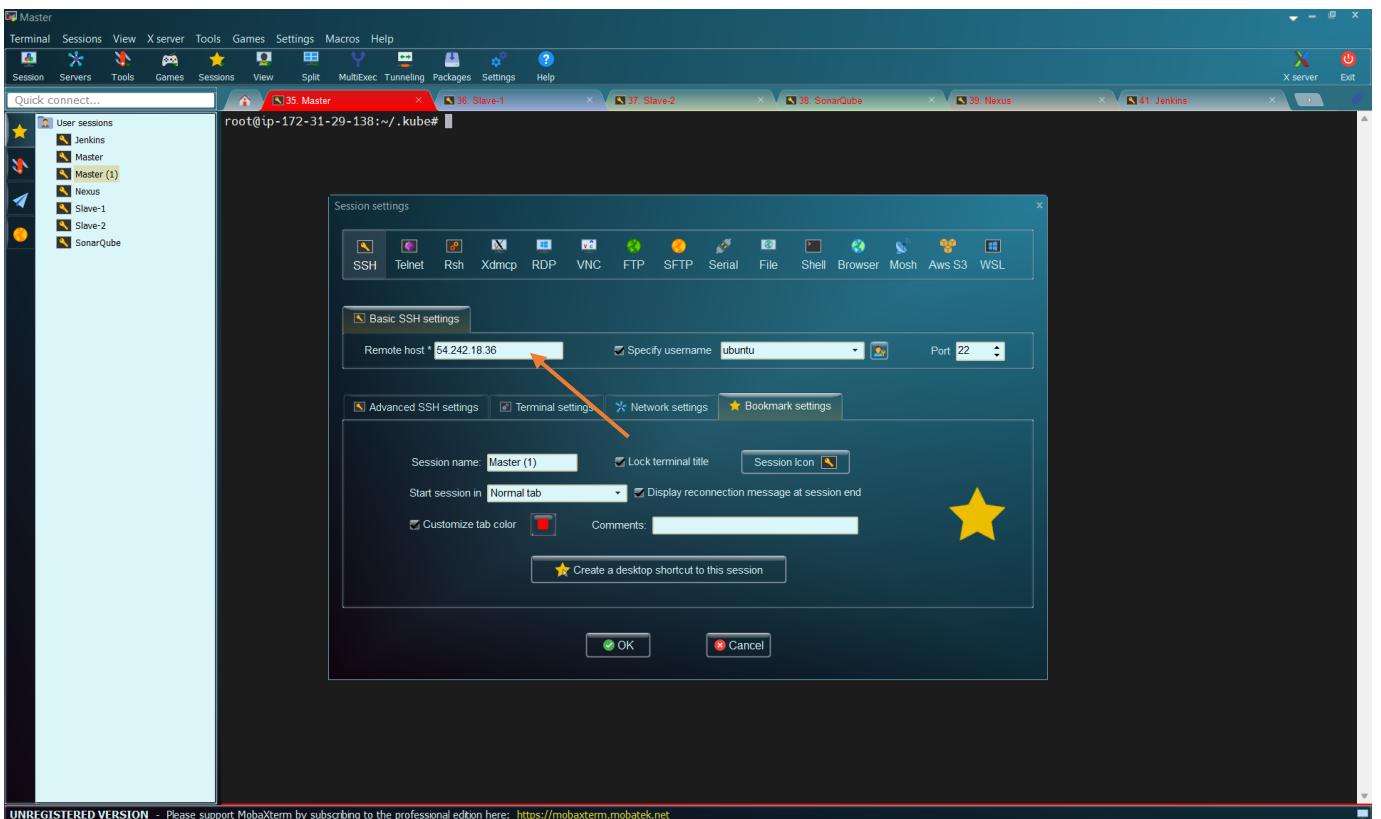
root@ip-172-31-29-138:~/.kube# █

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

We want to rename “Master(1)” to “Jenkins”. Right-click on “Master(1)”



### Select “Rename Session”



Copy the Public IP address of our “Monitor” virtual machine and paste here.

Instances (1/7) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Master	i-0ef4c1feb0e19e91	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c
Slave-1	i-0582e758212bd7f3d	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c
Slave-2	i-0a3f8f290a4942356	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c
<b>Monitor</b>	<b>i-0f054e188e84cebb4</b>	<b>Running</b>	<b>t2.large</b>	<b>2/2 checks passed</b>	<b>View alarms +</b>	<b>us-east-1c</b>
SonarQube	i-0d35e1225e07f37be	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c
Nexus	i-0a271aa7284fa9682	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1c
Jenkins	i-00ef31279720e00a2	Running	t2.large	2/2 checks passed	View alarms +	us-east-1c

**i-0f054e188e84cebb4 (Monitor)**

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

**Instance summary**

Instance ID: i-0f054e188e84cebb4  
 Instance state: Running  
 Hostname type: IP name: ip-172-31-17-38.ec2.internal  
 Answer private resource DNS name: IPv4 (A)  
 Public IPv4 address: 54.160.221.85 | open address  
 Private IP DNS name (IPv4 only): ip-172-31-17-38.ec2.internal  
 Instance type: t2.large  
 Private IPv4 addresses: 172.31.17.58  
 Public DNS: ec2-54-160-221-85.compute-1.amazonaws.com | open address  
 Elastic IP addresses: -

Copy the Public IP: 54.160.221.85 and paste the MobaXterm

Master

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

root@ip-172-31-29-138:~/.kube#

Session settings

Basic SSH settings: Remote host: 54.160.221.85, Specify username: ubuntu, Port: 22

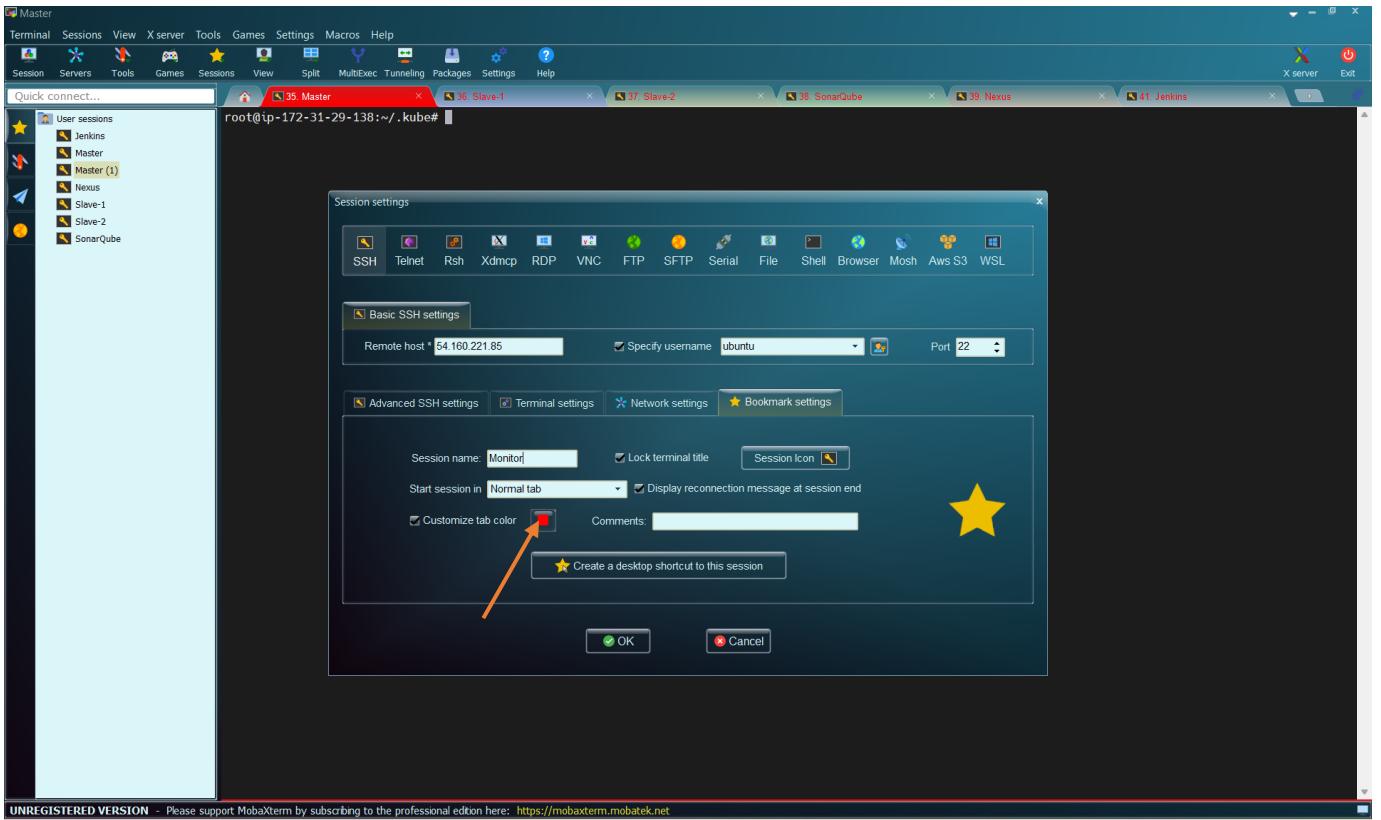
Advanced SSH settings: Start session in: Normal tab, Lock terminal title, Session Icon: Star icon

Session name: Master (1)

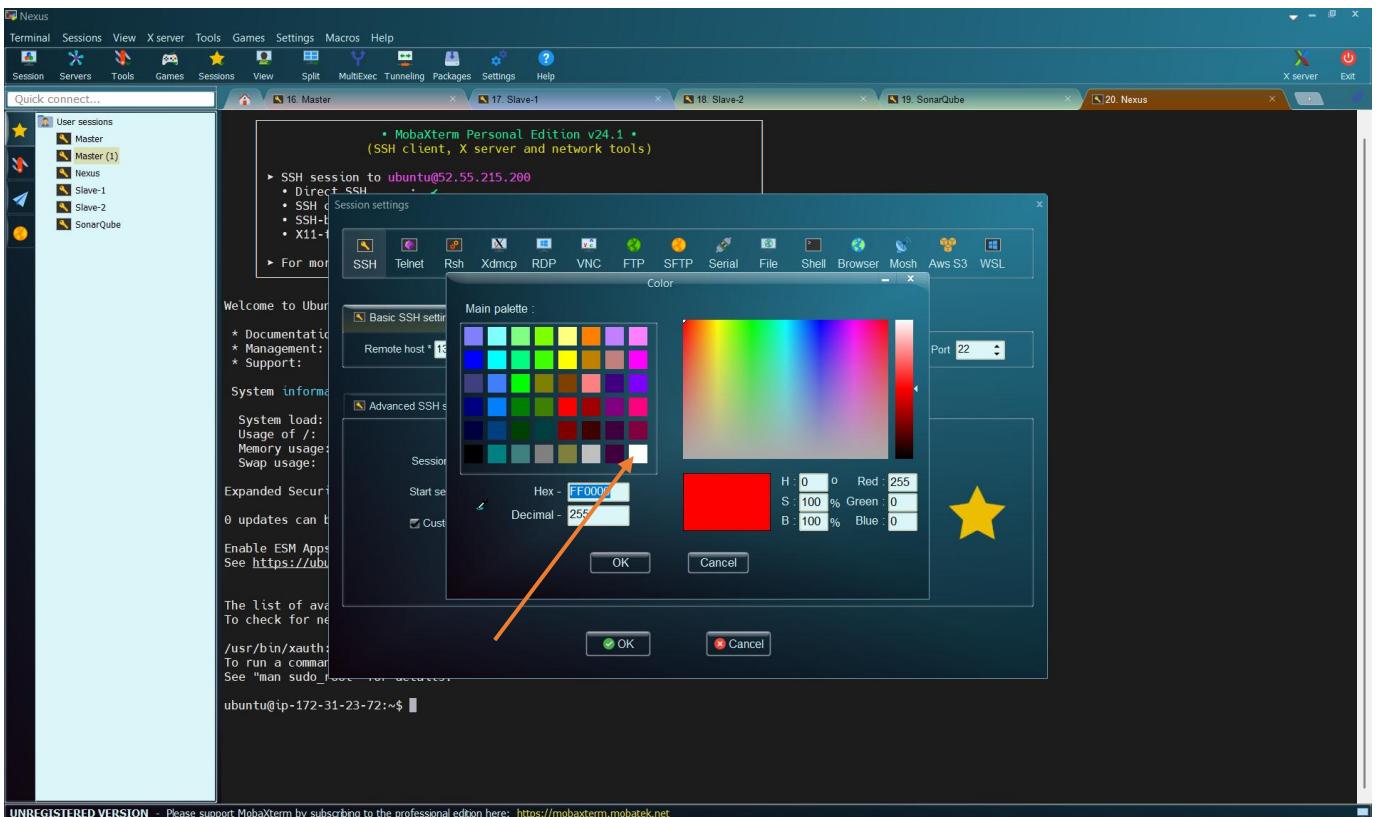
OK Cancel

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

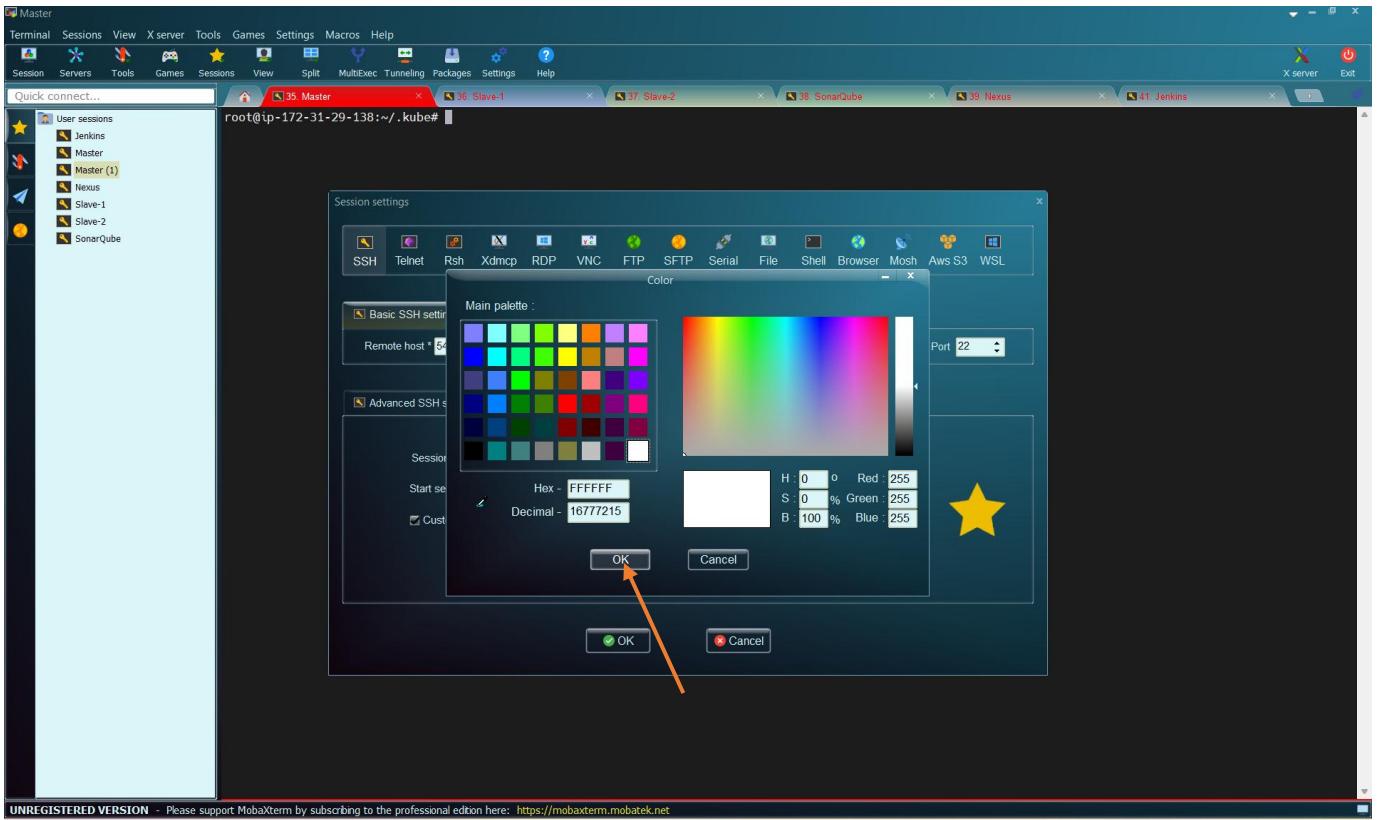
Then, change the name to “Monitor”



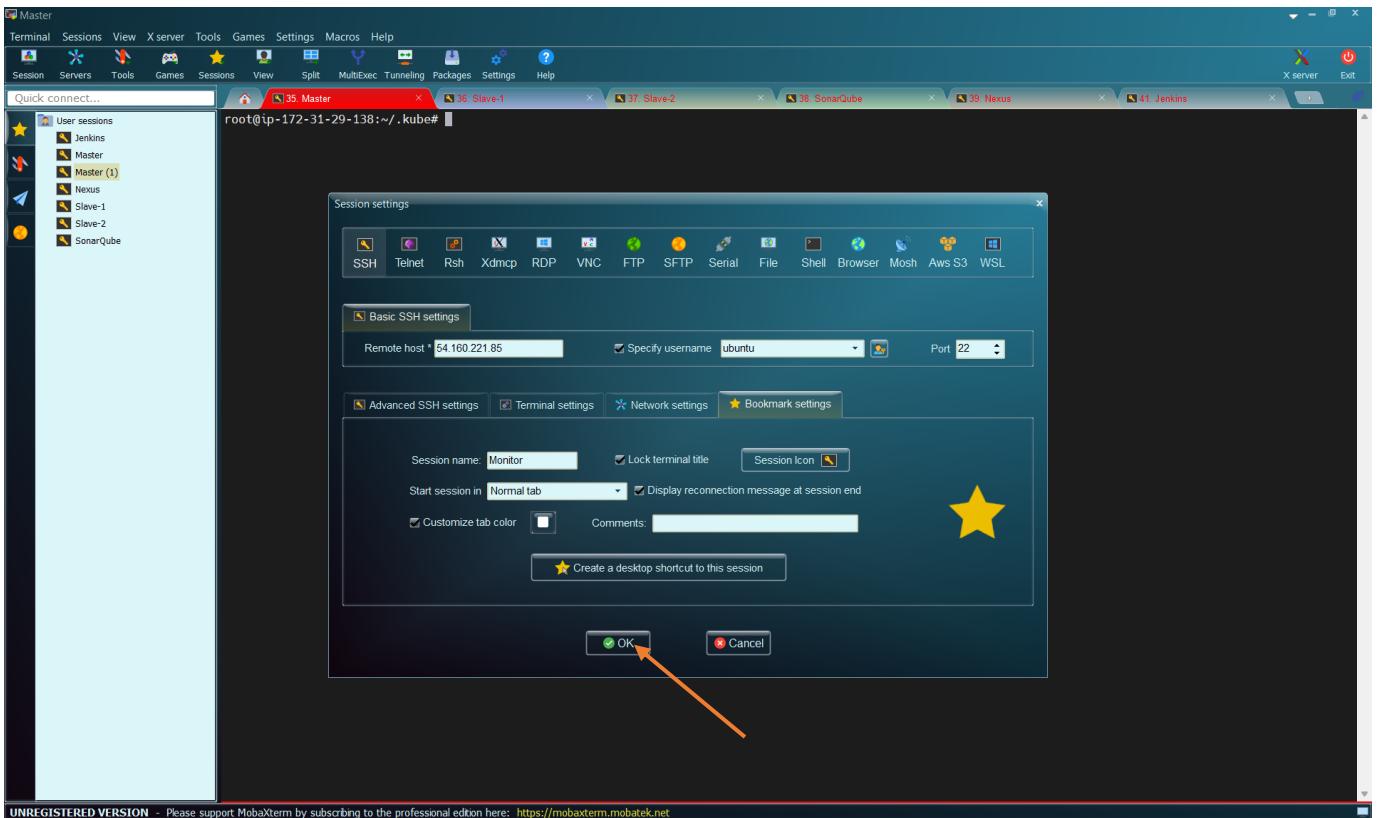
Then, let us change the color. Click on the RED box on “Customize tab color”



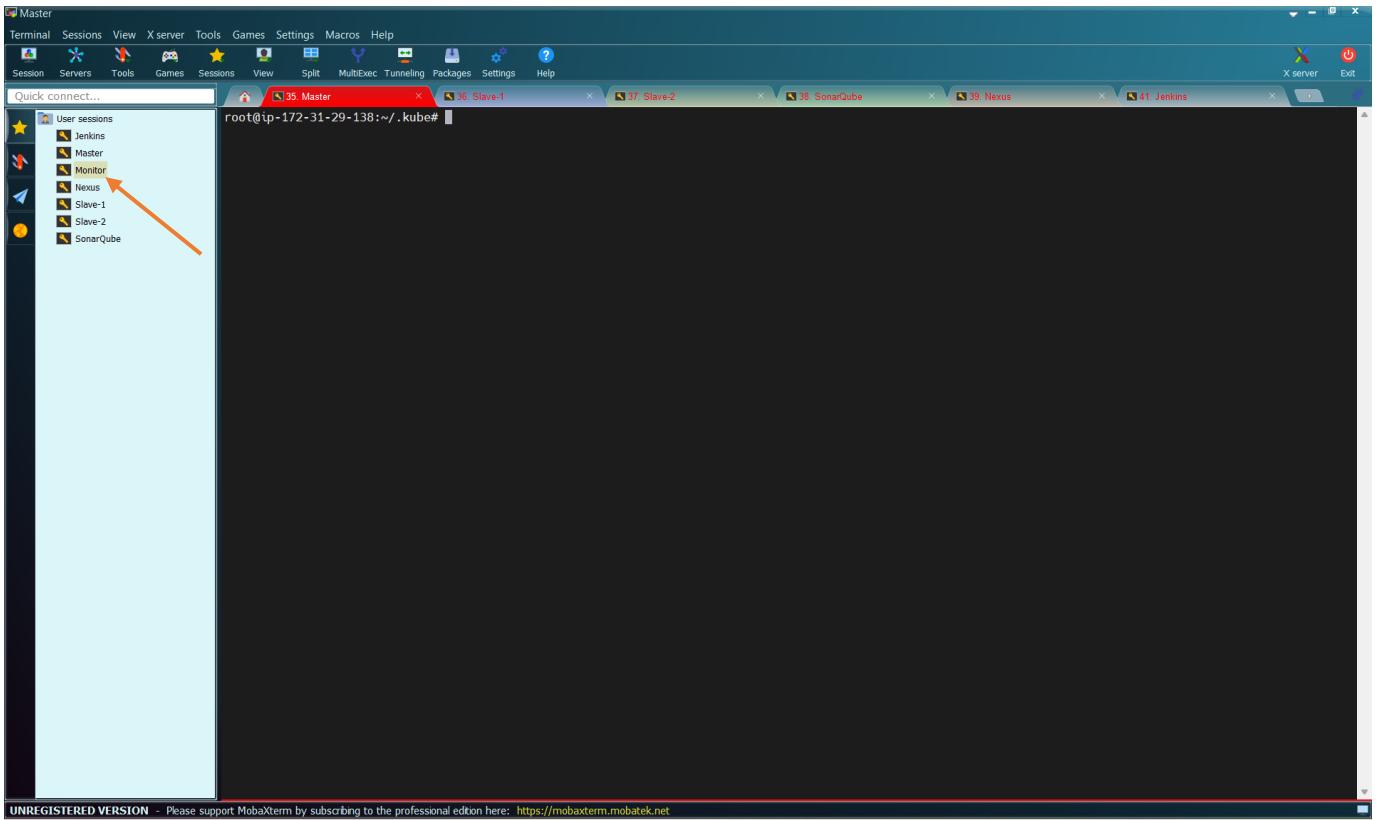
Let us select “White”box



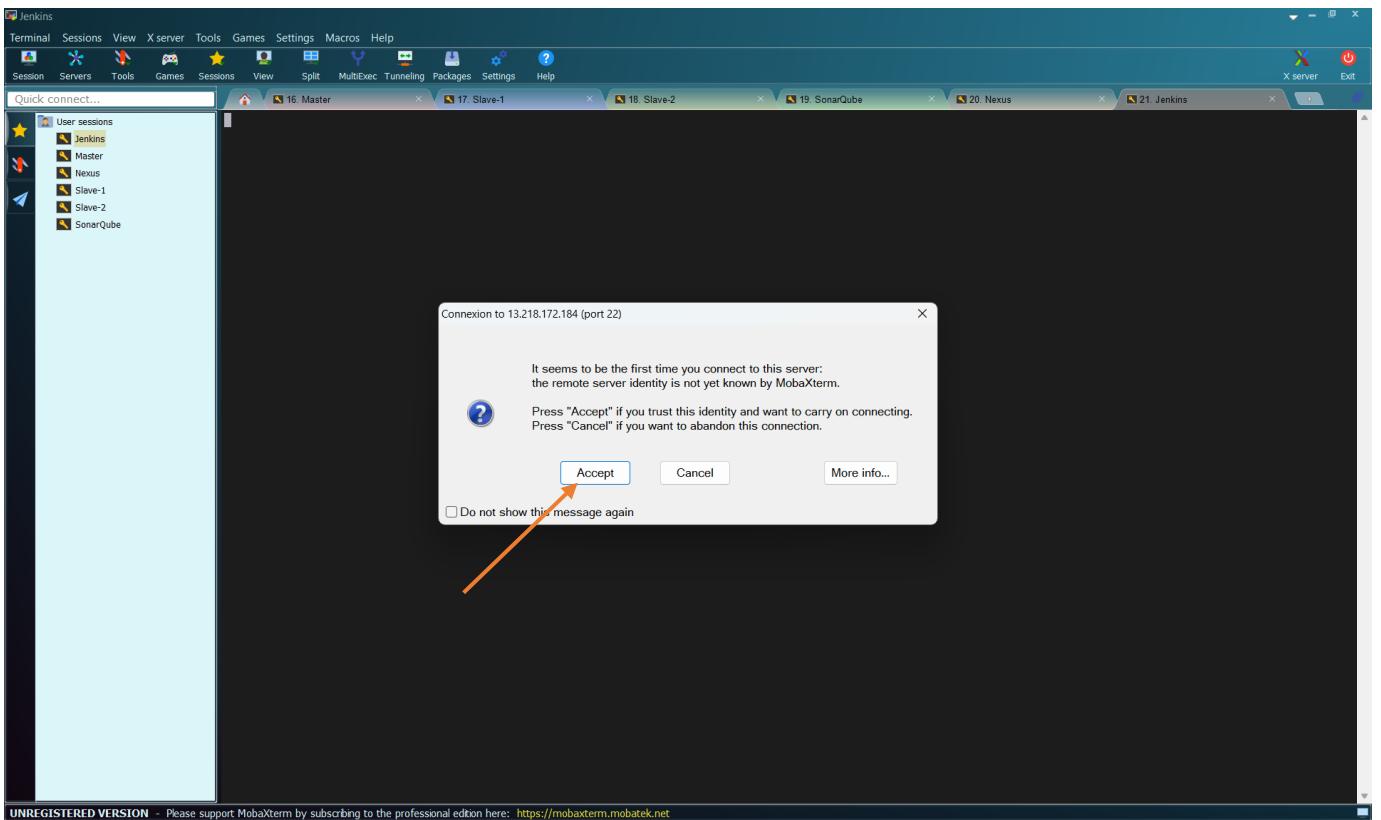
Click on “OK”



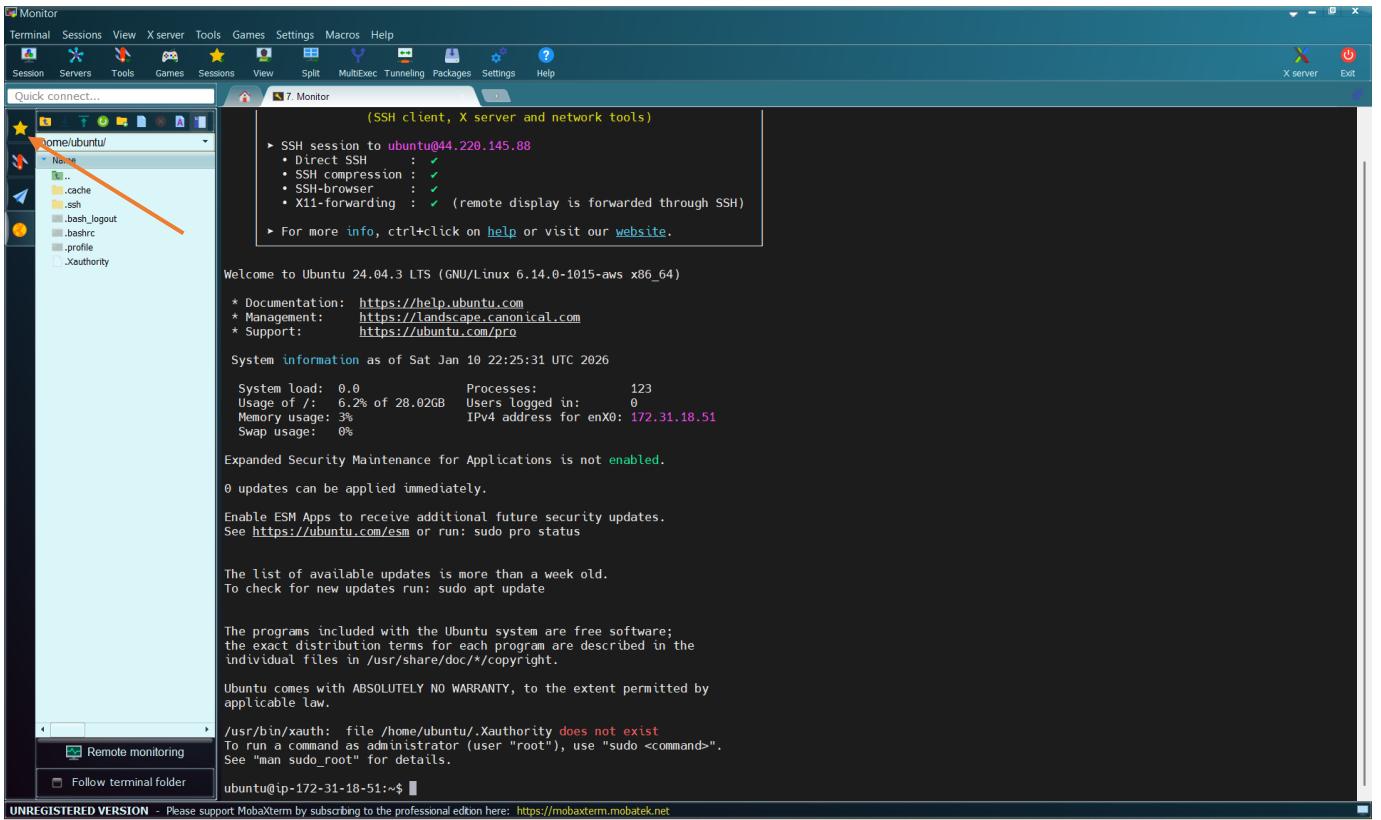
And click on “OK” again



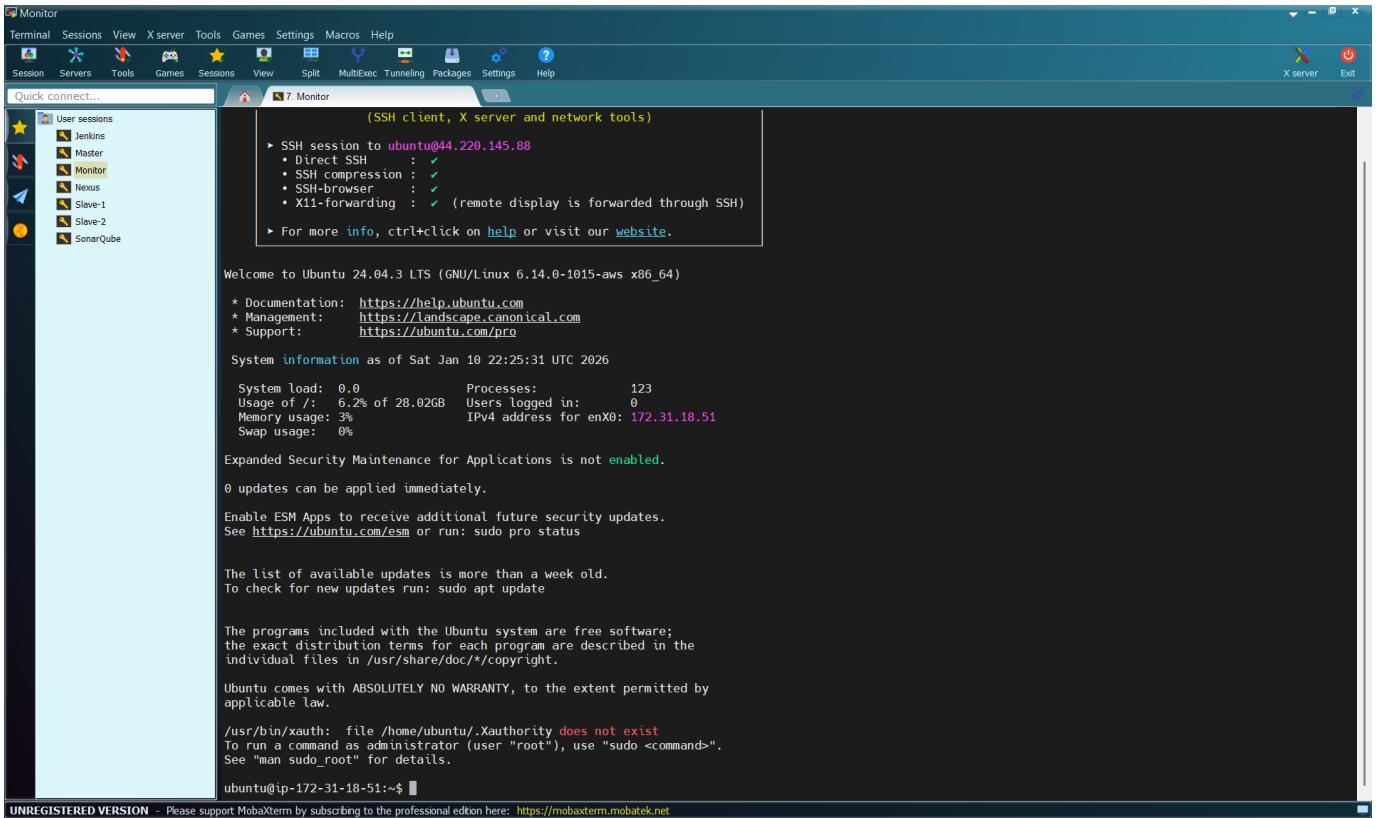
Then, double-click on the session “Monitor”



Click on “Accept”

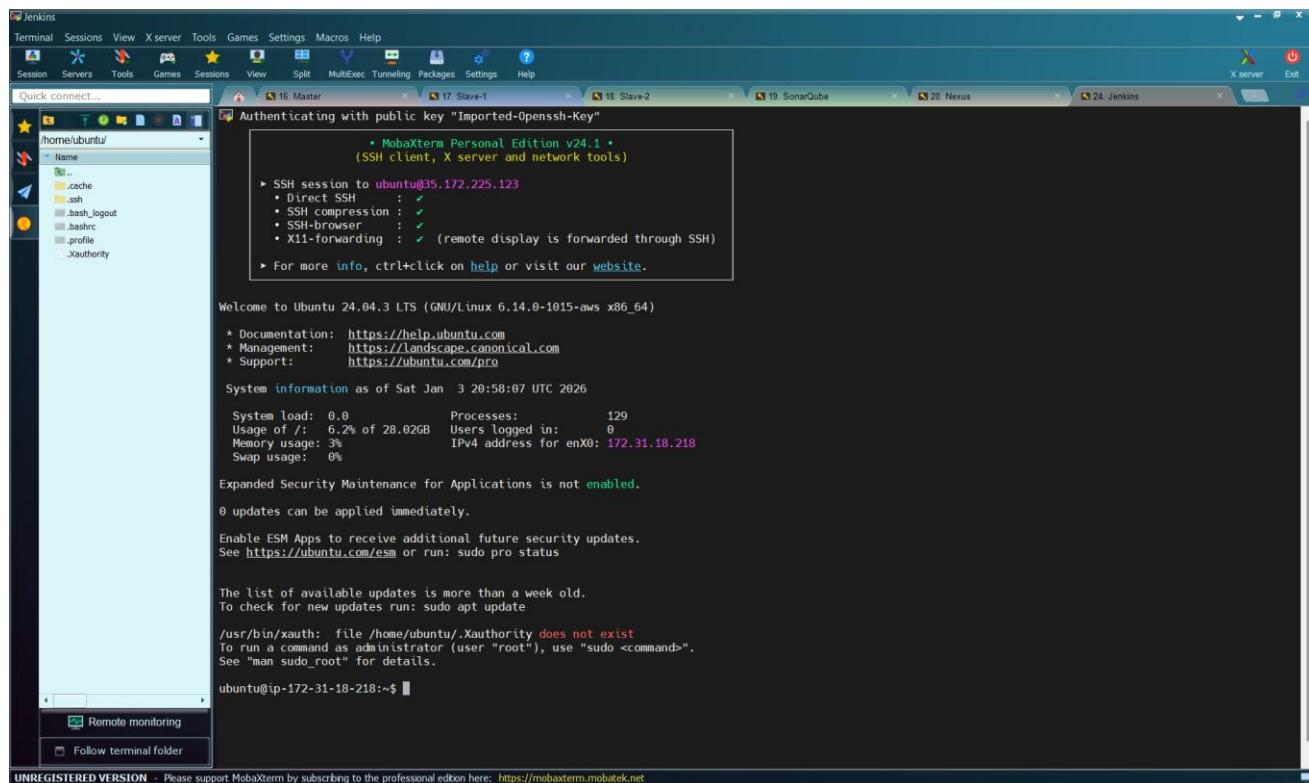


You can see that the tab of “Jenkins” is “Grey”. Click on the star



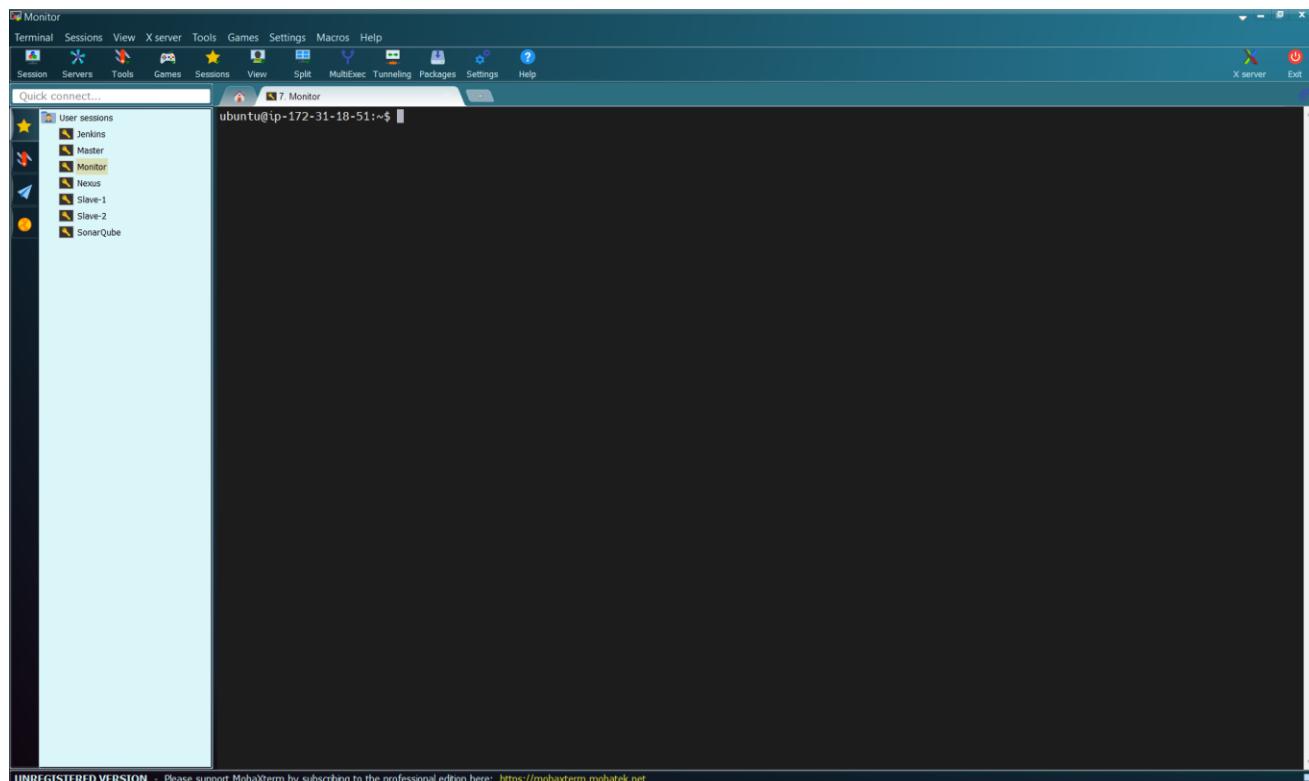
### 3.8.3 Install Prometheus

Let us install Prometheus.



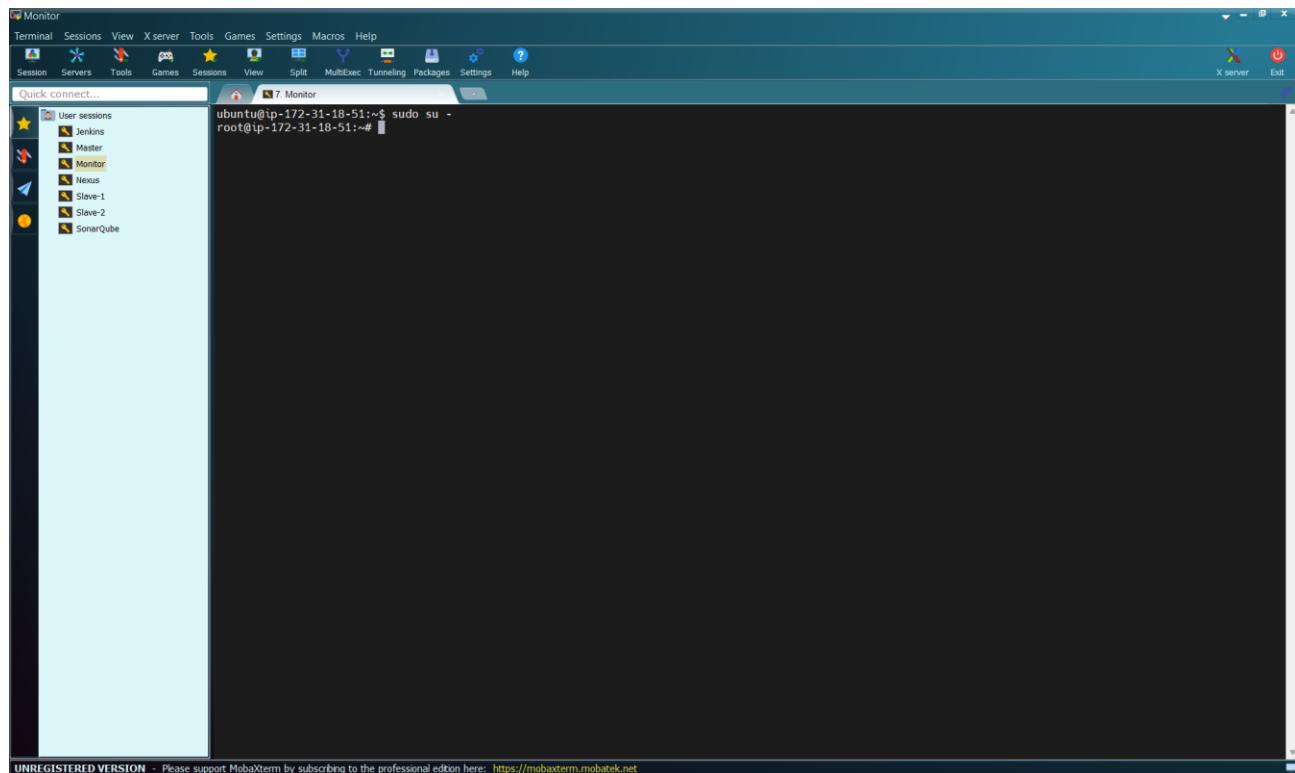
Run the command to clear the terminal:

```
clear
```



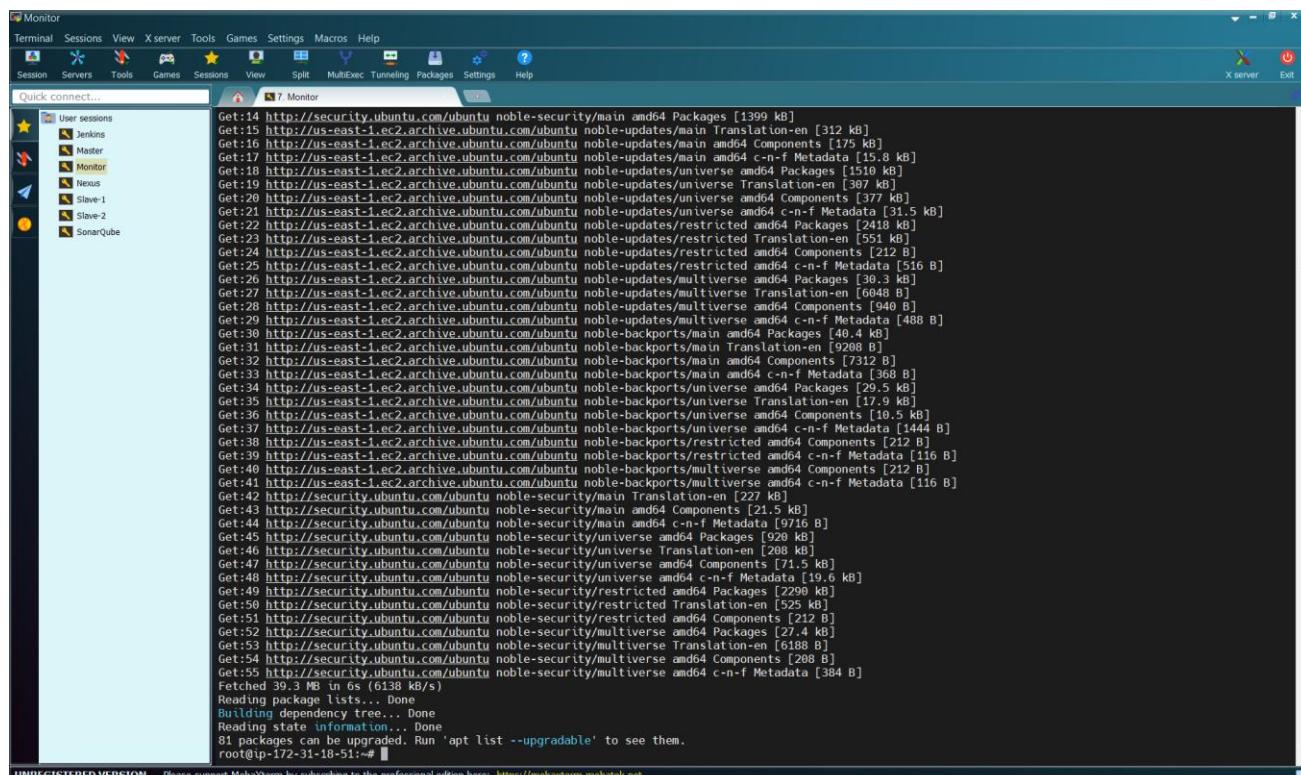
Give the Monitor server the root user privilege by running the command:

```
sudo su -
```



Update the packages using the command:

```
sudo apt update
```



Let us download Prometheus. Go to <https://prometheus.io/download/>

The screenshot shows the Prometheus download page. The 'prometheus' section is selected, displaying the 3.9.1 / 2026-01-07 release. An orange arrow points to the 'prometheus-3.9.1.linux-amd64.tar.gz' file entry in the table.

File name	OS	Arch	Size	SHA256 Checksum
prometheus-3.9.1.darwin-amd64.tar.gz	darwin	amd64	137.12 MiB	e551f30206e503edfdca8748215a18be49215427f138f64665ec382ece3191
prometheus-3.9.1.darwin-arm64.tar.gz	darwin	arm64	130.56 MiB	0908d168789a80a7e56fc56fdb1fb6572e46d1a8e9ab3e62197b95d51b
<b>prometheus-3.9.1.linux-amd64.tar.gz</b>	linux	amd64	125.84 MiB	86a6999dd6aaacbd994acde93c77cf314d4be1c8e7b7c58f44355c77b32c584
prometheus-3.9.1.windows-amd64.zip	windows	amd64	129.67 MiB	50e29e701fea961c880109f017db3900baa106e9b0fe95888e86ff97dbc6e

Below this, the 3.5.0 / 2025-07-14 LTS section is shown, containing similar file entries for Darwin, Linux, and Windows.

### alertmanager

Prometheus Alertmanager

File name	OS	Arch	Size	SHA256 Checksum
alertmanager-0.30.0.darwin-amd64.tar.gz	darwin	amd64	36.62 MiB	244525cca9457879e67e6794dc98ad3ee780075738c27bd648ae5404b223aa

Right-click on “prometheus-3.9.1.linux-amd64.tar.gz”

The screenshot shows the same Prometheus download page, but now the 'prometheus-3.9.1.linux-amd64.tar.gz' file has been right-clicked. A context menu is open, with an orange arrow pointing to the 'Copy Link Address' option.

Context menu options include:

- Open link in new tab
- Open link in split view
- Open link in new window
- Open link in incognito window
- Open link in GitHub
- Save link as...
- Copy Link Address (highlighted)
- AdBlock — block ads across the web
- Inspect

The URL <https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz> is visible at the bottom of the browser's address bar.

Select “Copy Link Address”

<https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz>

Then run the command:

```
wget https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
```

```
root@ip-172-31-18-51:~# wget https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
--2026-01-21 23:04:26 - https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
Resolving github.com (github.com) [149.82.113.4]
Connecting to github.com (github.com) [149.82.113.4]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://release-assets.githubusercontent.com/github-production-release-asset/6838921/66cad596-6d94-4fe8-a72a-23b7cb5b2b81?sp=r&sv=2018-11-09&sr=b&sp
=&https://sse2026-01-10T23%3A49%3A56%26rscd=attachment%3Bfilename%3Dprometheus-3.9.1.linux-amd64.tar.gz&rscf=application%2Foctet-stream&skid=96c2d410-5711-43a1
-aedd-ab1947aa7ab08sktk=398a6654-997b-47e9-b12b-9515b8964de5kskt=2026-01-10T23%3A49%3A517&skse=2026-01-10T23%3A49%3A56%26rscs=b&svk=2018-11-09&si=q=31xyimF%f
2BxI7xM4tMp9K0%2EZRZLH7FuWrBDs3D5jwIwxy10eXA101jKV101jCjhGc101jUJ9zT1Nj19_eyJpc3M0i1naX0odnWtY29tIiwYXVki1oicmVsWfz7S1h-3NlDHmZ210afVifxN1cmNvbnRlhno
iY29tIiwiaV5MsImV4c16MT-200A40tgMKwibmJmIoxNzY4MDq2MjYmlCjwYX0ljoicmVsWfz7Wfz7S1hVjdg1vhb15hb91lmNvcmlud2luZ93cysuZ01f0_jMv74v_qYR0e-0
n0_XlThDpFmInK2hDDYsecjxdpPg&response-content-disposition=attachment%3Bfilename%3Dprometheus-3.9.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2026-01-21 23:04:29 - https://release-assets.githubusercontent.com/github-production-release-asset/6838921/66cad596-6d94-4fe8-a72a-23b7cb5b2b81?sp=r&sv=201
9-3-1109&si=q=31xyimF%f2BxI7xM4tMp9K0%2EZRZLH7FuWrBDs3D5jwIwxy10eXA101jKV101jCjhGc101jUJ9zT1Nj19_eyJpc3M0i1naX0odnWtY29tIiwYXVki1oicmVsWfz7S1h-3NlDHmZ210afVifxN1cmNvbnRlhno
iY29tIiwiaV5MsImV4c16MT-200A40tgMKwibmJmIoxNzY4MDq2MjYmlCjwYX0ljoicmVsWfz7Wfz7S1hVjdg1vhb15hb91lmNvcmlud2luZ93cysuZ01f0_jMv74v_qYR0e-0n0_XlThDpFmInK2hDDYsecjxdpPg&response-content-disposition=attachment%3Bfilename%3Dprometheus-3.9.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving release-assets.githubusercontent.com (release-assets.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 131949217 (120M) [application/octet-stream]
Saving to: 'prometheus-3.9.1.linux-amd64.tar.gz'

prometheus-3.9.1.linux-amd64.tar.gz    100%[=====] 125.84M 89.4MB/s    in 1.4s

2026-01-20 23:04:22 (9.40 MB/s) - 'prometheus-3.9.1.linux-amd64.tar.gz' saved [131949217/131949217]

root@ip-172-31-18-51:~#
```

Then check the content of the download using the command:

15

Monitor  
Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help ?

Quick connect... 7 Monitor

User sessions  
Jenkins Master Monitor Nexus Slave-1 Slave-2 SonarQube

```
root@ip-172-31-18-51:~# wget https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 149.82.113.4
Connecting to github.com (github.com)|149.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://release-assets.githubusercontent.com/github-production-release-asset/6838921/66cad596-6d94-4fe8-a72a-23b7bc5b2b812spref&sv=2018-11-09&s=1&sp=ef&t=2026-01-10T22%3A49%3A56Z&scd=attachment%3B+filename%3Dprometheus-3.9.1.linux-amd64.tar.gz&scf=application%2Foctet-stream&skoid=96c2d410-5711-43a1-aedd-ab1947aa0b86skt1398a6654-997b-47e9-b12b-951b58964dc6skt12026-01-10T22%3A49%3A5176skv=2026-01-10T22%3A49%3A5676skv=b6skv=2018-11-09&s=1&xjyjmnFs%3Bzr1zYw1zWd1tpkX0zEX2LH7IuWp0z830jw7ye0xeXa01JKV101LChb6c1031Uu2L1mz19_eyjpcM-01JmxAxRwHuY2H1uLyyXX1k1i1cmVzAFz7S1hc3NldHm721QahVidxW1cmNldbnRhn0yY29tLH0pfmInK2xb00YsecjxdpPg&response-content-disposition=attachment%3B%20filename%40prometheus-3.9.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
-> 2026-01-10 23:04:20... https://release-assets.githubusercontent.com/github-production-release-asset/6838921/66cad596-6d94-4fe8-a72a-23b7bc5b2b812spref&sv=2018-11-09&s=1&sp=ef&t=2026-01-10T22%3A49%3A56Z&scd=attachment%3B+filename%3Dprometheus-3.9.1.linux-amd64.tar.gz&scf=application%2Foctet-stream&skoid=96c2d410-5711-43a1-aedd-ab1947aa0b86skt1398a6654-997b-47e9-b12b-951b58964dc6skt12026-01-10T22%3A49%3A5176skv=2026-01-10T22%3A49%3A455655skv=b6skv=2018-11-09&s=1&xjyjmnFs%3Bzr1zYw1zWd1tpkX0zEX2LH7IuWp0z830jw7ye0xeXa01JKV101LChb6c1031Uu2L1mz19_eyjpcM-01JmxAxRwHuY2H1uLyyXX1k1i1cmVzAFz7S1hc3NldHm721QahVidxW1cmNldbnRhn0yY29tLH0pfmInK2xb00YsecjxdpPg&response-content-disposition=attachment%3B%20filename%40prometheus-3.9.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving release-assets.githubusercontent.com (release-assets.githubusercontent.com)... 185.199.111.133, 185.199.188.133, 185.199.199.133, ...
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 131949217 (126M) [application/octet-stream]
Saving to: 'prometheus-3.9.1.linux-amd64.tar.gz'

prometheus-3.9.1.linux-amd64.tar.gz    100%[=====] 125.84M 89.4MB/s   in 1.4s

2026-01-10 23:04:22 (89.40 MB/s) - 'prometheus-3.9.1.linux-amd64.tar.gz' saved [131949217/131949217]

root@ip-172-31-18-51:~# ls
prometheus-3.9.1.linux-amd64.tar.gz  snap
root@ip-172-31-18-51:~#
```

Then, untar the Prometheus folder using the command:

```
tar -xvf prometheus-3.9.1.linux-amd64.tar.gz
```

```
root@ip-172-31-18-51:~# wget https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
--2026-01-10 23:04:20-- https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://release-assets.githubusercontent.com/github-production-release-asset/6838921/66cad4596-6d94-4fe8-a72a-23b7bc5b2b81?sp=r&sv=2018-11-09&sr=b&sp=r&httpsse2026-01-10T23%3A49%3A56%2fscd=attachment%3B+filename%3Dprometheus-3.9.1.linux-amd64.tar.gz&rsc=application%2foctet-stream&skid=96c2d410-5711-43a1-aedd-ab1947aa7ab08&skt=398a6654-997b-47e9-b12b-9515b89664de&st=2026-01-10T23%3A49%3A56%2fscs=b&skv=2018-11-09&sig=31xyjmnfs%2Bx1qf7Xm4tMg%K0%2fXR2LHZtFuWrR0%30Jw!vtey!oexA10!CKV10!LChbg!0!JUz1N19_eYjpc3M!0!JnaRodWluY29tIw!vYXVkj!oicmVsZWfZWFzcV8ch!vZHjdglvb15!bg9!lmVcmmlud2u!g93c!y5u!X0!f0..MjZ4v_qYr0-e0n9_XlHdpfInXkX000Ysec!xdPq!&response-content-disposition=attachment%3B+filename%3Dprometheus-3.9.1.linux-amd64.tar.gz&response-content-type=application%2foctet-stream [following]
--2026-01-10 23:04:20-- https://release-assets.githubusercontent.com/github-production-release-asset/6838921/66cad4596-6d94-4fe8-a72a-23b7bc5b2b81?sp=r&sv=2018-11-09&sr=b&sp=r&httpsse2026-01-10T23%3A49%3A56%2fscd=attachment%3B+filename%3Dprometheus-3.9.1.linux-amd64.tar.gz&rsc=application%2foctet-stream&skid=96c2d410-5711-43a1-aedd-ab1947aa7ab08&skt=398a6654-997b-47e9-b12b-9515b89664de&st=2026-01-10T23%3A49%3A56%2fscs=b&skv=2018-11-09&sig=31xyjmnfs%2Bx1qf7Xm4tMg%K0%2fXR2LHZtFuWrR0%30Jw!vtey!oexA10!CKV10!LChbg!0!JUz1N19_eYjpc3M!0!JnaRodWluY29tIw!vYXVkj!oicmVsZWfZWFzcV8ch!vZHjdglvb15!bg9!lmVcmmlud2u!g93c!y5u!X0!f0..MjZ4v_qYr0-e0n9_XlHdpfInXkX000Ysec!xdPq!&response-content-disposition=attachment%3B+filename%3Dprometheus-3.9.1.linux-amd64.tar.gz&response-content-type=application%2foctet-stream [following]
--2026-01-10 23:04:20-- https://release-assets.githubusercontent.com/(release-assets.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 131949217 (126M) [application/octet-stream]
Saving to: 'prometheus-3.9.1.linux-amd64.tar.gz'

prometheus-3.9.1.linux-amd64.tar.gz 100%[=====] 125.84M 89.4MB/s in 1.4s

2026-01-10 23:04:22 (89.4 MB/s) - 'prometheus-3.9.1.linux-amd64.tar.gz' saved [131949217/131949217]

root@ip-172-31-18-51:~# ls
prometheus-3.9.1.linux-amd64.tar.gz snap
root@ip-172-31-18-51:~# tar -xvf prometheus-3.9.1.linux-amd64.tar.gz
prometheus-3.9.1.linux-amd64/
prometheus-3.9.1.linux-amd64/NOTICE
prometheus-3.9.1.linux-amd64/LICENSE
prometheus-3.9.1.linux-amd64/prometheus.yml
prometheus-3.9.1.linux-amd64/promtool
prometheus-3.9.1.linux-amd64/prometheus
root@ip-172-31-18-51:~#
```

Then, let us remove the untar file using the command:

```
rm -rf prometheus-3.9.1.linux-amd64.tar.gz
```

```
root@ip-172-31-18-51:~# wget https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
--2026-01-10 23:04:20-- https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://release-assets.githubusercontent.com/github-production-release-asset/6838921/66cad4596-6d94-4fe8-a72a-23b7bc5b2b81?sp=r&sv=2018-11-09&sr=b&sp=r&httpsse2026-01-10T23%3A49%3A56%2fscd=attachment%3B+filename%3Dprometheus-3.9.1.linux-amd64.tar.gz&rsc=application%2foctet-stream&skid=96c2d410-5711-43a1-aedd-ab1947aa7ab08&skt=398a6654-997b-47e9-b12b-9515b89664de&st=2026-01-10T23%3A49%3A56%2fscs=b&skv=2018-11-09&sig=31xyjmnfs%2Bx1qf7Xm4tMg%K0%2fXR2LHZtFuWrR0%30Jw!vtey!oexA10!CKV10!LChbg!0!JUz1N19_eYjpc3M!0!JnaRodWluY29tIw!vYXVkj!oicmVsZWfZWFzcV8ch!vZHjdglvb15!bg9!lmVcmmlud2u!g93c!y5u!X0!f0..MjZ4v_qYr0-e0n9_XlHdpfInXkX000Ysec!xdPq!&response-content-disposition=attachment%3B+filename%3Dprometheus-3.9.1.linux-amd64.tar.gz&response-content-type=application%2foctet-stream [following]
--2026-01-10 23:04:20-- https://release-assets.githubusercontent.com/(release-assets.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 131949217 (126M) [application/octet-stream]
Saving to: 'prometheus-3.9.1.linux-amd64.tar.gz'

prometheus-3.9.1.linux-amd64.tar.gz 100%[=====] 125.84M 89.4MB/s in 1.4s

2026-01-10 23:04:22 (89.4 MB/s) - 'prometheus-3.9.1.linux-amd64.tar.gz' saved [131949217/131949217]

root@ip-172-31-18-51:~# ls
prometheus-3.9.1.linux-amd64.tar.gz snap
root@ip-172-31-18-51:~# tar -xvf prometheus-3.9.1.linux-amd64.tar.gz
prometheus-3.9.1.linux-amd64/
prometheus-3.9.1.linux-amd64/NOTICE
prometheus-3.9.1.linux-amd64/LICENSE
prometheus-3.9.1.linux-amd64/prometheus.yml
prometheus-3.9.1.linux-amd64/promtool
prometheus-3.9.1.linux-amd64/prometheus
root@ip-172-31-18-51:~# rm -rf prometheus-3.9.1.linux-amd64.tar.gz
root@ip-172-31-18-51:~#
```

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

Check the content using the command:

ls

```
root@ip-172-31-18-51:~# wget https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
--2026-01-10 23:04:29 - https://github.com/prometheus/prometheus/releases/download/v3.9.1/prometheus-3.9.1.linux-amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.4
Connecting to github.com (github.com)|140.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://release-assets.githubusercontent.com/github-production-release-asset/6838921/66ca4596-6d94-4fe8-a72a-23b7bc5b2b81?sp=r&sv=2018-11-09&s=b5sp=rhttps&se=2026-01-10T23%3A49%3A56&rsc=attachmetn30prometheus-3.9.1.linux-amd64.tar.gz&srct=application%2foctet-stream&koid=9bc2d410-5711-43a1-aedd-ab1947aa7ab085ktid=398a6654-997b-47e9-b12b-9515b896b4de&skt=2026-01-10T23%3A49%3A56&skse=2018-11-09&sig=31xyimfsjz2bx1q7x04tMg9K0%2fXRZlH2TuWrRDS%30&wt=eyJ0eXA0iJKV10iLjhBcG10iUz11Nj9_eypc3M0iJnaKRodUjY291IwLYXVK1jocmVsZWfzZS1h3NVLdMu2210aHvjdXHlcmVbRlbn0uY291Iwia25jota2yMSi1mV4c16Mtc200A01g2MwihMmjoxh2Y4Dg2HjYwLjwXRojoicmVsZWfzWfzc2V0cIjvHVidg1vb15ibg91lmNvcmluId2uG93:y5u/X0i0_f0_jMv24v_qYR0e-n0_XlH0pimnkXb00YsecjxdpPgkResponse-content-disposition=attachment%3B%2ffilename%3Dprometheus-3.9.1.linux-amd64.tar.gz&response-content-type=application%2foctet-stream [following]
--2026-01-10 23:04:29 - https://release-assets.githubusercontent.com/github-production-release-asset/6838921/66ca4596-6d94-4fe8-a72a-23b7bc5b2b81?sp=r&sv=2018-11-09&s=b5sp=rhttps&se=2026-01-10T23%3A49%3A56&rsc=attachmetn30prometheus-3.9.1.linux-amd64.tar.gz&srct=application%2foctet-stream&koid=9bc2d410-5711-43a1-aedd-ab1947aa7ab085ktid=398a6654-997b-47e9-b12b-9515b896b4de&skt=2026-01-10T23%3A49%3A56&skse=2018-11-09&sig=31xyimfsjz2bx1q7x04tMg9K0%2fXRZlH2TuWrRDS%30&wt=eyJ0eXA0iJKV10iLjhBcG10iUz11Nj9_eypc3M0iJnaKRodUjY291IwLYXVK1jocmVsZWfzZS1h3NVLdMu2210aHvjdXHlcmVbRlbn0uY291Iwia25jota2yMSi1mV4c16Mtc200A01g2MwihMmjoxh2Y4Dg2HjYwLjwXRojoicmVsZWfzWfzc2V0cIjvHVidg1vb15ibg91lmNvcmluId2uG93:y5u/X0i0_f0_jMv24v_qYR0e-n0_XlH0pimnkXb00YsecjxdpPgkResponse-content-disposition=attachment%3B%2ffilename%3Dprometheus-3.9.1.linux-amd64.tar.gz&response-content-type=application%2foctet-stream
Resolving release-assets.githubusercontent.com (release-assets.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.111.133|:443... connected.
Length: 131949217 (120M) [application/octet-stream]
Saving to: 'prometheus-3.9.1.linux-amd64.tar.gz'

prometheus-3.9.1.linux-amd64.tar.gz    100%[=====] 125.84M  89.4MB/s   in 1.4s

2026-01-10 23:04:22 (89.4 MB/s) - 'prometheus-3.9.1.linux-amd64.tar.gz' saved [131949217/131949217]

root@ip-172-31-18-51:~# ls
prometheus-3.9.1.linux-amd64.tar.gz  snap
root@ip-172-31-18-51:~# rm -rf prometheus-3.9.1.linux-amd64.tar.gz
root@ip-172-31-18-51:~# ls
prometheus-3.9.1.linux-amd64  snap
root@ip-172-31-18-51:~#
```

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

Then, go into the Prometheus folder using the command:

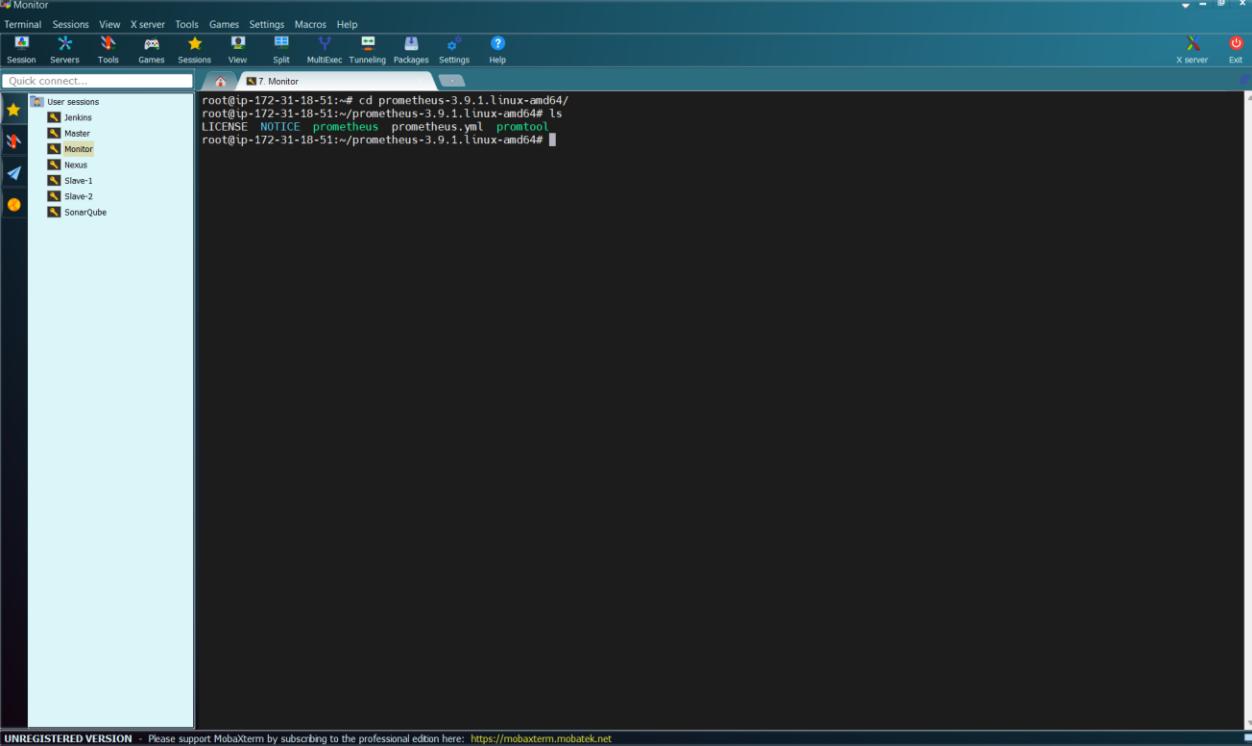
```
cd prometheus-3.9.1.linux-amd64/
```

```
root@ip-172-31-18-51:~# cd prometheus-3.9.1.linux-amd64/
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64#
```

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

Check the content using the command:

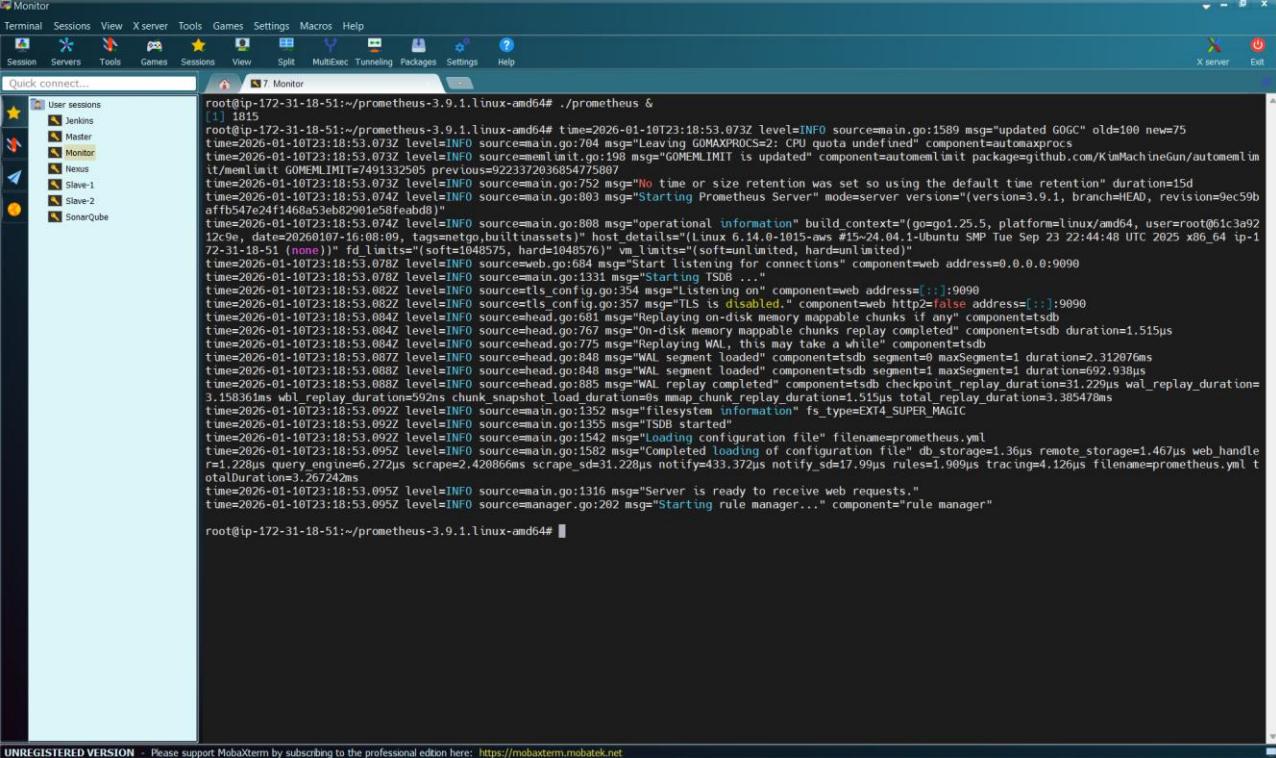
ls



```
root@ip-172-31-18-51:~# cd prometheus-3.9.1.linux-amd64/
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# ls
LICENSE NOTICE prometheus prometheus.yml promtool
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64#
```

Then, let us execute the file to install Prometheus using the command to run it in the background:

./prometheus &



```
[1] 1815
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# ./prometheus &
[1] 1815
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# time=2026-01-10T23:18:53.073Z level=INFO source=main.go:1589 msg="updated GOGC" old=100 new=75
time=2026-01-10T23:18:53.073Z level=INFO source=main.go:704 msg="Leaving GOOS=ROCS=2: CPU quota undefined" component=automaxprocs
time=2026-01-10T23:18:53.073Z level=INFO source=memlimit.go:198 msg="GOMEMLIMIT is updated" component=automemlimit package=github.com/KimMachineGun/automemlim
it/memlimit=GOMEMLIMIT=749132959 previous=922372036854775807
time=2026-01-10T23:18:53.073Z level=INFO source=main.go:752 msg="No time or size retention was set so using the default time retention" duration=15d
time=2026-01-10T23:18:53.073Z level=INFO source=main.go:752 msg="Starting Prometheus Server" mode=server version="v3.9.1, branch=HEAD, revision=9ec59b
afffb547e24f1468a53ab02991c58fbabdb"
time=2026-01-10T23:18:53.074Z level=INFO source=main.go:889 msg="Starting Prometheus Server" mode=server version="v3.9.1, branch=HEAD, revision=9ec59b
afffb547e24f1468a53ab02991c58fbabdb"
time=2026-01-10T23:18:53.074Z level=INFO source=main.go:889 msg="operational information" build.context="(go=golang 1.25.5, platform=linux/amd64, user=root@061c3a92
12:9e0: date=2026/01/16 08:09:00, os=netgo, builtinsets)" host.details="(Linux 6.14.0-1015-mes #15~24.04.1-Ubuntu SMP Tue Sep 23 22:44:48 UTC 2025 x86_64 ip-1
72-31-18-51 ({none}))" fs.freespace="soft=1048575, hard=1048576" vm.limits="soft=unlimited, hard=unlimited"
time=2026-01-10T23:18:53.078Z level=INFO source=web.go:684 msg="Start listening for connections" component=web address=0.0.0.0:9090
time=2026-01-10T23:18:53.078Z level=INFO source=main.go:1331 msg="Starting TSDB ... "
time=2026-01-10T23:18:53.078Z level=INFO source=tsdb.go:354 msg="Listening on" component=web http2=false address=:1:9090
time=2026-01-10T23:18:53.082Z level=INFO source=tsdb.go:357 msg="TLS is disabled." component=web http2=false address=:1:9090
time=2026-01-10T23:18:53.084Z level=INFO source=tsdb.go:681 msg="Replaying on-disk memory mappable chunks if any" component=tsdb
time=2026-01-10T23:18:53.084Z level=INFO source=tsdb.go:767 msg="On-disk memory mappable chunks replay completed" component=tsdb duration=1.515μs
time=2026-01-10T23:18:53.084Z level=INFO source=tsdb.go:775 msg="Replaying WAL, this may take a while" component=tsdb
time=2026-01-10T23:18:53.087Z level=INFO source=tsdb.go:848 msg="WAL segment loaded" component=tsdb segment=0 maxSegment=1 duration=2.312076ms
time=2026-01-10T23:18:53.088Z level=INFO source=tsdb.go:848 msg="WAL segment loaded" component=tsdb segment=1 maxSegment=1 duration=692.938μs
time=2026-01-10T23:18:53.088Z level=INFO source=tsdb.go:889 msg="WAL replay completed" component=tsdb checkpoint_replay_duration=31.229μs wal_replay_duration=
3.158361ms wbl_replay_duration=592ms chunk_snapshot.load.duration=0s mmap_chunk_replay.duration=1.515μs total_replay.duration=3.385478ms
time=2026-01-10T23:18:53.092Z level=INFO source=main.go:1352 msg="filesystem information" fs_type=EXT4 SUPER_MAGIC
time=2026-01-10T23:18:53.092Z level=INFO source=main.go:1355 msg="TSDB started"
time=2026-01-10T23:18:53.092Z level=INFO source=main.go:1542 msg="Loading configuration file" filename=prometheus.yml
time=2026-01-10T23:18:53.095Z level=INFO source=main.go:1582 msg="Completed loading of configuration file" db_storage=1.36μs remote_storage=1.467μs web_handle
r=1.228μs query_engine=2.42086μs scrape_sd=31.228μs notify=433.372μs notify_sd=17.99μs rules=1.909μs tracing=4.126μs filename=prometheus.yml t
otalDuration=3.267242ms
time=2026-01-10T23:18:53.095Z level=INFO source=manager.go:202 msg="Starting rule manager..." component=rule manager
time=2026-01-10T23:18:53.095Z level=INFO source=manager.go:202 msg="Starting rule manager..." component=rule manager

root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64#
```

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

Prometheus has been installed and it is running on the background.

### 3.8.4 Access Prometheus on the browser

Prometheus is running on port 9090. We can access it on the browser by using

`http://<Public IP address of Monitor Server>:9090`

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, CloudShell, Feedback, and Console Mobile App. The main area displays a table of instances. One instance, 'Monitor' (ID: i-0e74df0ad88830c9e), is selected and highlighted with a blue border. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. The 'Monitor' instance is listed as 'Running' on 't2.large' instances. Below the table, the details for the selected instance are shown. The 'Details' tab is active. Under 'Instance summary', the 'Public IPv4 address' field is highlighted with an orange arrow and contains the value '44.220.145.88'. Other fields in this section include Instance ID (i-0e74df0ad88830c9e), Instance state (Running), Hostname type (IP name: ip-172-31-18-51.ec2.internal), Answer private resource DNS name (IPv4 (A)), and Instance type (t2.large). To the right, there are sections for Private IPv4 addresses (172.31.18.51), Public DNS (ec2-44-220-145-88.compute-1.amazonaws.com), and Elastic IP addresses (empty).

Copy the Public IP address: 44.220.145.88

That is

`http://44.220.145.88:9090`

The screenshot shows the Prometheus web interface. At the top, there's a header with the Prometheus logo, a search bar containing 'Enter expression (press Shift+Enter for newlines)', and a 'Query' button. Below the header are tabs for 'Table', 'Graph', and 'Explain'. A sidebar on the right contains various icons for monitoring and configuration. The main area displays the message 'No data queried yet' and a 'Add query' button.

We are now on Prometheus browser.

### 3.8.5 Install Grafana

Let us install Grafana. Go to <https://grafana.com/grafana/download>

The screenshot shows the Grafana download page. The top navigation bar includes links for 'Products', 'Open Source', 'Solutions', 'Learn', 'Docs', 'Pricing', 'Downloads', 'Contact us', and 'Sign in'. The main content area has a yellow header with 'Grafana' and a sub-header with 'Overview', 'Grafana project', 'Grafana Alerting', and 'Download'. Below this is a section titled 'Download Grafana' with a 'Nightly Builds' button. A callout box for 'Grafana Cloud' is shown, stating: 'You can use Grafana Cloud to avoid installing, maintaining, and scaling your own instance of Grafana. Create a free account to get started, which includes free forever access to 10K series Prometheus metrics, 50GB logs, 50GB traces, & more.' Below the callout are dropdown menus for 'Version' (set to '12.3.1') and 'Edition' (set to 'Enterprise'). A note explains that the Enterprise Edition is the default and recommended edition. Under 'Release Info', there are icons for 'Linux' (highlighted with a red arrow), 'Windows', 'Mac', 'Docker', and 'Linux on ARM64'. At the bottom, there's a terminal-like box showing command-line instructions for Ubuntu and Debian:

```
Ubuntu and Debian (64 Bit) SHA256: 542374ffcc126cf81b0249a3268ba1a05ab0abdc49b2204e3323469a29addf78
sudo apt-get install -y adduser libfontconfig1 musl
wget https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
sudo dpkg -i grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
```

Select “Linux”, since we are using “Ubuntu” servers and scroll down

grafana.com/grafana/download

## Grafana

- Overview
- Grafana project
- Grafana Alerting
- Download

Release Date: December 16, 2025

Release Info:

**Linux**   **Windows**   **Mac**   **Docker**   **Linux on ARM64**

**Ubuntu and Debian (64 Bit)**   SHA256: 542374ffcc126cf81b0249a3268ba1a05ab0abdc49b2204e3323469a29addf78

```
sudo apt-get install -y adduser libfontconfig1 musl
wget https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
sudo dpkg -i grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
```

Read the Ubuntu / Debian [installation guide](#) for more information. We also provide an [APT package repository](#).

**Standalone Linux Binaries (64 Bit)**   SHA256: ecbc6fa95d86413c4547419000c459fad6f8ab1a4b8da069332fcfc647d50747

```
wget https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise_12.3.1_20271043721_linux_amd64.tar.gz
tar -zxf grafana-enterprise_12.3.1_20271043721_linux_amd64.tar.gz
```

**Red Hat, CentOS, RHEL, and Fedora (64 Bit)**   SHA256: 1b109fce4e99f36350ceb8273d506e3e4f7d45cf3842ea2c99b396c2a3ed5cb

```
sudo yum install -y https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise_12.3.1_20271043721_linux_amd64.rpm
```

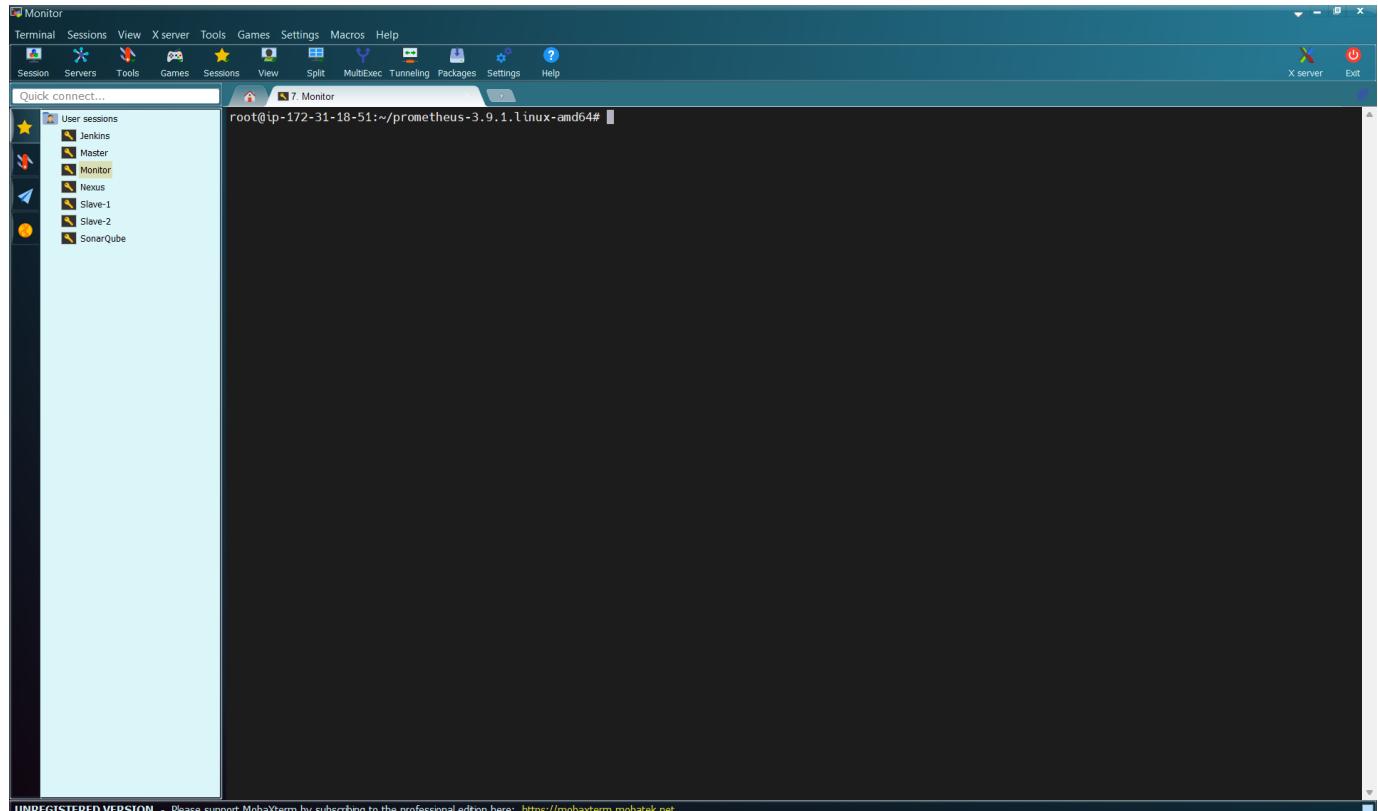
Read the Red Hat and Fedora [installation guide](#) for more information. We also provide a [YUM package repository](#).

**OpenSUSE and SUSE (64 Bit)**   SHA256: 1b109fce4e99f36350ceb8273d506e3e4f7d45cf3842ea2c99b396c2a3ed5cb

```
wget https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise_12.3.1_20271043721_linux_amd64.rpm
sudo rpm -ivh grafana-enterprise_12.3.1_20271043721_linux_amd64.rpm
```

Then, run these commands, one after another

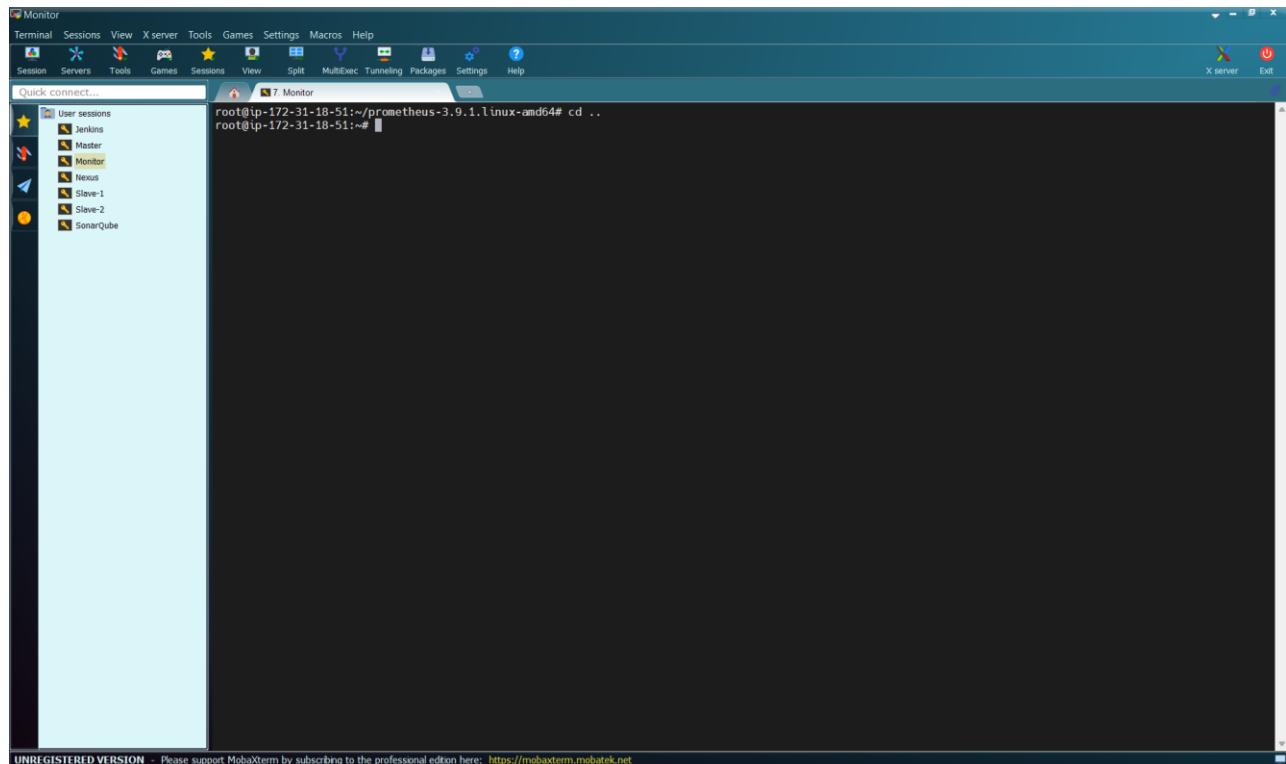
```
sudo apt-get install -y adduser libfontconfig1 musl
wget https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
sudo dpkg -i grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
```



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

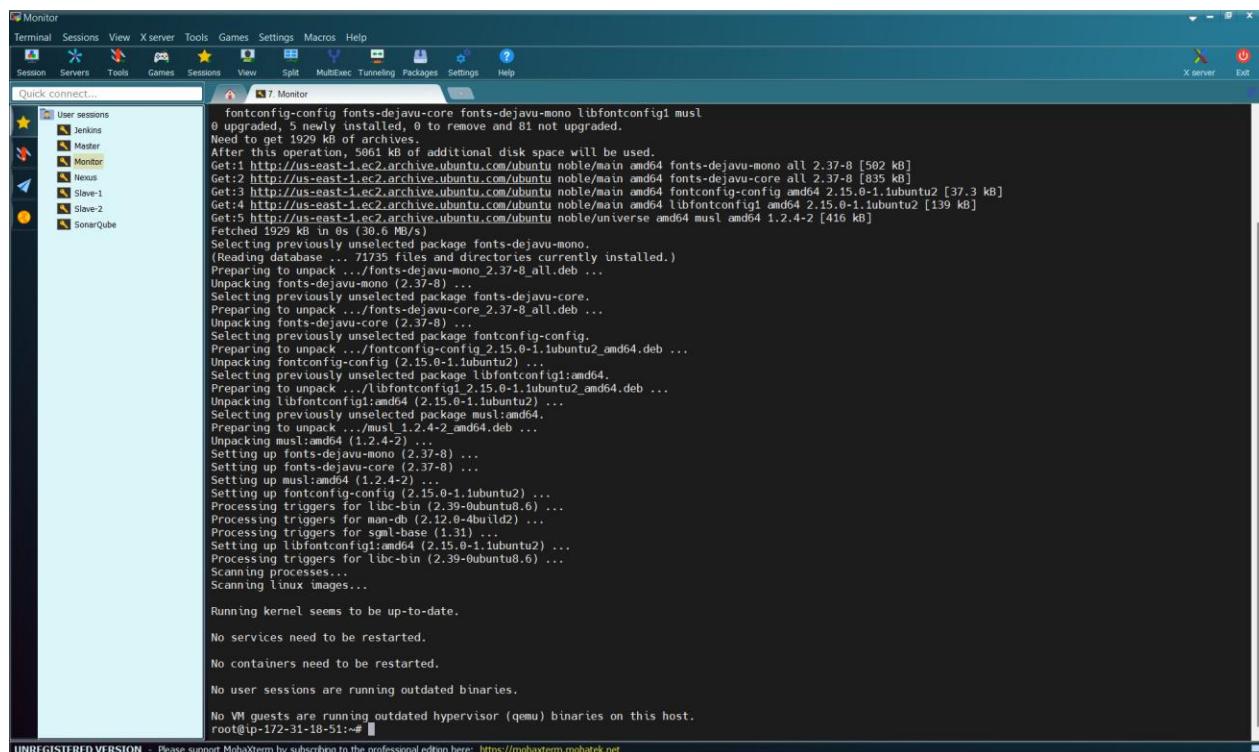
Run the command to take you back

cd ..



First run the command:

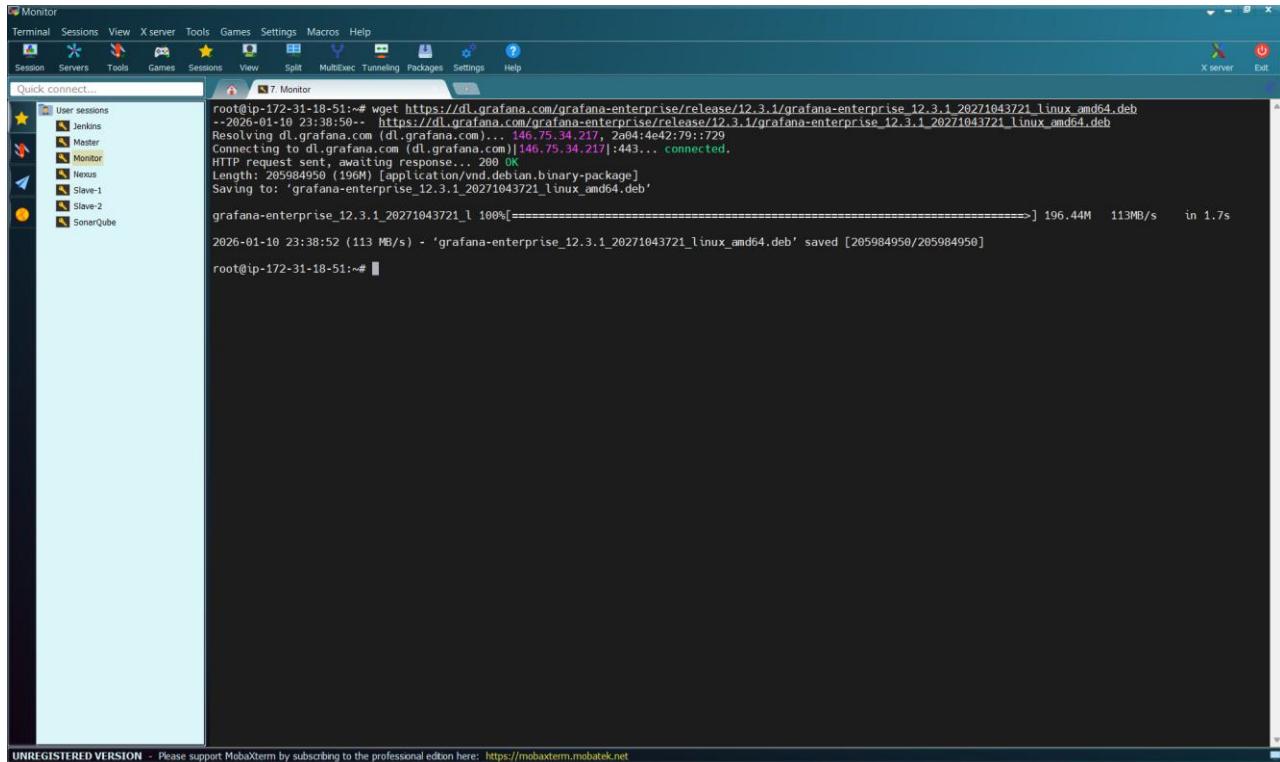
```
sudo apt-get install -y adduser libfontconfig1 musl
```



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

Then, run the command to download the Grafana:

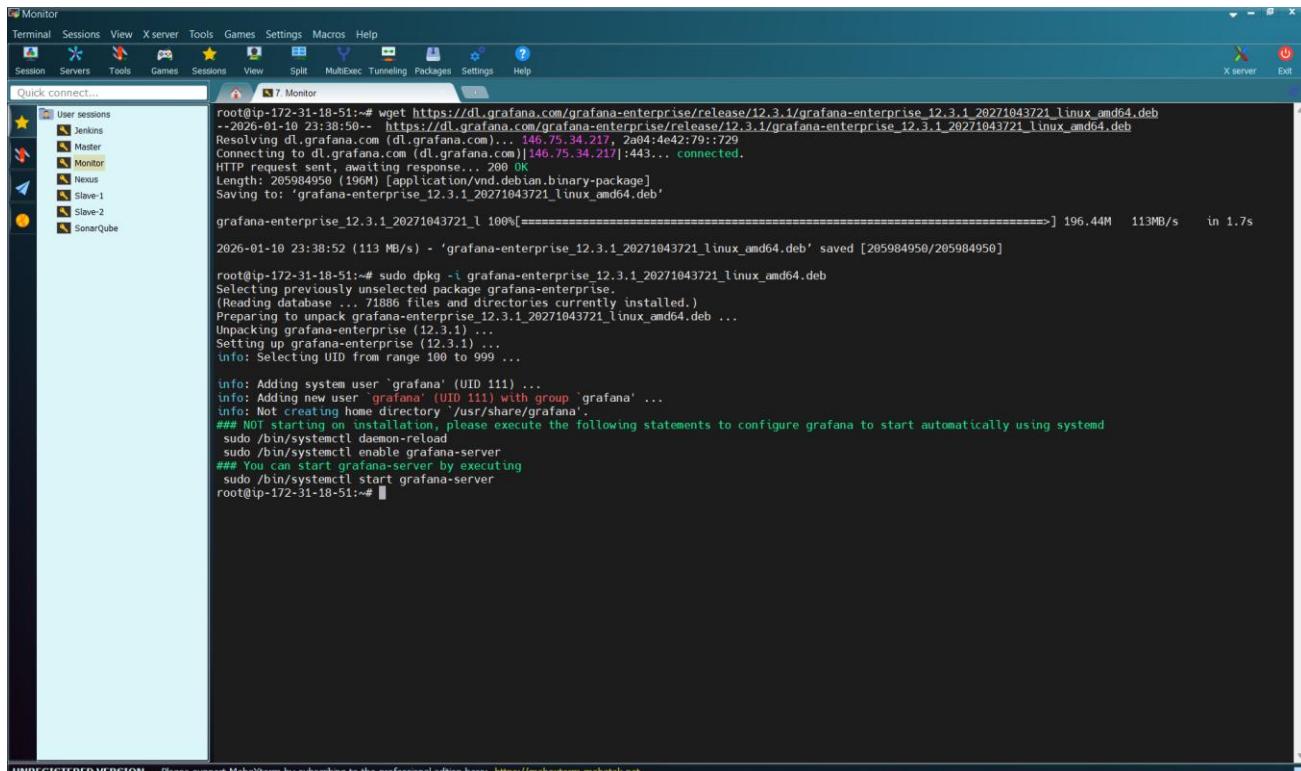
```
wget https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
```



The screenshot shows a MobaXterm window titled "Monitor". The terminal tab is active, displaying the command "wget https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise\_12.3.1\_20271043721\_linux\_amd64.deb" being executed. The output shows the progress of the download, indicating it's saving to the current directory. The session list on the left shows several other sessions like Jenkins, Master, Monitor, Nexus, Slave-1, Slave-2, and SonarQube.

Finally, run the command:

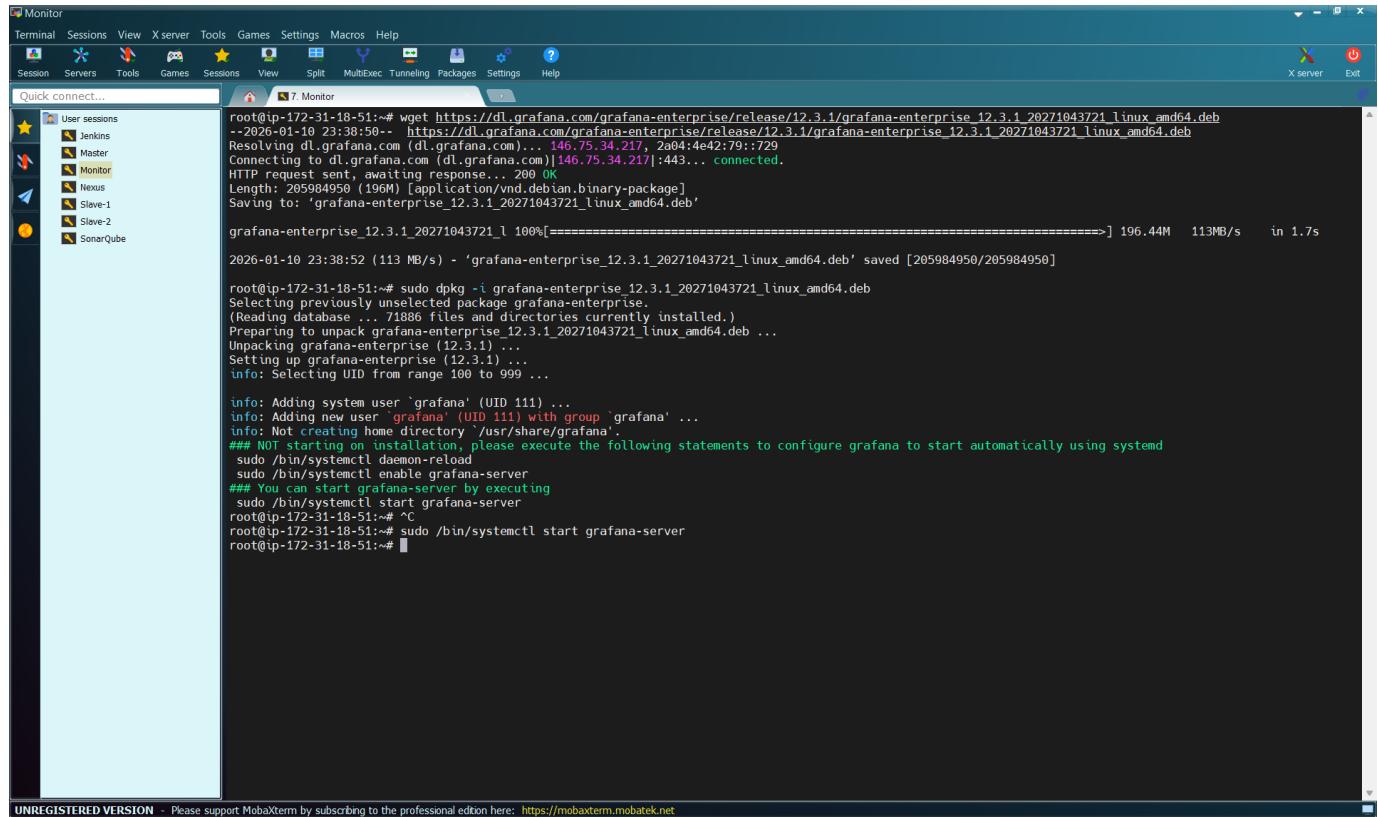
```
sudo dpkg -i grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
```



The screenshot shows a MobaXterm window titled "Monitor". The terminal tab is active, displaying the command "sudo dpkg -i grafana-enterprise\_12.3.1\_20271043721\_linux\_amd64.deb" being executed. The output shows the package being selected and unpacked. It also includes instructions for configuring Grafana to start automatically using systemd. The session list on the left shows several other sessions like Jenkins, Master, Monitor, Nexus, Slave-1, Slave-2, and SonarQube.

Then, let us start Grafana by running this command:

```
sudo /bin/systemctl start grafana-server
```



The screenshot shows a MobaXterm window titled "Monitor". The terminal session is titled "7. Monitor". The session list on the left shows "User sessions" with entries: Jenkins, Master, Monitor, Nexus, Slave-1, Slave-2, and SonarQube. The terminal window displays the following command and its output:

```
root@ip-172-31-18-51:~# wget https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
--2026-01-10 23:38:50-- https://dl.grafana.com/grafana-enterprise/release/12.3.1/grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
Resolving dl.grafana.com (dl.grafana.com)... 146.75.34.217:2a04:4e42:79::729
Connecting to dl.grafana.com (dl.grafana.com)|146.75.34.217|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 205984950 (196M) [application/vnd.debian.binary-package]
Saving to: 'grafana-enterprise_12.3.1_20271043721_linux_amd64.deb'

grafana-enterprise_12.3.1_20271043721_l 100%[=====] 196.44M 113MB/s in 1.7s

2026-01-10 23:38:52 (113 MB/s) - 'grafana-enterprise_12.3.1_20271043721_linux_amd64.deb' saved [205984950/205984950]

root@ip-172-31-18-51:~# sudo dpkg -i grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
Selecting previously unselected package grafana-enterprise.
(Reading database ... 71886 files and directories currently installed.)
Preparing to unpack grafana-enterprise_12.3.1_20271043721_linux_amd64.deb ...
Unpacking grafana-enterprise (12.3.1) ...
Setting up grafana-enterprise (12.3.1) ...
  info: Selecting UID from range 100 to 999 ...
  info: Adding system user 'grafana' (UID 111) ...
  info: Adding new user 'grafana' (UID 111) with group 'grafana' ...
  info: Not creating home directory '/usr/share/grafana'.
## NOT starting on installation, please execute the following statements to configure grafana to start automatically using systemd
  sudo /bin/systemctl daemon-reload
  sudo /bin/systemctl enable grafana-server
## You can start grafana-server by executing
  sudo /bin/systemctl start grafana-server
root@ip-172-31-18-51:~# C
root@ip-172-31-18-51:~# sudo /bin/systemctl start grafana-server
root@ip-172-31-18-51:~# ■
```

By default, Grafana will be running on Port 3000.

### 3.8.6 Access Grafana on the browser

Prometheus is running on port 3000. We can access it on the browser by using

```
http://<Public IP address of Monitor Server>:3000
```

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like Dashboard, EC2 Global View, Events, Instances (with sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager), Images, Elastic Block Store, Network & Security, and CloudShell/Feedback/Console Mobile App. The main area displays a table of instances. One instance, 'Monitor' (ID: i-0e74df0ad88830c9e), is selected and highlighted with a blue border. The 'Details' tab is active, showing instance summary information. The 'Public IP4 address' field contains '44.220.145.88' with a link to 'open address'. An orange arrow points to this specific IP address.

Copy the Public IP address: 44.220.145.88

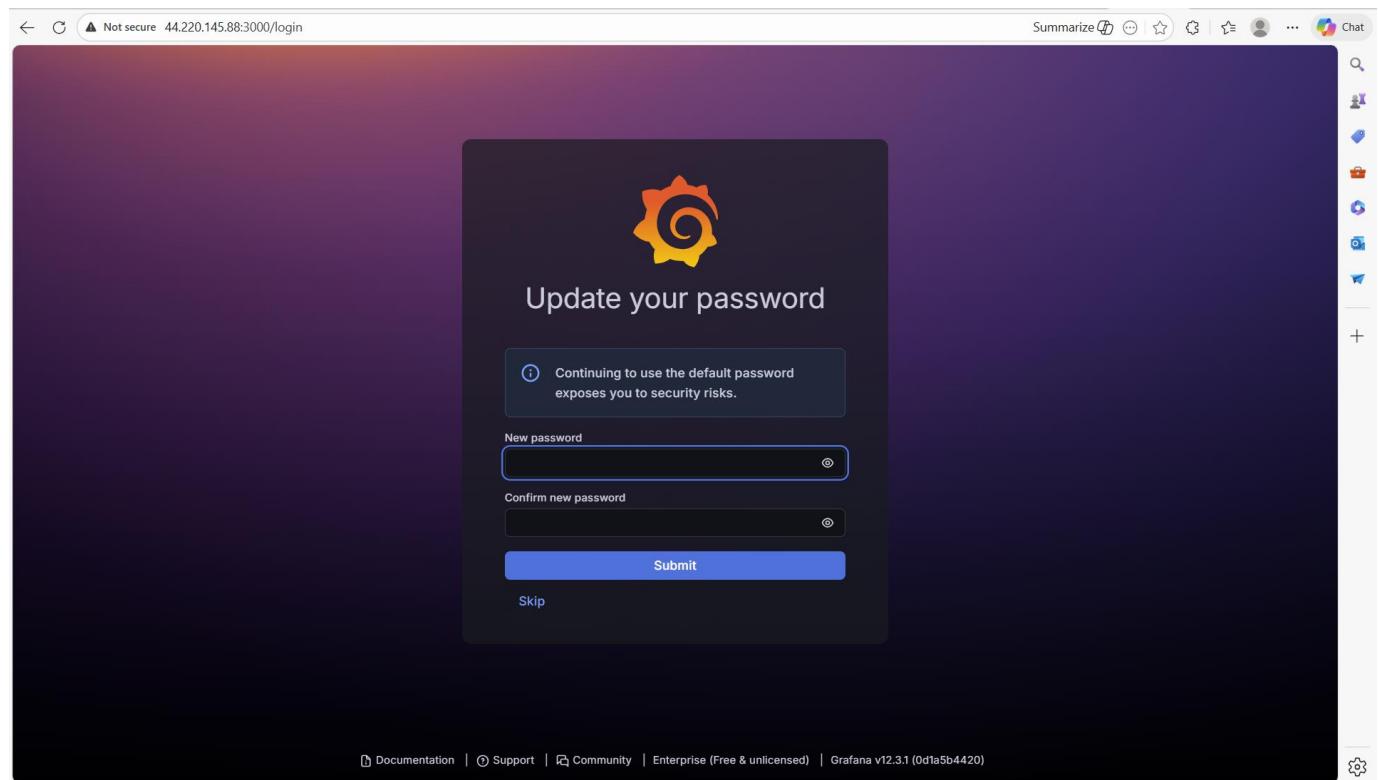
That is

<http://44.220.145.88:3000>

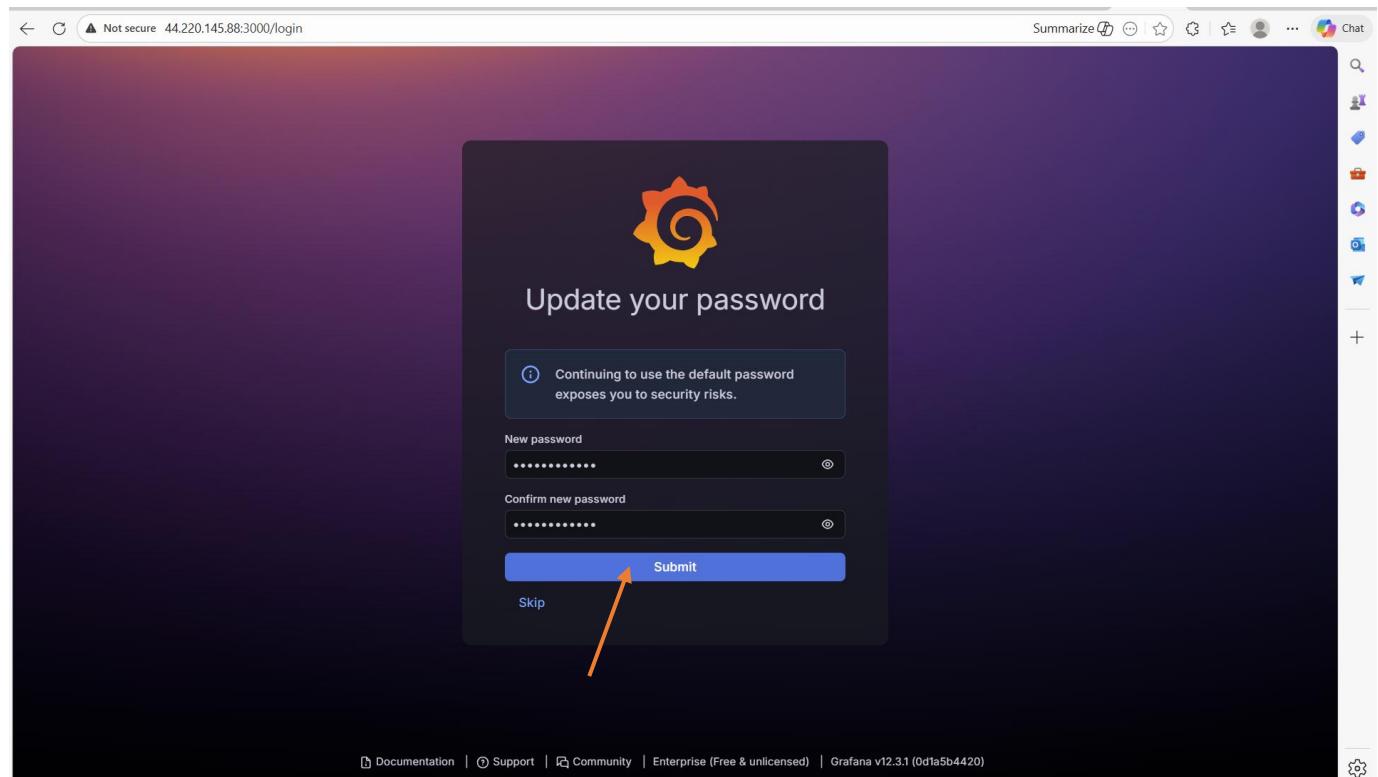
The screenshot shows a web browser window with the URL 'Not secure 44.220.145.88:3000/login'. The page has a dark background with a central light-colored login form. At the top is the Grafana logo, followed by the text 'Welcome to Grafana'. Below that is a form with two input fields: 'Email or username' and 'Password', both with placeholder text ('email or username' and 'password'). There is also a 'Log in' button and a 'Forgot your password?' link. At the bottom of the form, there are links for 'Documentation', 'Support', 'Community', 'Enterprise (Free & unlicensed)', and 'Grafana v12.3.1 (0d1a5b4420)'. The right side of the browser window shows the standard Chrome interface with tabs, a search bar, and other controls.

We are now on Grafana browser. You can see the reason why we enabled the ports from 3000 to 10,000.

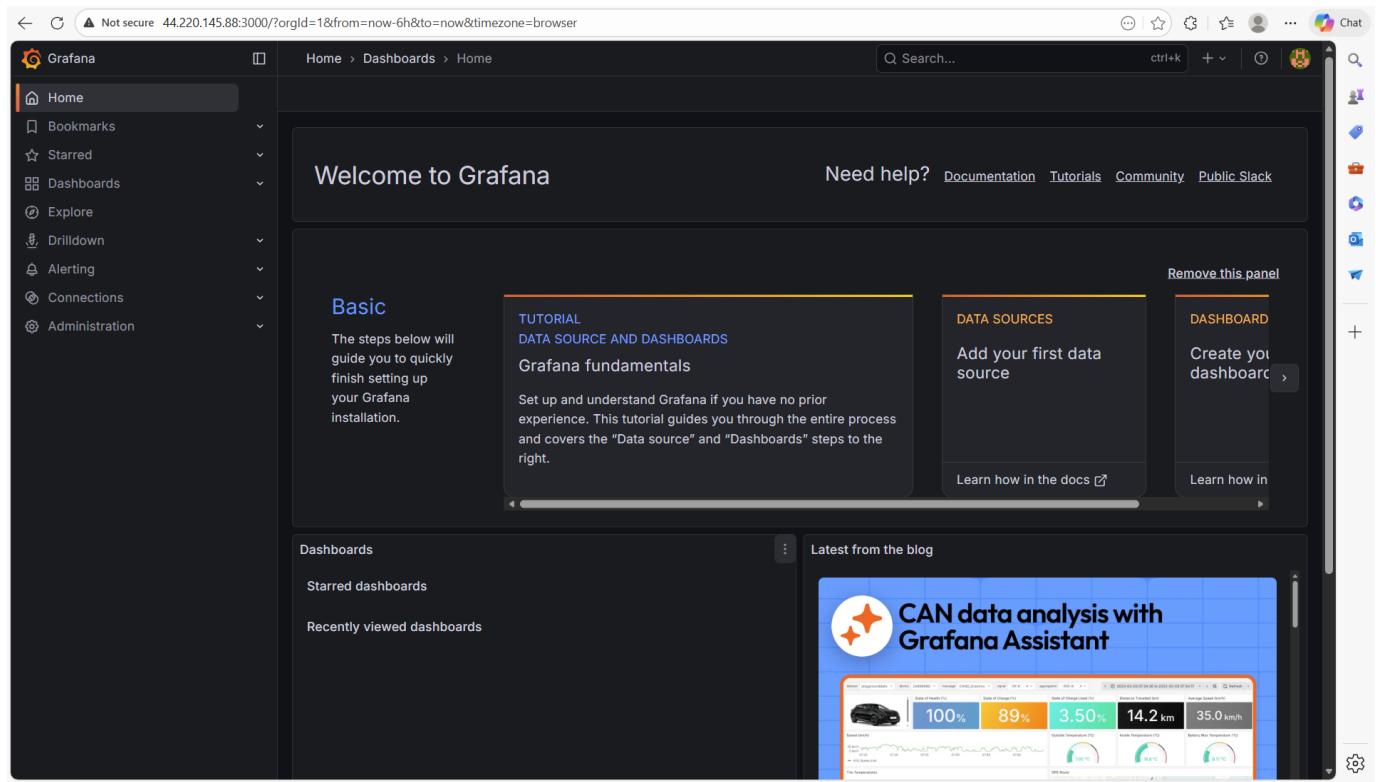
Let us login with the default username “admin” and Password “admin”



Then, enter your new password and confirm



Click on “submit”



We are now in Grafana GUI.

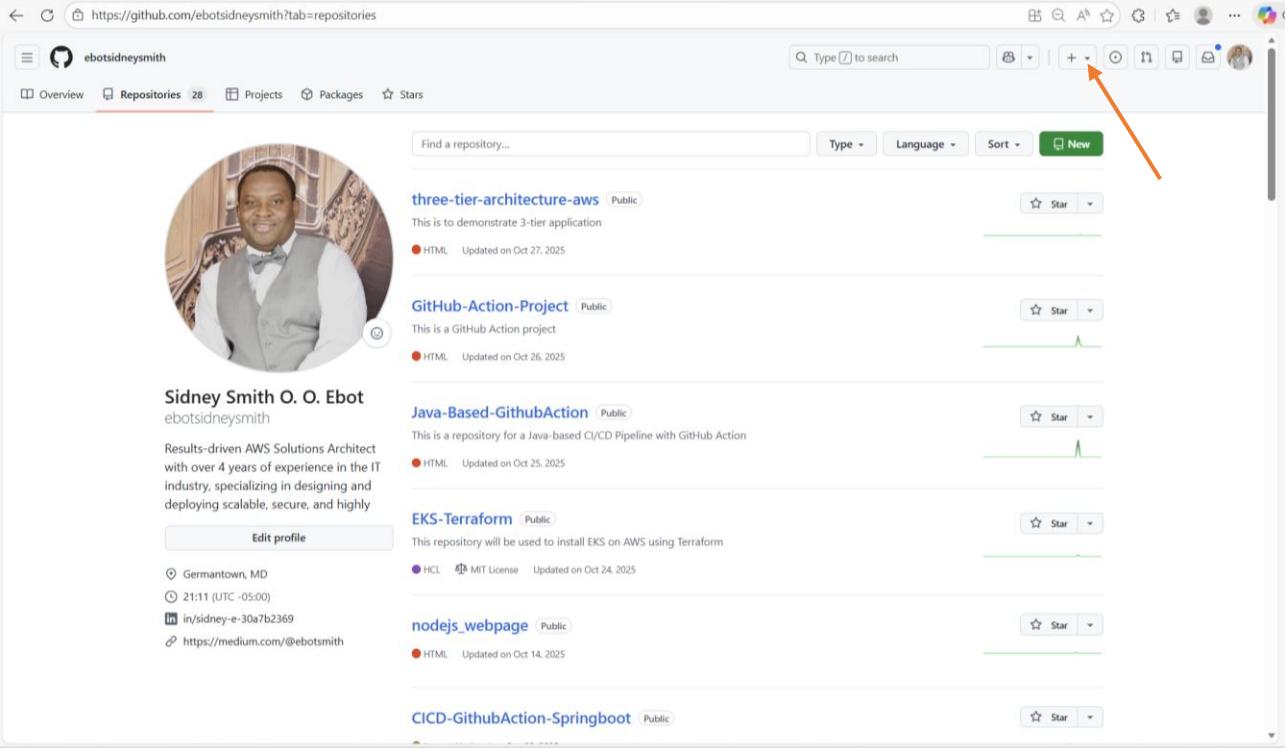
## 4 Phase 2: Private Git Set up

In this phase, we are going to create the Git repository that should be private and push our files to the repository.

### 4.1 Create a Private Git Repository

In this part, we are going to create a private Git repository so that no outside entity will be able to access it.

Log in to your GitHub account



The screenshot shows a GitHub user profile for 'ebotsidneysmith'. The profile picture is of a man in a suit. Below the profile picture, the user's name is 'Sidney Smith O. O. Ebot' and their GitHub handle is 'ebotsidneysmith'. A bio states: 'Results-driven AWS Solutions Architect with over 4 years of experience in the IT industry, specializing in designing and deploying scalable, secure, and highly available systems.' There are links to 'Edit profile', 'Gmail', 'LinkedIn', 'Medium', and 'GitHub'. The user has 28 repositories, 8 projects, and 0 packages. The 'Repositories' tab is selected. The search bar at the top right has a '+' button, which is highlighted with an orange arrow. Below the search bar, there are filters for 'Type', 'Language', and 'Sort', and a 'New' button. A list of repositories is shown:

- three-tier-architecture-aws** [Public]  
This is to demonstrate 3-tier application  
HTML Updated on Oct 27, 2025
- GitHub-Action-Project** [Public]  
This is a GitHub Action project  
HTML Updated on Oct 26, 2025
- Java-Based-GithubAction** [Public]  
This is a repository for a Java-based CI/CD Pipeline with Github Action  
HTML Updated on Oct 25, 2025
- EKS-Terraform** [Public]  
This repository will be used to install EKS on AWS using Terraform  
HCL MIT License Updated on Oct 24, 2025
- nodejs\_webpage** [Public]  
HTML Updated on Oct 14, 2025
- CICD-GithubAction-Springboot** [Public]

We will start by creating the repository. Click on “+”

The screenshot shows the GitHub profile of user ebotsidneysmith. At the top, there is a message stating "Your repository 'ebotsidneysmith/Boardgame' was successfully deleted." Below this, there is a circular profile picture of Sidney Smith O. O. Ebot. To the right of the profile picture, there is a list of recently deleted repositories:

- three-tier-architecture-aws** (Public)  
This is to demonstrate 3-tier application  
HTML Updated on Oct 27, 2025
- GitHub-Action-Project** (Public)  
This is a GitHub Action project  
HTML Updated on Oct 26, 2025
- Java-Based-GithubAction** (Public)  
This is a repository for a Java-based CI/CD Pipeline with GitHub Action  
HTML Updated on Oct 25, 2025
- EKS-Terraform** (Public)  
This repository will be used to install EKS on AWS using Terraform  
HCL MIT License Updated on Oct 24, 2025
- nodejs\_webpage** (Public)  
HTML Updated on Oct 14, 2025

In the top right corner of the page, there is a context menu with several options: "New issue", "New repository" (which is highlighted with a blue border and has an orange arrow pointing to it), "Import repository", "New codespace", "New gist", "New organization", and "New project".

## Select “New Repository”

The screenshot shows the "Create a new repository" form on GitHub. The form is divided into two main sections: "General" and "Configuration".

**General** section:

- Owner \***: ebotsidneysmith
- Repository name \***: /
- Great repository names are short and memorable. How about [fantastic-invention](#)?
- Description**: (empty)

**Configuration** section:

- Choose visibility \***: Public
- Add README**: Off
- Add .gitignore**: No .gitignore
- Add license**: No license

At the bottom of the form is a green "Create repository" button.

Give the repository a name. We will call it “Boardgame”

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

**1 General**

Owner \* ebotsidneysmith / Repository name \* Boardgame

Boardgame is available.

Great repository names are short and memorable. How about [silver-adventure](#)?

Description

0 / 350 characters

**2 Configuration**

Choose visibility \* Public

Add README Off

Add .gitignore No .gitignore

Add license No license

**Create repository**

For the “Description”, enter “Jenkins Project”

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

**1 General**

Owner \* ebotsidneysmith / Repository name \* Boardgame

Boardgame is available.

Great repository names are short and memorable. How about [silver-adventure](#)?

Description Jenkins Project

15 / 350 characters

**2 Configuration**

Choose visibility \* Public

Add README Off

Add .gitignore No .gitignore

Add license No license

**Create repository**

Then, click on the drop down on “Choose Visibility”

<https://github.com/new>

New repository

### Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

**1 General**

Owner \* ebotsidneysmith / Repository name \* Boardgame  
Boardgame is available.

Great repository names are short and memorable. How about [silver-adventure](#)?

Description Jenkins Project  
15 / 350 characters

**2 Configuration**

Choose visibility \* Public  
Choose who can see and commit to this repository

Add README  
READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore  
.gitignore tells git which files not to track. [About ignoring files](#)

Add license  
Licenses explain how others can use your code. [About licenses](#)

**Public**  
Anyone on the internet can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

No license

**Create repository**

Select “Private”

<https://github.com/new>

New repository

### Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

**1 General**

Owner \* ebotsidneysmith / Repository name \* Boardgame  
Boardgame is available.

Great repository names are short and memorable. How about [silver-adventure](#)?

Description Jenkins Project  
15 / 350 characters

**2 Configuration**

Choose visibility \* Private  
Choose who can see and commit to this repository

Add README  
READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore  
.gitignore tells git which files not to track. [About ignoring files](#)

Add license  
Licenses explain how others can use your code. [About licenses](#)

Off

No .gitignore

No license

**Create repository**

Click on “Create Repository”

**Start coding with Codespaces**  
Add a README file and start coding in a secure, configurable, and dedicated development environment.  
[Create a codespace](#)

**Add collaborators to this repository**  
Search for people using their GitHub username or email address.  
[Invite collaborators](#)

**Quick setup — if you've done this kind of thing before**

Set up in Desktop or HTTPS SSH <https://github.com/ebotsidneysmith/Boardgame.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# Boardgame" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ebotsidneysmith/Boardgame.git
git push -u origin main
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/ebotsidneysmith/Boardgame.git
git branch -M main
git push -u origin main
```

We have created our repository on GitHub.

## 4.2 Push the Source code to the repository

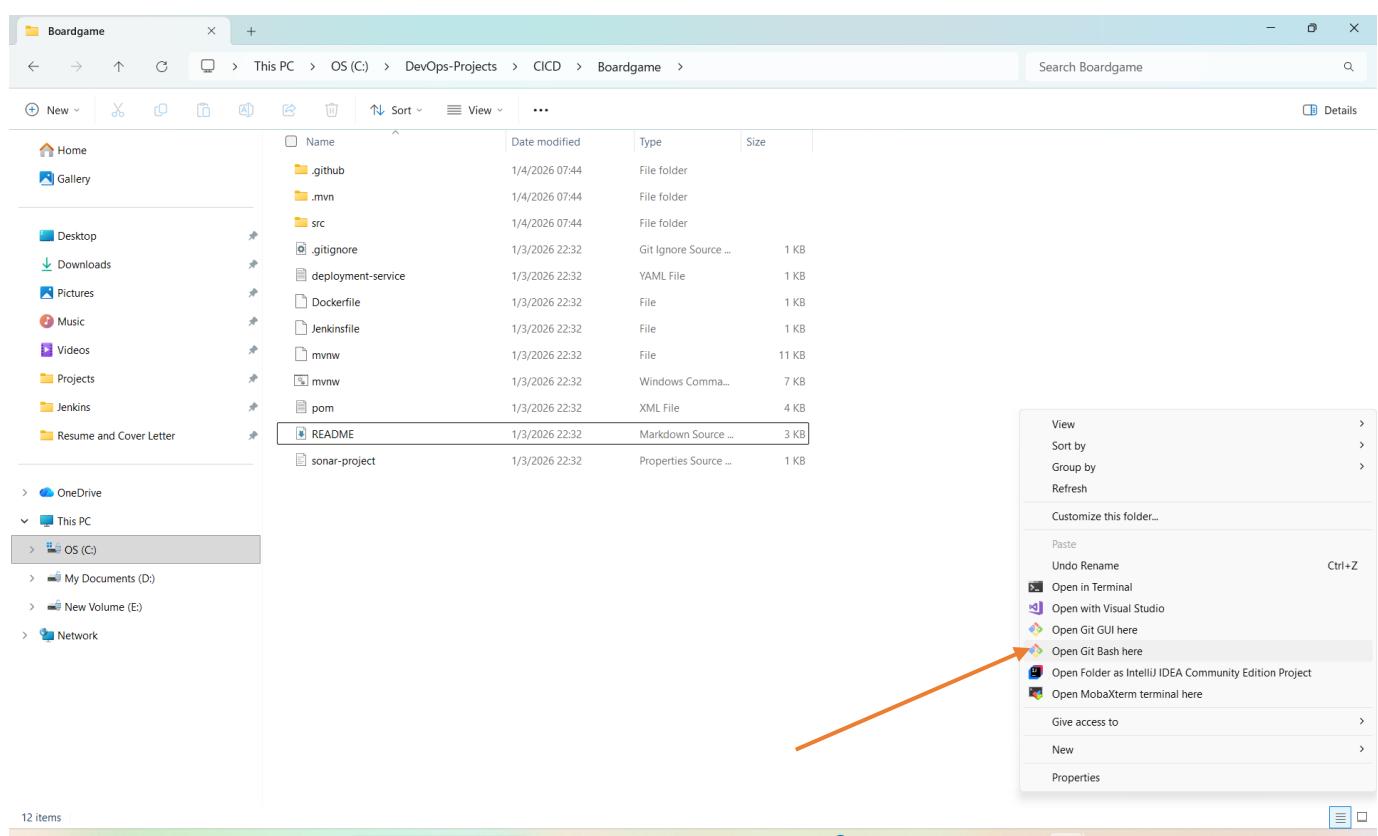
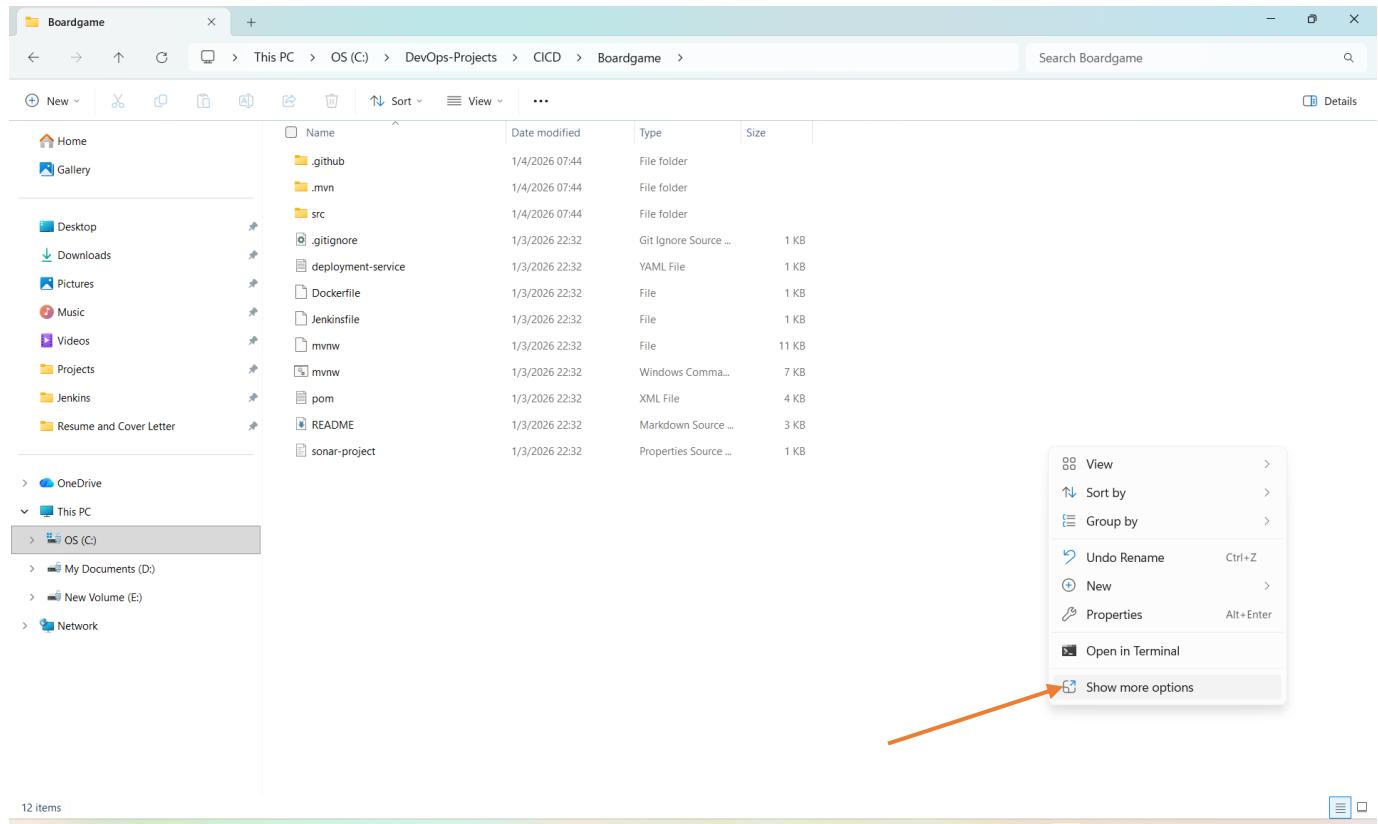
We are going to push the source code to the created repository in this part.

We have to navigate to the folder in our Local machine where we have our project files. Our source code is in this path in our local machine: C:\DevOps-Projects\CICD\Boardgame

Go to the directory

Name	Date modified	Type	Size
.github	1/4/2026 07:44	File folder	
.mvn	1/4/2026 07:44	File folder	
src	1/4/2026 07:44	File folder	
.gitignore	1/3/2026 22:32	Git Ignore Source ...	1 KB
deployment-service	1/3/2026 22:32	YAML File	1 KB
Dockerfile	1/3/2026 22:32	File	1 KB
Jenkinsfile	1/3/2026 22:32	File	1 KB
mvnw	1/3/2026 22:32	File	11 KB
mvnw	1/3/2026 22:32	Windows Comma...	7 KB
pom	1/3/2026 22:32	XML File	4 KB
README	1/3/2026 22:32	Markdown Source ...	3 KB
sonar-project	1/3/2026 22:32	Properties Source ...	1 KB

Open Git Bash from here. This is done by right-clicking here



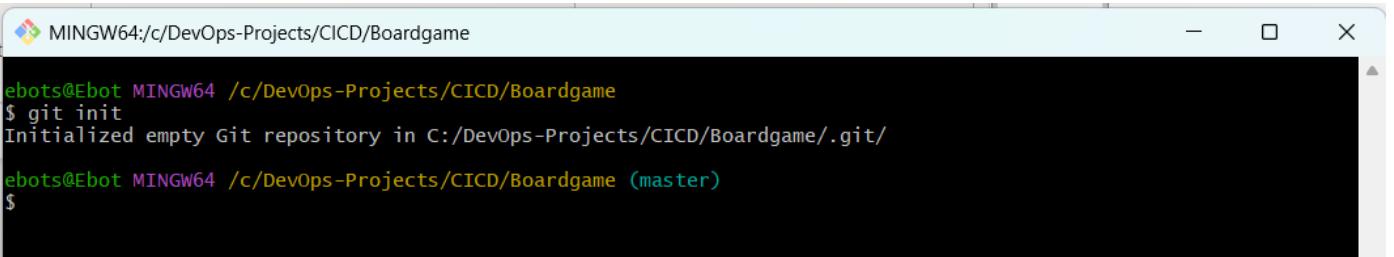


ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame-main

```
$
```

We are now in the repository in our local machine where the project files are stored. Initialize the repository in the local machine by using the command:

```
git init
```



```
ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame
```

```
$ git init
```

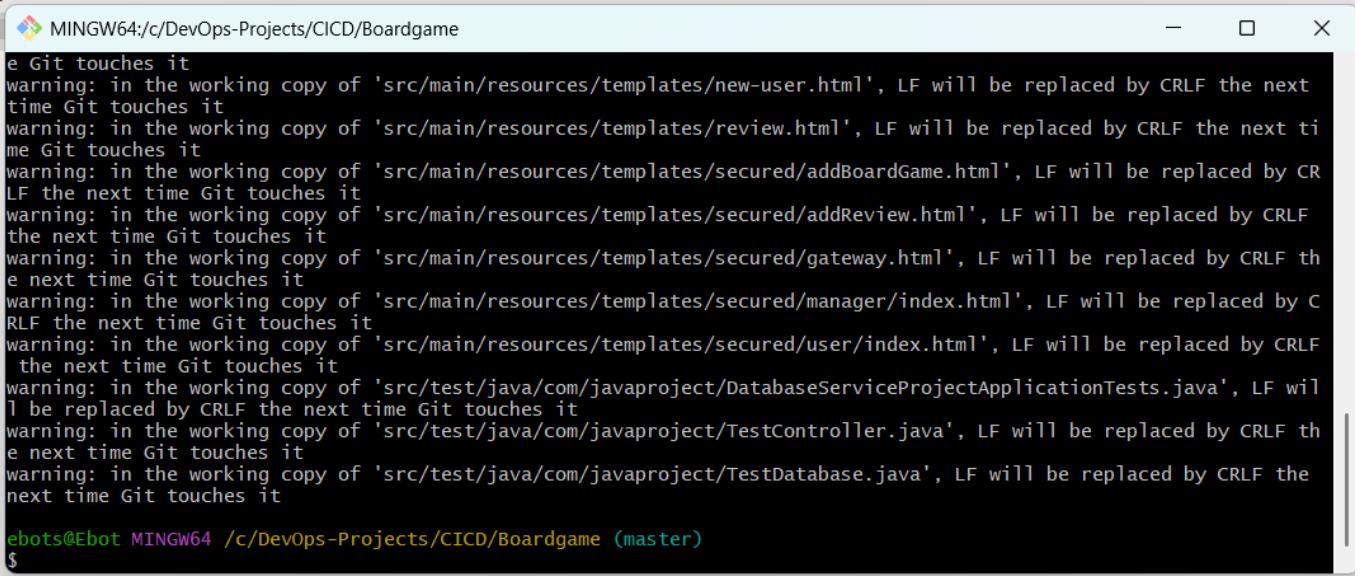
```
Initialized empty Git repository in C:/DevOps-Projects/CICD/Boardgame/.git/
```

```
ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (master)
```

```
$
```

We stage the files by using the command:

```
git add .
```



```
ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame
```

```
e Git touches it
```

```
warning: in the working copy of 'src/main/resources/templates/new-user.html', LF will be replaced by CRLF the next time Git touches it
```

```
warning: in the working copy of 'src/main/resources/templates/review.html', LF will be replaced by CRLF the next time Git touches it
```

```
warning: in the working copy of 'src/main/resources/templates/secured/addBoardGame.html', LF will be replaced by CR LF the next time Git touches it
```

```
warning: in the working copy of 'src/main/resources/templates/secured/addReview.html', LF will be replaced by CRLF the next time Git touches it
```

```
warning: in the working copy of 'src/main/resources/templates/secured/gateway.html', LF will be replaced by CRLF the next time Git touches it
```

```
warning: in the working copy of 'src/main/resources/templates/secured/manager/index.html', LF will be replaced by CRLF the next time Git touches it
```

```
warning: in the working copy of 'src/main/resources/templates/secured/user/index.html', LF will be replaced by CRLF the next time Git touches it
```

```
warning: in the working copy of 'src/test/java/com/javaproject/DatabaseServiceProjectApplicationTests.java', LF will be replaced by CRLF the next time Git touches it
```

```
warning: in the working copy of 'src/test/java/com/javaproject/TestController.java', LF will be replaced by CRLF the next time Git touches it
```

```
warning: in the working copy of 'src/test/java/com/javaproject/TestDatabase.java', LF will be replaced by CRLF the next time Git touches it
```

```
ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (master)
```

```
$
```

We check the status of the files by using the command:

```
git status
```

```
MINGW64:/c/DevOps-Projects/CICD/Boardgame
new file: src/main/resources/static/img/favicon.ico
new file: src/main/resources/static/img/spring-boot-logo.png
new file: src/main/resources/static/js/script.js
new file: src/main/resources/templates/boardgame.html
new file: src/main/resources/templates/error/permission-denied.html
new file: src/main/resources/templates/fragments/footer.html
new file: src/main/resources/templates/fragments/header.html
new file: src/main/resources/templates/fragments/links.html
new file: src/main/resources/templates/index.html
new file: src/main/resources/templates/login.html
new file: src/main/resources/templates/new-user.html
new file: src/main/resources/templates/review.html
new file: src/main/resources/templates/secured/addBoardGame.html
new file: src/main/resources/templates/secured/addReview.html
new file: src/main/resources/templates/secured/gateway.html
new file: src/main/resources/templates/secured/manager/index.html
new file: src/main/resources/templates/secured/user/index.html
new file: src/test/java/com/javaproject/DatabaseServiceProjectApplicationTests.java
new file: src/test/java/com/javaproject/TestController.java
new file: src/test/java/com/javaproject/TestDatabase.java

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (master)
$ |
```

You can see the files that have been staged but not committed. Let us now commit the files by using the command:

```
git commit -m "Initial commit"
```

```
MINGW64:/c/DevOps-Projects/CICD/Boardgame
create mode 100644 src/main/resources/static/css/style.css
create mode 100644 src/main/resources/static/img/favicon.ico
create mode 100644 src/main/resources/static/img/spring-boot-logo.png
create mode 100644 src/main/resources/static/js/script.js
create mode 100644 src/main/resources/templates/boardgame.html
create mode 100644 src/main/resources/templates/error/permission-denied.html
create mode 100644 src/main/resources/templates/fragments/footer.html
create mode 100644 src/main/resources/templates/fragments/header.html
create mode 100644 src/main/resources/templates/fragments/links.html
create mode 100644 src/main/resources/templates/index.html
create mode 100644 src/main/resources/templates/login.html
create mode 100644 src/main/resources/templates/new-user.html
create mode 100644 src/main/resources/templates/review.html
create mode 100644 src/main/resources/templates/secured/addBoardGame.html
create mode 100644 src/main/resources/templates/secured/addReview.html
create mode 100644 src/main/resources/templates/secured/gateway.html
create mode 100644 src/main/resources/templates/secured/manager/index.html
create mode 100644 src/main/resources/templates/secured/user/index.html
create mode 100644 src/test/java/com/javaproject/DatabaseServiceProjectApplicationTests.java
create mode 100644 src/test/java/com/javaproject/TestController.java
create mode 100644 src/test/java/com/javaproject/TestDatabase.java

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (master)
$ |
```

Switch to main branch using the command:

```
git branch -M main
```

```
MINGW64:/c/DevOps-Projects/CICD/Boardgame
create mode 100644 src/main/resources/static/js/script.js
create mode 100644 src/main/resources/templates/boardgame.html
create mode 100644 src/main/resources/templates/error/permission-denied.html
create mode 100644 src/main/resources/templates/fragments/footer.html
create mode 100644 src/main/resources/templates/fragments/header.html
create mode 100644 src/main/resources/templates/fragments/links.html
create mode 100644 src/main/resources/templates/index.html
create mode 100644 src/main/resources/templates/login.html
create mode 100644 src/main/resources/templates/new-user.html
create mode 100644 src/main/resources/templates/review.html
create mode 100644 src/main/resources/templates/secured/addBoardGame.html
create mode 100644 src/main/resources/templates/secured/addReview.html
create mode 100644 src/main/resources/templates/secured/gateway.html
create mode 100644 src/main/resources/templates/secured/manager/index.html
create mode 100644 src/main/resources/templates/secured/user/index.html
create mode 100644 src/test/java/com/javaproject/DatabaseServiceProjectApplicationTests.java
create mode 100644 src/test/java/com/javaproject/TestController.java
create mode 100644 src/test/java/com/javaproject/TestDatabase.java

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (master)
$ git branch -M main

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (main)
$
```

Let us add the files to our remote repository by using the command:

```
git remote add origin https://github.com/ebotsidneysmith/Boardgame.git
```

```
MINGW64:/c/DevOps-Projects/CICD/Boardgame
create mode 100644 src/main/resources/templates/fragments/footer.html
create mode 100644 src/main/resources/templates/fragments/header.html
create mode 100644 src/main/resources/templates/fragments/links.html
create mode 100644 src/main/resources/templates/index.html
create mode 100644 src/main/resources/templates/login.html
create mode 100644 src/main/resources/templates/new-user.html
create mode 100644 src/main/resources/templates/review.html
create mode 100644 src/main/resources/templates/secured/addBoardGame.html
create mode 100644 src/main/resources/templates/secured/addReview.html
create mode 100644 src/main/resources/templates/secured/gateway.html
create mode 100644 src/main/resources/templates/secured/manager/index.html
create mode 100644 src/main/resources/templates/secured/user/index.html
create mode 100644 src/test/java/com/javaproject/DatabaseServiceProjectApplicationTests.java
create mode 100644 src/test/java/com/javaproject/TestController.java
create mode 100644 src/test/java/com/javaproject/TestDatabase.java

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (master)
$ git branch -M main

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (main)
$ git remote add origin https://github.com/ebotsidneysmith/Boardgame.git

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (main)
$
```

Let us push the files to our master branch in the GitHub repository by using the command:

```
git push -u origin main
```

```
MINGW64:/c/DevOps-Projects/CICD/Boardgame
create mode 100644 src/test/java/com/javaproject/TestController.java
create mode 100644 src/test/java/com/javaproject/TestDatabase.java

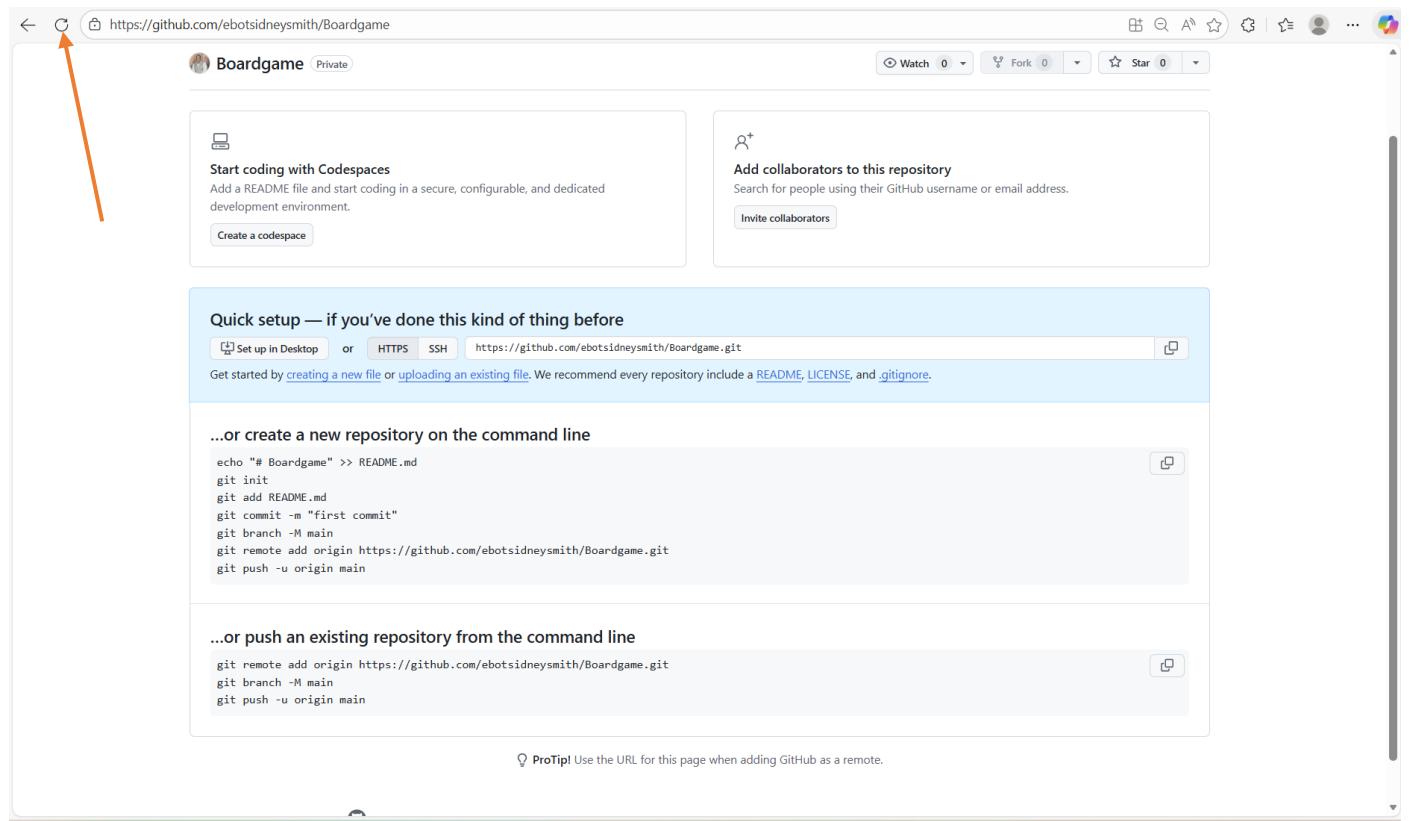
ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (master)
$ git branch -M main

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (main)
$ git remote add origin https://github.com/ebotsidneysmith/Boardgame.git

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (main)
$ git push -u origin main
Enumerating objects: 74, done.
Counting objects: 100% (74/74), done.
Delta compression using up to 16 threads
Compressing objects: 100% (61/61), done.
Writing objects: 100% (74/74), 104.68 KiB | 10.47 MiB/s, done.
Total 74 (delta 10), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (10/10), done.
To https://github.com/ebotsidneysmith/Boardgame.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

ebots@Ebot MINGW64 /c/DevOps-Projects/CICD/Boardgame (main)
$
```

Go to the GitHub page



Refresh the page

The screenshot shows a GitHub repository page for a project named 'Boardgame'. The repository is private, as indicated by the 'Private' link in the top right. At the top, there are buttons for 'Watch', 'Fork', and 'Star'. The main navigation bar includes 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. Below the navigation, there's a summary of the repository: 'main' branch, 1 branch, 0 tags, 1 commit from 'ebotsmith2000' (Initial commit, 7da1992, 7 minutes ago). A 'Go to file' search bar and an 'Add file' button are also present. The repository contains several files: .github/workflows, .mvn/wrapper, src, .gitignore, Dockerfile, Jenkinsfile, README.md, deployment-service.yaml, mvnw, mvnw.cmd, pom.xml, and sonar-project.properties. All files are listed as 'Initial commit' made 7 minutes ago. To the right of the code area, there are sections for 'About', 'Jenkins Project', 'Readme', 'Activity', 'Stars', 'Watching', and 'Forks'. Below that is a 'Releases' section with a link to 'Create a new release'. The 'Packages' section indicates 'No packages published' with a link to 'Publish your first package'. The 'Languages' section shows a breakdown: HTML 50.1%, Java 45.5%, JavaScript 3.2%, Dockerfile 1.1%, and CSS 0.1%. The README file content is displayed below the code area.

You can see that we have pushed our project file from the local machine to the GitHub repository.

### 4.3 Create a GitHub Token

Let us create a GitHub token that we will use to login to our GitHub account. Go to your GitHub account.

The screenshot shows a GitHub user profile page for 'ebotsidneysmith'. The top navigation bar includes 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. On the right side of the header, there is a user icon with an orange arrow pointing to it, indicating where to click to access the token creation interface. The main content area shows the 'Boardgame' repository details, identical to the previous screenshot. To the right of the repository, there are sections for 'About', 'Jenkins Project', 'Readme', 'Activity', 'Stars', 'Watching', and 'Forks'. Below that is a 'Releases' section with a link to 'Create a new release'. The 'Packages' section indicates 'No packages published' with a link to 'Publish your first package'. The 'Languages' section shows a breakdown: HTML 50.1%, Java 45.5%, JavaScript 3.2%, Dockerfile 1.1%, and CSS 0.1%. The README file content is displayed below the code area.

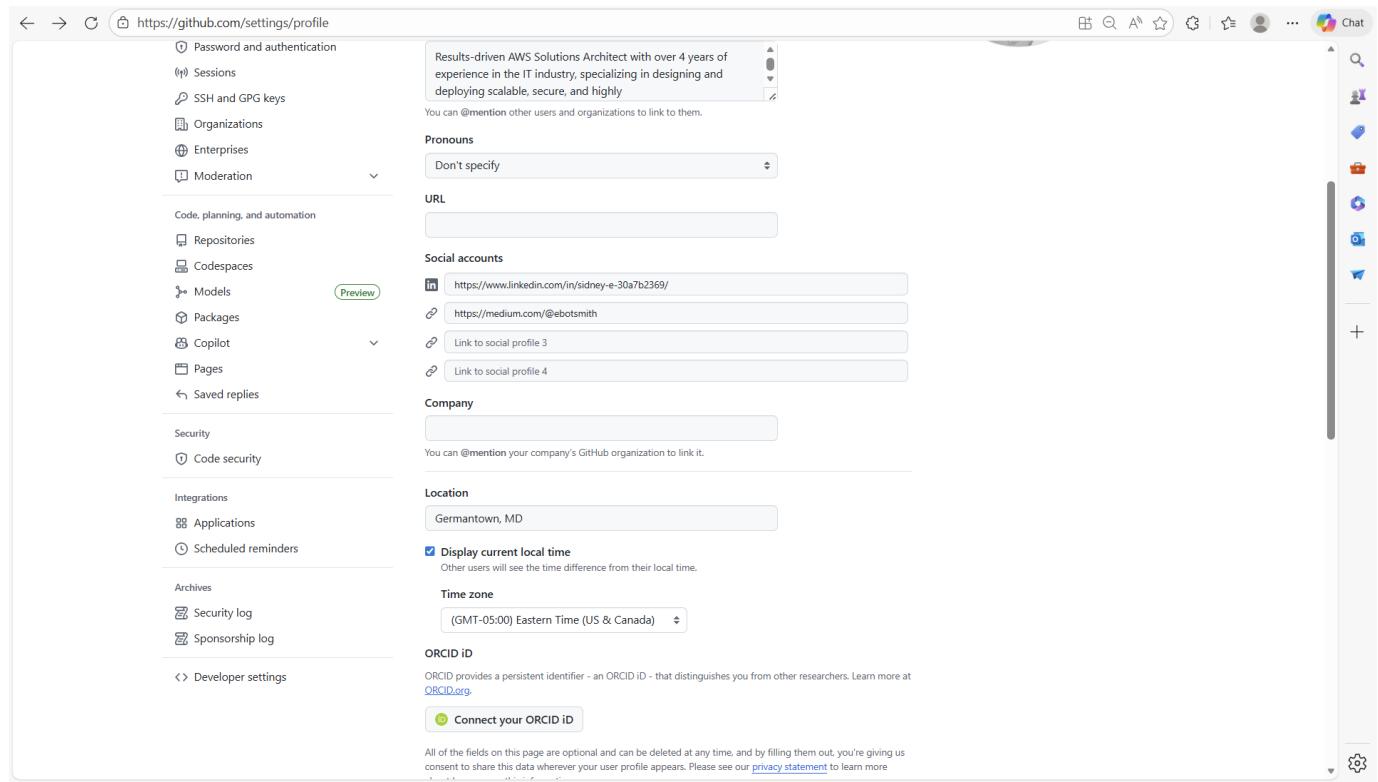
Click on the account image

A screenshot of a GitHub repository page for 'Boardgame'. The user menu is open on the right, showing various options like Profile, Repositories, and Settings. An orange arrow points to the 'Settings' option in the list.

Select “Settings”

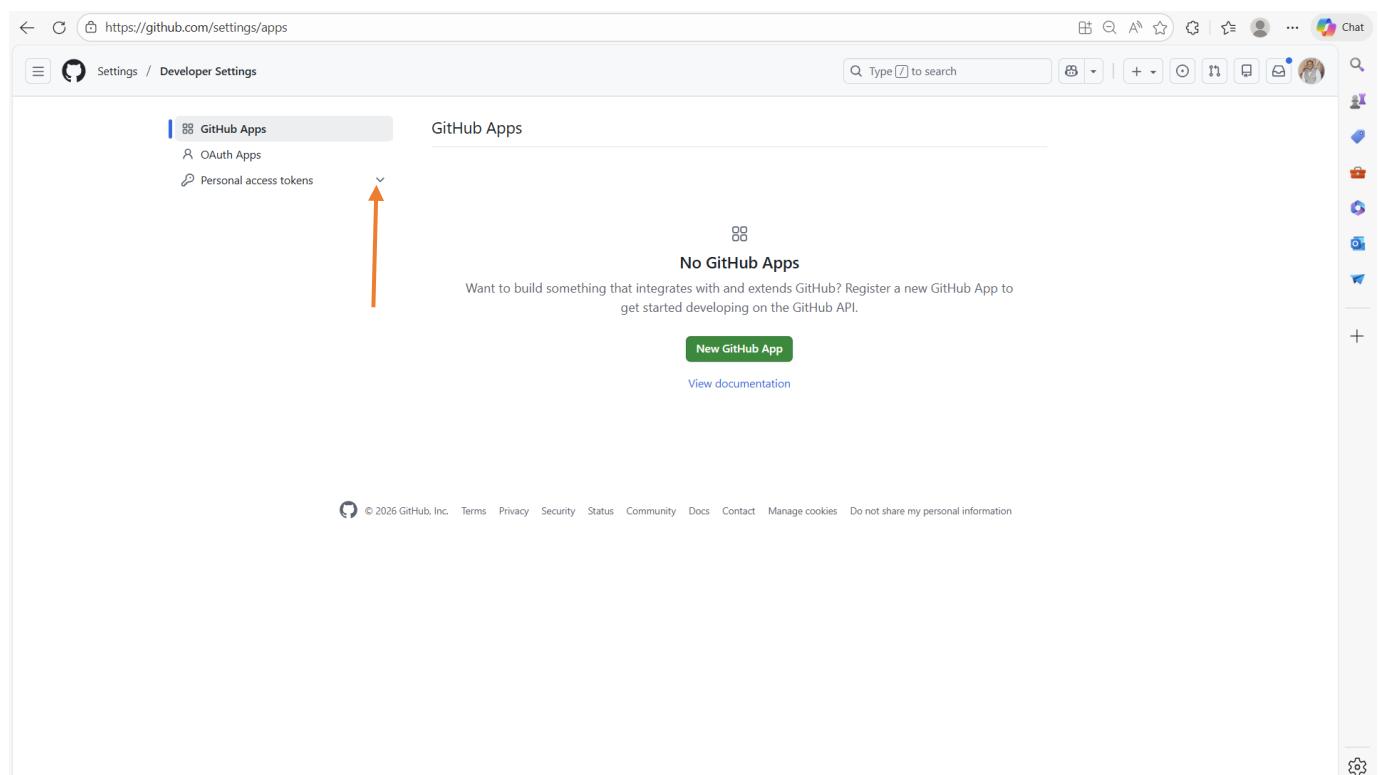
A screenshot of the GitHub Settings page for 'Sidney Smith O. O. Ebot'. The left sidebar shows various settings categories like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and licensing, Password and authentication, Sessions, SSH and GPG keys, Organizations, Enterprises, and Moderation. The 'Public profile' tab is selected. The main area displays the public profile information, including Name (Sidney Smith O. O. Ebot), Profile picture (a photo of Sidney Smith), Public email (Select a verified email to display), Bio (Results-driven AWS Solutions Architect with over 4 years of experience in the IT industry, specializing in designing and deploying scalable, secure, and highly), Pronouns (Don't specify), URL (empty), Social accounts (LinkedIn and Medium links), and Company (empty). A preview link for sessions is visible at the bottom of the sidebar.

Scroll down



The screenshot shows the GitHub Profile Settings page at <https://github.com/settings/profile>. The left sidebar contains sections like Password and authentication, Sessions, SSH and GPG keys, Organizations, Enterprises, and Moderation. Below that are Code, planning, and automation sections: Repositories, Codespaces, Models, Packages, Copilot, Pages, and Saved replies. Further down are Security, Code security, Integrations, Applications, Scheduled reminders, Archives, Security log, and Sponsorship log. At the bottom is a Developer settings section. The main area displays profile information: a bio about being a results-driven AWS Solutions Architect, pronouns set to 'Don't specify', and a URL field. It also lists social accounts (LinkedIn and Medium profiles) and a company field. Location is set to Germantown, MD. A checkbox for 'Display current local time' is checked. The Time zone is set to (GMT-05:00) Eastern Time (US & Canada). An ORCID ID section is present with a link to connect it. A note at the bottom states that fields are optional and can be deleted.

Click on “Developer Settings”



The screenshot shows the GitHub Developer Settings page at <https://github.com/settings/apps>. The left sidebar has tabs for GitHub Apps, OAuth Apps, and Personal access tokens. The GitHub Apps tab is selected. The main content area is titled "GitHub Apps" and shows a message: "No GitHub Apps". It encourages users to "Want to build something that integrates with and extends GitHub? Register a new GitHub App to get started developing on the GitHub API." A green "New GitHub App" button is visible. At the bottom, there's a copyright notice for GitHub Inc. and links for Terms, Privacy, Security, Status, Community, Docs, Contact, Manage cookies, and a "Do not share my personal information" checkbox.

Click on the drop down on “Personal Access Token”

The screenshot shows the GitHub Developer Settings page at <https://github.com/settings/apps>. The left sidebar has 'Personal access tokens' selected under 'GitHub Apps'. The main area displays the 'GitHub Apps' interface with the heading 'No GitHub Apps'. A red arrow points from the text 'Select “Token (Classic)”' to the 'Tokens (classic)' link in the sidebar.

Select “Token (Classic)”

The screenshot shows the GitHub Developer Settings page at <https://github.com/settings/tokens>. The left sidebar has 'Tokens (classic)' selected under 'Personal access tokens'. The main area shows the 'Personal access tokens (classic)' section with a 'Generate new token' button. A red arrow points from the text 'Click on “Generate New Token”' to the 'Generate new token' button.

Click on “Generate New Token”

Personal access tokens (classic)

Tokens you have generated that can be used to access the [GitHub API](#).

Generate new token

Fine-grained, repo-scoped

Generate new token (classic)

For general use

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used to authenticate to the API over Basic Authentication.

© 2026 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my personal information

<https://github.com/settings/tokens/new>

## Select “Generate new Token (classic)”

Confirm access

Signed in as @ebotsidneysmith

Password

Forgot password?

Confirm

Tip: You are entering sudo mode. After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity.

Terms Privacy Docs Contact GitHub Support Manage cookies Do not share my personal information

Enter your GitHub password and click on “Confirm”

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

What's this token for?

Expiration

30 days (Feb 03, 2026)

The token will expire on the selected date

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> <b>repo</b>	Full control of private repositories
<input type="checkbox"/> <b>repostatus</b>	Access commit status
<input type="checkbox"/> <b>repo_deployment</b>	Access deployment status
<input type="checkbox"/> <b>public_repo</b>	Access public repositories
<input type="checkbox"/> <b>repoinvite</b>	Access repository invitations
<input type="checkbox"/> <b>security_events</b>	Read and write security events
<input type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input type="checkbox"/> <b>read:packages</b>	Download packages from GitHub Package Registry
<input type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input type="checkbox"/> <b>admin:org</b>	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> <b>write:org</b>	Read and write org and team membership, read and write org projects
<input type="checkbox"/> <b>read:org</b>	Read org and team membership, read org projects
<input type="checkbox"/> <b>manage_runners:org</b>	Manage org runners and runner groups
<input type="checkbox"/> <b>admin:public_key</b>	Full control of user/public keys

Give the token a name, I will call it “mytoken”

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

mytoken

What's this token for?

Expiration

30 days (Feb 03, 2026)

The token will expire on the selected date

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> <b>repo</b>	Full control of private repositories
<input type="checkbox"/> <b>repstatus</b>	Access commit status
<input type="checkbox"/> <b>repo_deployment</b>	Access deployment status
<input type="checkbox"/> <b>public_repo</b>	Access public repositories
<input type="checkbox"/> <b>repoinvite</b>	Access repository invitations
<input type="checkbox"/> <b>security_events</b>	Read and write security events
<input type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input type="checkbox"/> <b>read:packages</b>	Download packages from GitHub Package Registry
<input type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input type="checkbox"/> <b>admin:org</b>	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> <b>write:org</b>	Read and write org and team membership, read and write org projects
<input type="checkbox"/> <b>read:org</b>	Read org and team membership, read org projects
<input type="checkbox"/> <b>manage_runners:org</b>	Manage org runners and runner groups
<input type="checkbox"/> <b>admin:public_key</b>	Full control of user/public keys

Then, check the necessary boxes. I will select everything except “delete package” and “delete repo”

The screenshot shows the GitHub Developer Settings page for creating a new personal access token. The left sidebar has sections for GitHub Apps, OAuth Apps, Personal access tokens (selected), Fine-grained tokens, and Tokens (classic). The main area is titled "New personal access token (classic)". It includes fields for Note (containing "mytoken"), what the token is for, expiration (set to 30 days), and a list of scopes. The "repo" scope is checked. Other scopes listed include workflow, write:packages, delete:packages, admin:org, and admin:public\_key. The "admin:public\_key" scope is highlighted with a blue border.

Scroll down

This screenshot shows the expanded list of scopes available for generating a token. The list includes various enterprise and organization-level permissions. An orange arrow points from the bottom of the previous screenshot down to the "Generate token" button at the bottom of this screen. The "Generate token" button is green with white text.

Click on “Generate Token”

The screenshot shows the GitHub developer settings page at <https://github.com/settings/tokens>. The user has selected the 'Personal access tokens' tab. A message at the top states: "Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved." Below this, a section titled "Personal access tokens (classic)" contains a button to "Generate new token". A note says: "Tokens you have generated that can be used to access the [GitHub API](#)." A warning message in a blue box says: "Make sure to copy your personal access token now. You won't be able to see it again!" A generated token, "ghp\_jG2b5rXHd7U6Kd1scwUeRKpaw9Wlf1zjGGf", is listed with a copy icon and a delete button. A note below explains: "Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#)." At the bottom, there's a footer with links to GitHub's Terms, Privacy, Security, Status, Community, Docs, Contact, Manage cookies, and a link to "Do not share my personal information".

Copy the token, we will use it later

ghp\_jG2b5rXHd7U6Kd1scwUeRKpaw9Wlf1zjGGf

#### 4.4 Make the Repository Visible

In this part, we will make the repository visible to us.

## 5 Phase 3: CI/CD Pipeline

In this phase, we are going to create the CI/CD pipeline. We will install the necessary plugins on Jenkins and configure the different tools.

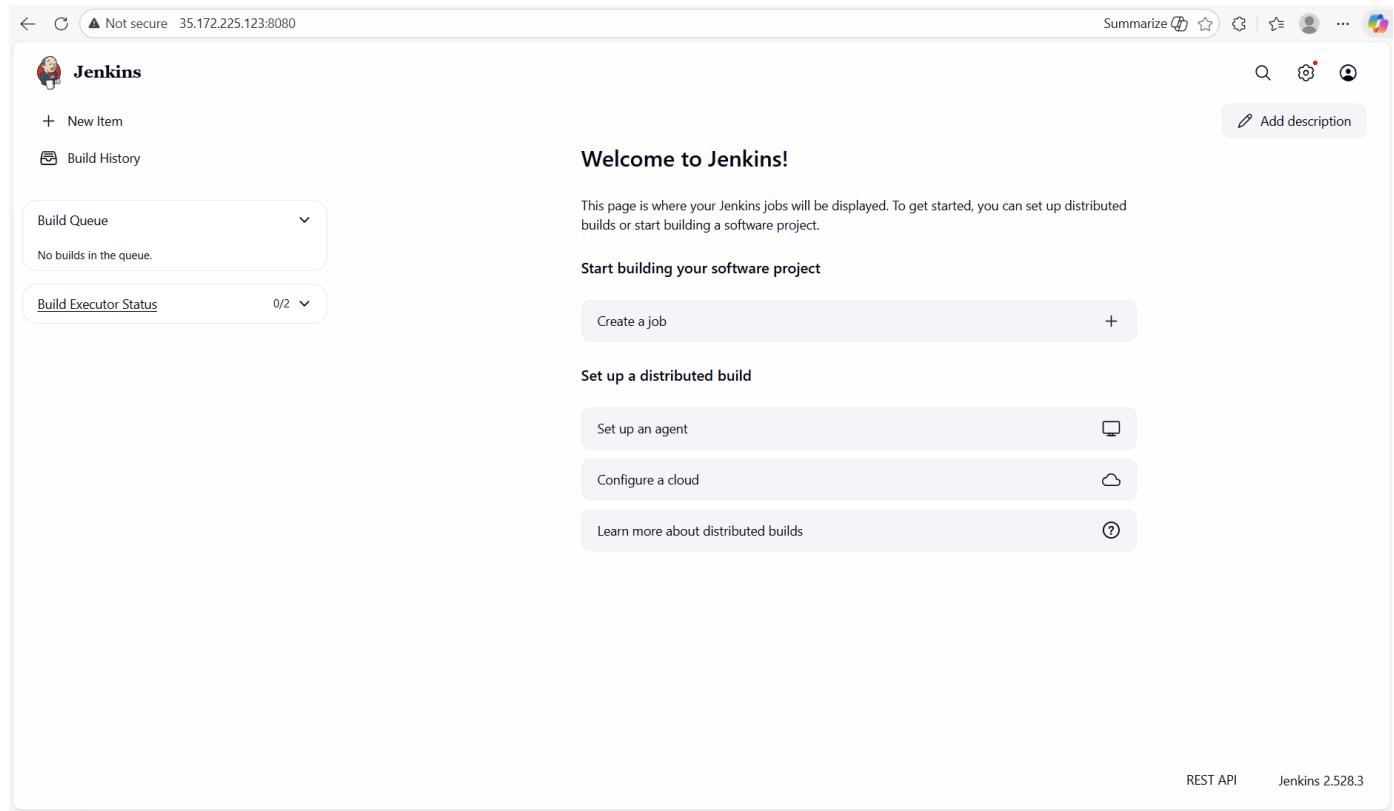
### 5.1 Install the Plugins

We are going to install these plugins: Eclipse Temurin Installer, Pipeline Maven Integration, Config File Provider, SonarQube Scanner, Kubernetes CLI, Kubernetes, Docker, Docker Pipeline Step.

#### 5.1.1 Install JDK Plugins

We are going to install the plugin “Eclipse Temurin Installer”. This plugin enables Jenkins to automatically install and configure the Eclipse Temurin JDK (formerly known as AdoptOpenJDK).

Let us install the JDK plugins. Go to the Jenkins Page



The screenshot shows the Jenkins dashboard at the URL <http://35.172.225.123:8080>. The page title is "Jenkins". On the left, there are links for "New Item" and "Build History". Below these are two dropdown menus: "Build Queue" (No builds in the queue) and "Build Executor Status" (0/2). In the center, the main content area is titled "Welcome to Jenkins!". It includes a message about starting a software project, a "Create a job" button, and a "Set up a distributed build" section with links for "Set up an agent", "Configure a cloud", and "Learn more about distributed builds". At the bottom right, there are links for "REST API" and "Jenkins 2.528.3".

Click on “**Jenkins 2.528.3**”

The screenshot shows the Jenkins home page. On the right side, there is a sidebar with various icons. An orange arrow points from the text "Click on the drop down on ‘Manage Jenkins’" to the "About Jenkins" link in the sidebar.

**Welcome to Jenkins!**

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

**Start building your software project**

[Create a job](#) +

**Set up a distributed build**

[Set up an agent](#)

[Configure a cloud](#)

[Learn more about distributed builds](#)

[About Jenkins](#)

[Get involved](#)

[Website](#)

REST API Jenkins 2.528.3

Click on “About Jenkins”

The screenshot shows the "About Jenkins" page. At the top, there is a navigation bar with links for "Jenkins", "Manage Jenkins", and "About Jenkins". Below the navigation bar is a large background image featuring the Jenkins logo character. On the right side, there is a sidebar with a "Get involved" button. The main content area displays the "Jenkins" section, which includes the version "Version 2.528.3" and a brief description: "The leading open source automation server which enables developers around the world to reliably build, test, and deploy their software." Below this, there is a table titled "Mavenized dependencies" with columns for "Name", "Maven ID", and "License". The table lists several dependencies:

Name	Maven ID	License
"Java Concurrency in Practice" book annotations	net.jcip/jcip-annotations:1.0	Creative Commons Attribution License
Annotation Indexer	org.jenkins-ciannotation-indexer:1.18	MIT License
ANTLR 4 Runtime	org.antlr:antlr4-runtime:4.13.2	BSD-3-Clause
Apache Ant Core	org.apache.ant:ant:1.10.15	The Apache Software License, Version 2.0
Apache Ant Launcher	org.apache.ant:ant-launcher:1.10.15	The Apache Software License, Version 2.0
Apache Commons BeanUtils	commons-beanutils:commons-beanutils:1.11.0	Apache-2.0

Click on the drop down on “Manage Jenkins”

Not secure 35.172.225.123:8080/manage/about/

Jenkins / Manage Jenkins / About Jenkins

Actions

- System Configuration
- System
- Tools
- Plugins
- Nodes
- Clouds
- Appearance



**Jenkins**  
Version 2.528.3

The leading open source continuous integration server for building and testing software.

Mavenized dependency management

Name ↓

Name	Version	License
"Java Concurrency in Practice" book annotations	net.jcip:jcip-annotations:1.0	Creative Commons Attribution License
Annotation Indexer	org.jenkins-ci:annotation-indexer:1.18	MIT License
ANTLR 4 Runtime	org.antlr:antlr4-runtime:4.13.2	BSD-3-Clause
Apache Ant Core	org.apache.ant:ant:1.10.15	The Apache Software License, Version 2.0
Apache Ant Launcher	org.apache.ant:ant-launcher:1.10.15	The Apache Software License, Version 2.0
commons-beanutils:commons-beanutils	1.11.0	Apache-2.0

35.172.225.123:8080/manage/pluginManager / Utilities

Get involved

Click on “Plugins”

Not secure 35.172.225.123:8080/manage/pluginManager/

Jenkins / Manage Jenkins / Plugins

Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Search plugin updates

No updates available

Disabled rows are already upgraded, awaiting restart. Shaded but selectable rows are in progress or failed.

REST API Jenkins 2.528.3

Click on “Available Plugins”

Not secure 35.172.225.123:8080/manage/pluginManager/available

## Plugins

Available plugins

Install	Name ↓	Released	Health
<input type="checkbox"/>	Pipeline Graph Analysis 245.v88f03631a_b_21 Library plugins (for use by other plugins)	3 mo 26 days ago	<span style="color: green;">97</span>
<input type="checkbox"/>	PAM Authentication 1.12 Security	10 mo ago	<span style="color: green;">91</span>
<input type="checkbox"/>	JavaMail API 1.6.2-11 Library plugins (for use by other plugins)	10 mo ago	<span style="color: green;">96</span>
<input type="checkbox"/>	Command Agent Launcher 123.v37cfdc92ef67 Agent Management	8 mo 29 days ago	<span style="color: green;">93</span>
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 83.v417146707a_3d Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	11 mo ago	<span style="color: green;">96</span>
<input type="checkbox"/>	SSH server 3.374.v19b_d59ce6610 Adds SSH server functionality to Jenkins, exposing CLI commands through it.	4 mo 27 days ago	<span style="color: green;">97</span>
<input type="checkbox"/>	Pipeline: REST API 2.38 User Interface	8 mo 9 days ago	<span style="color: green;">96</span>
<input type="checkbox"/>	JSch dependency 0.2.16-95.v3eeeb_55fa_b_78		

## Search for “JDK”

Not secure 35.172.225.123:8080/manage/pluginManager/available

Summarize ☆ ? ! \* % < > ... ...

## Plugins

Available plugins

Search: jdk

Install	Name ↓	Released	Health
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 83.v417146707a_3d Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	11 mo ago	<span style="color: green;">96</span>
<input type="checkbox"/>	JDK Parameter 1.3 Build Parameters	11 mo ago	<span style="color: orange;">63</span>
<input checked="" type="checkbox"/>	Eclipse Temurin installer 146.v1898676a_f04e Provides an installer for the JDK tool that downloads the Eclipse Temurin™ build based upon OpenJDK from the <a href="#">AdoptOpenJDK Working Group</a> .	6 mo 3 days ago	<span style="color: green;">96</span>
<input type="checkbox"/>	openJDK-native-plugin 1.8	2 yr 1 mo ago	<span style="color: orange;">75</span>
<input type="checkbox"/>	Kotlin 1.x Standard Library Plugin for JDK 8 1.3.20-1.4 Bundles Kotlin 1.x Standard Library for JDK 8	6 yr 11 mo ago	<span style="color: orange;">76</span>
<input type="checkbox"/>	graphiteIntegrator 1.2 Miscellaneous	10 yr ago	<span style="color: orange;">56</span>

## Select “Eclipse Temurin Installer”

Not secure 35.172.225.123:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Search: jdk

Install	Name	Released	Health
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 83.v417146707a_3d	Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	11 mo ago (96)
<input type="checkbox"/>	JDK Parameter 1.3	Build Parameters	11 mo ago (63)
<input checked="" type="checkbox"/>	Eclipse Temurin installer 146.v1898676a_f04e	Provides an installer for the JDK tool that downloads the Eclipse Temurin™ build based upon OpenJDK from the <a href="#">Adoptium Working Group</a> .	6 mo 3 days ago (96)
<input type="checkbox"/>	openJDK-native-plugin 1.8		2 yr 1 mo ago (75)
<input type="checkbox"/>	Kotlin 1.x Standard Library Plugin for JDK 8 1.3.20-1.4	Bundles Kotlin 1.x Standard Library for JDK 8	6 yr 11 mo ago (76)
<input type="checkbox"/>	graphiteIntegrator 1.2	Miscellaneous	10 yr ago (56)
This plugin allows you to send these metrics : number of tests, tests skipped, tests failed, build duration, cobertura total line coverage and cobertura total branch coverage to one or more graphite servers. If you don't have a graphite server you can use : <a href="https://www.hostedgraphite.com">https://www.hostedgraphite.com</a> to test. For cobertura metrics you need to install cobertura plugin and run cobertura:cobertura in goals section. Be sure to run jenkins in a Jdk 7, because the plugin only works with this version of jdk.			

REST API Jenkins 2 528.3

Click on “Install”

Not secure 35.172.225.123:8080/manage/pluginManager/updates/

Jenkins / Manage Jenkins / Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

commons-lang3 v3.x Jenkins API	Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
commons-text API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

We have installed the jdk plugins

The screenshot shows the Jenkins management interface for plugin updates. The left sidebar has links for Updates, Available plugins, Installed plugins, Advanced settings, and Download progress. The 'Download progress' link is highlighted. The main content area is titled 'Download progress' and shows a table of Jenkins API plugins and their download status. All items listed under 'Preparation' and the main list have a green checkmark and the word 'Success'.

	Preparation	
commons-lang3 v3.x Jenkins API	Success	
Ionicons API	Success	
Folders	Success	
OWASP Markup Formatter	Success	
ASM API	Success	
JSON Path API	Success	
Structs	Success	
Pipeline: Step API	Success	
commons-text API	Success	
Token Macro	Success	
Build Timeout	Success	
bouncycastle API	Success	
Credentials	Success	
Plain Credentials	Success	
Variant	Success	
SSH Credentials	Success	
Credentials Binding	Success	
SCM API	Success	
Pipeline: API	Success	
Timestamper	Success	
Caffeine API	Success	
Script Security	Success	

### 5.1.2 Install Maven Plugins

Let us install the Maven plugins. Go to the Jenkins Page

This screenshot is identical to the one above, showing the Jenkins management interface for plugin updates. The left sidebar and main content area are the same, displaying the 'Download progress' table for Jenkins API plugins. All items listed under 'Preparation' and the main list have a green checkmark and the word 'Success'.

Click on “Available Plugins”

Not secure 35.172.225.123:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

### Plugins

- Updates
- Available plugins**
- Installed plugins
- Advanced settings
- Download progress

Search available plugins

Install	Name ↓	Released	Health
<input type="checkbox"/>	Pipeline Graph Analysis 245.v88f03631a_b_21 Library plugins (for use by other plugins) Provides a REST API to access pipeline and pipeline run data.	3 mo 26 days ago	<span style="color: green;">97</span>
<input type="checkbox"/>	PAM Authentication 1.12 Security Adds Unix Pluggable Authentication Module (PAM) support to Jenkins	10 mo ago	<span style="color: green;">91</span>
<input type="checkbox"/>	JavaMail API 1.6.2-11 Library plugins (for use by other plugins) This plugin provides the JavaMail API for other plugins.	10 mo ago	<span style="color: green;">96</span>
<input type="checkbox"/>	Command Agent Launcher 123.v37cfdc92ef67 Agent Management Allows agents to be launched using a specified command.	8 mo 29 days ago	<span style="color: green;">93</span>
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 83.v417146707a_3d Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	11 mo ago	<span style="color: green;">96</span>
<input type="checkbox"/>	SSH server 3.374.v19b_d59ce6610 Adds SSH server functionality to Jenkins, exposing CLI commands through it.	4 mo 27 days ago	<span style="color: green;">97</span>
<input type="checkbox"/>	Pipeline: REST API 2.38 User Interface Provides a REST API to access pipeline and pipeline run data.	8 mo 9 days ago	<span style="color: green;">96</span>
<input type="checkbox"/>	JSch dependency 0.2.16-95.v3eeeb_55fa_b_78		

## Search for “Maven”

Not secure 35.172.225.123:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

### Plugins

- Updates
- Available plugins**
- Installed plugins
- Advanced settings
- Download progress

Search available plugins

Install	Name ↓	Released	Health
<input type="checkbox"/>	Maven Integration 3.27 Build Tools This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTS as well as the automated configuration of various Jenkins publishers such as Junit.	4 mo 25 days ago	<span style="color: green;">100</span>
<input type="checkbox"/>	Config File Provider 1006.vc7366c201f57 Groovy-related External Site/Tool Integrations Maven Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files...) loaded through the UI which will be copied to the job workspace.	1 mo 2 days ago	<span style="color: green;">100</span>
<input type="checkbox"/>	Jira 3.21 External Site/Tool Integrations Maven jira This plugin integrates Jenkins to Atlassian Jira.	18 days ago	<span style="color: green;">100</span>
<input type="checkbox"/>	Pipeline Maven Integration 1611.v6a_00c04177b_b_pipeline Maven This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path.	2 days 8 hr ago	<span style="color: green;">100</span>
<input type="checkbox"/>	Artifactory 4.0.8 pipeline This plugin allows your build jobs to deploy artifacts and resolve dependencies to and from Artifactory, and then have them linked to the build job that created them. The plugin includes a vast collection of features, including a rich pipeline API library and release management for Maven and Gradle builds with Staging and Promotion.	1 yr 5 mo ago	<span style="color: green;">82</span>

Select “**Maven Integration**”, “**Config File Provider**” and “**Pipeline Maven Integration**”

The screenshot shows the Jenkins plugin manager interface. On the left, a sidebar menu includes 'Updates', 'Available plugins' (which is selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main content area is titled 'Maven' and lists several plugins:

Install	Name	Released	Health
<input checked="" type="checkbox"/>	Maven Integration 3.27	Build Tools 5 mo 0 days ago	<span>(100)</span>
<input checked="" type="checkbox"/>	Config File Provider 1006.vc7366c201f57	Groovy-related External Site/Tool Integrations Maven 1 mo 7 days ago	<span>(100)</span>
<input type="checkbox"/>	Jira 3.21	External Site/Tool Integrations Maven jira 22 days ago	<span>(100)</span>
<input checked="" type="checkbox"/>	Pipeline Maven Integration 1611.v6a_00c04177b_b_- pipeline Maven	pipeline Maven 7 days 6 hr ago	<span>(100)</span>
<input type="checkbox"/>	Artifactory 4.0.8	pipeline This plugin allows your build jobs to deploy artifacts and resolve dependencies to and from Artifactory, and then have them linked to the build job that created them. The plugin includes a vast collection of features, including a rich pipeline API library and release management for Maven and Gradle builds with Staging and Promotion. 1 yr 6 mo ago	<span>(82)</span>

Click on “Install”

The screenshot shows the Jenkins plugin manager interface. On the left, a sidebar menu includes 'Updates', 'Available plugins' (selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main content area is titled 'Download progress' and shows the status of various plugin downloads:

Preparation	
commons-lang3 v3.x Jenkins API	<span>Success</span>
Ionicons API	<span>Success</span>
Folders	<span>Success</span>
OWASP Markup Formatter	<span>Success</span>
ASM API	<span>Success</span>
JSON Path API	<span>Success</span>
Structs	<span>Success</span>
Pipeline: Step API	<span>Success</span>
commons-text API	<span>Success</span>
Token Macro	<span>Success</span>
Build Timeout	<span>Success</span>
bouncycastle API	<span>Success</span>
Credentials	<span>Success</span>
Plain Credentials	<span>Success</span>
Variant	<span>Success</span>
SSH Credentials	<span>Success</span>
Credentials Binding	<span>Success</span>
SCM API	<span>Success</span>
Pipeline: API	<span>Success</span>
Timestamper	<span>Success</span>
Caffeine API	<span>Success</span>
Script Security	<span>Success</span>

We have installed the Maven plugins

### 5.1.3 Install Pipeline Stage View Plugins

Let us install the Pipeline Stage View plugins. Go to the Jenkins Page

The screenshot shows the Jenkins 'Manage Jenkins' section under 'Plugins'. The 'Download progress' tab is selected. A table lists various Jenkins API and utility plugins, each with a green checkmark indicating success. The listed plugins include commons-lang3 v3.x Jenkins API, Ionicons API, Folders, OWASP Markup Formatter, ASM API, JSON Path API, Struts, Pipeline: Step API, commons-text API, Token Macro, Build Timeout, bouncycastle API, Credentials, Plain Credentials, Variant, SSH Credentials, Credentials Binding, SCM API, Pipeline: API, Timestamper, Caffeine API, and Script Security.

Plugin	Status
commons-lang3 v3.x Jenkins API	Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Struts	Success
Pipeline: Step API	Success
commons-text API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

Click on “Available Plugins”

The screenshot shows the Jenkins 'Manage Jenkins' section under 'Plugins'. The 'Available plugins' tab is selected. A table lists several available Jenkins plugins, each with an 'Install' checkbox, the plugin name, its version, release date, and a green circular badge indicating the number of updates (e.g., 97). The listed plugins are: Pipeline Graph Analysis, PAM Authentication, JavaMail API, Command Agent Launcher, Oracle Java SE Development Kit Installer, SSH server, Pipeline: REST API, and JSch dependency.

Install	Name	Released	Health
<input type="checkbox"/>	Pipeline Graph Analysis 245.v88f03631a_b_21	3 mo 26 days ago	(97)
<input type="checkbox"/>	Library plugins (for use by other plugins)		
<input type="checkbox"/>	PAM Authentication 1.12	10 mo ago	(91)
<input type="checkbox"/>	Security		
<input type="checkbox"/>	Adds Unix Pluggable Authentication Module (PAM) support to Jenkins		
<input type="checkbox"/>	JavaMail API 1.6.2-11	10 mo ago	(96)
<input type="checkbox"/>	Library plugins (for use by other plugins)		
<input type="checkbox"/>	This plugin provides the JavaMail API for other plugins.		
<input type="checkbox"/>	Command Agent Launcher 123.v37cfcc92ef67	8 mo 29 days ago	(93)
<input type="checkbox"/>	Agent Management		
<input type="checkbox"/>	Allows agents to be launched using a specified command.		
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 83.v417146707a_3d	11 mo ago	(96)
<input type="checkbox"/>	Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.		
<input type="checkbox"/>	SSH server 3.374.v19b_d59ce6610	4 mo 27 days ago	(97)
<input type="checkbox"/>	Adds SSH server functionality to Jenkins, exposing CLI commands through it.		
<input type="checkbox"/>	Pipeline: REST API 2.38	8 mo 9 days ago	(96)
<input type="checkbox"/>	User Interface		
<input type="checkbox"/>	Provides a REST API to access pipeline and pipeline run data.		
<input type="checkbox"/>	JSch dependency 0.2.16-95.v3eeccb_55fa_b_78		

Then, search for “view”

Not secure 35.172.225.123:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

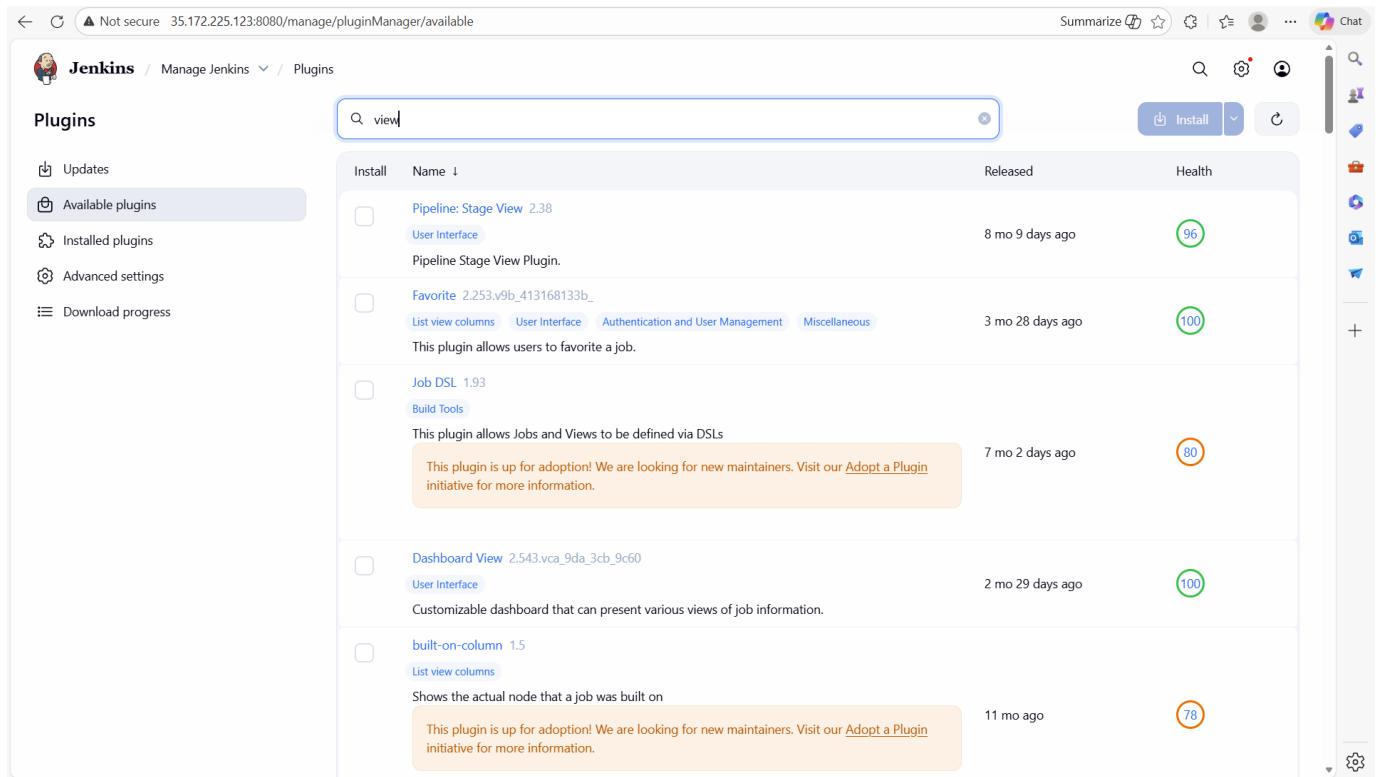
Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Install Name ↓ Released Health

Install	Name ↓	Released	Health
<input type="checkbox"/>	Pipeline: Stage View 2.38 User Interface Pipeline Stage View Plugin.	8 mo 9 days ago	96
<input type="checkbox"/>	Favorite 2.253.v9b_413168133b_ List view columns User Interface Authentication and User Management Miscellaneous	3 mo 28 days ago	100
<input type="checkbox"/>	Job DSL 1.93 Build Tools This plugin allows Jobs and Views to be defined via DSLs This plugin is up for adoption! We are looking for new maintainers. Visit our <a href="#">Adopt a Plugin</a> initiative for more information.	7 mo 2 days ago	80
<input type="checkbox"/>	Dashboard View 2.543.vca_9da_3cb_9c60 User Interface Customizable dashboard that can present various views of job information.	2 mo 29 days ago	100
<input type="checkbox"/>	built-on-column 1.5 List view columns Shows the actual node that a job was built on This plugin is up for adoption! We are looking for new maintainers. Visit our <a href="#">Adopt a Plugin</a> initiative for more information.	11 mo ago	78

Install



This screenshot shows the Jenkins plugin manager interface. The left sidebar has links for Updates, Available plugins (which is selected), Installed plugins, Advanced settings, and Download progress. The main area shows a table of available plugins. The first plugin listed is 'Pipeline: Stage View 2.38', which is selected for installation, indicated by a checked checkbox in the 'Install' column. The table includes columns for Install, Name, Released, and Health. The 'Health' column shows a green circle with the number 96. The 'Install' button is highlighted with a blue background and white text. A red arrow points from the text 'Select "Pipeline: Stage View"' to the 'Install' button.

Not secure 35.172.225.123:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

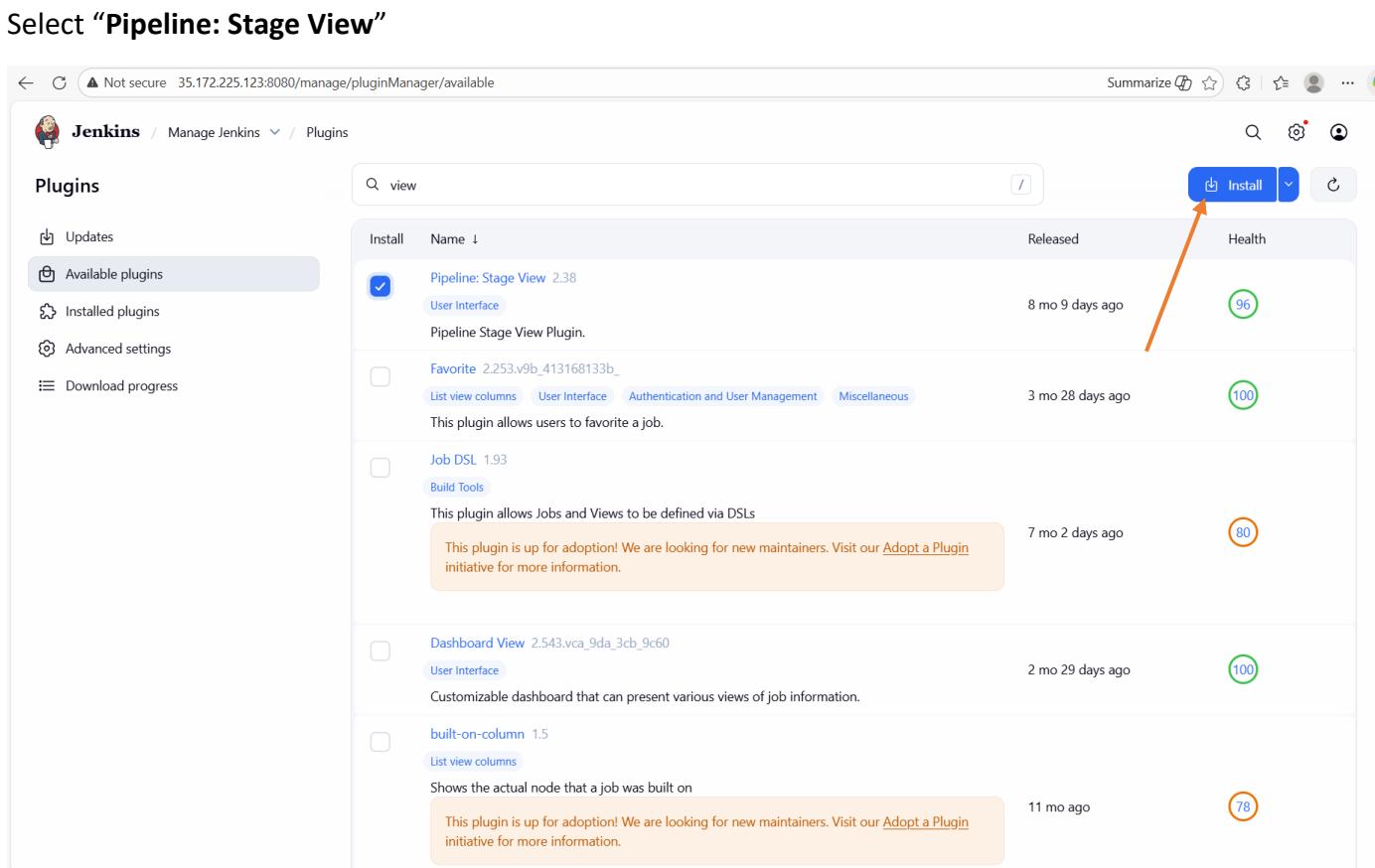
Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Install Name ↓ Released Health

Install	Name ↓	Released	Health
<input checked="" type="checkbox"/>	Pipeline: Stage View 2.38 User Interface Pipeline Stage View Plugin.	8 mo 9 days ago	96
<input type="checkbox"/>	Favorite 2.253.v9b_413168133b_ List view columns User Interface Authentication and User Management Miscellaneous	3 mo 28 days ago	100
<input type="checkbox"/>	Job DSL 1.93 Build Tools This plugin allows Jobs and Views to be defined via DSLs This plugin is up for adoption! We are looking for new maintainers. Visit our <a href="#">Adopt a Plugin</a> initiative for more information.	7 mo 2 days ago	80
<input type="checkbox"/>	Dashboard View 2.543.vca_9da_3cb_9c60 User Interface Customizable dashboard that can present various views of job information.	2 mo 29 days ago	100
<input type="checkbox"/>	built-on-column 1.5 List view columns Shows the actual node that a job was built on This plugin is up for adoption! We are looking for new maintainers. Visit our <a href="#">Adopt a Plugin</a> initiative for more information.	11 mo ago	78

Install



This screenshot shows the Jenkins plugin manager interface, identical to the one above but with a red arrow pointing from the text 'Click on "Install"' to the 'Install' button in the top right corner of the table header. The 'Install' button is highlighted with a blue background and white text.

Click on "Install"

The screenshot shows the Jenkins management interface for plugin updates. The left sidebar has links for Updates, Available plugins (which is selected), Installed plugins, Advanced settings, and Download progress. The main content area is titled 'Download progress' and shows a table of available Jenkins API plugins. Each plugin entry includes the plugin name, a status column with a green checkmark and 'Success', and a 'Preparation' section with three bullet points: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'.

Plugin	Status
commons-lang3 v3.x Jenkins API	Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
commons-text API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

We have installed the Pipeline Stage View plugins

#### 5.1.4 Install SonarQube Plugins

SonarQube is a code quality and security analysis tool. This plugin integrates Jenkins with SonarQube by providing a scanner that analyzes code during builds.

Let us install the SonarQube plugins. Go to the Jenkins Page

This screenshot shows the Jenkins management interface for plugin updates, specifically the 'Available plugins' section. The sidebar on the left has links for Updates, Available plugins (which is selected and highlighted in blue), Installed plugins, Advanced settings, and Download progress. The main content area is titled 'Download progress' and shows a table of available Jenkins API plugins. The 'Available plugins' link in the sidebar is highlighted with an orange arrow. The table structure is identical to the one in the previous screenshot, showing successful download progress for various Jenkins API plugins.

## Click on “Available Plugins”

The screenshot shows the Jenkins Manage Jenkins interface with the Plugins section selected. The 'Available plugins' tab is highlighted. A search bar at the top right contains the placeholder 'Search available plugins'. Below the search bar is an 'Install' button. The main area displays a table of available plugins, each with an 'Install' checkbox, the plugin name, its version, release date, and a green circular badge indicating its health status.

Install	Name ↓	Released	Health
<input type="checkbox"/>	PAM Authentication 1.12 Security Adds Unix Pluggable Authentication Module (PAM) support to Jenkins	10 mo ago	<span style="color: green;">91</span>
<input type="checkbox"/>	JavaMail API 1.6.2-11 Library plugins (for use by other plugins) This plugin provides the JavaMail API for other plugins.	10 mo ago	<span style="color: green;">96</span>
<input type="checkbox"/>	Command Agent Launcher 123.v37cfcd92ef67 Agent Management Allows agents to be launched using a specified command.	8 mo 29 days ago	<span style="color: green;">93</span>
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 83.v417146707a_3d Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	11 mo ago	<span style="color: green;">96</span>
<input type="checkbox"/>	SSH server 3.374.v19b_d59ce6610 Adds SSH server functionality to Jenkins, exposing CLI commands through it.	4 mo 27 days ago	<span style="color: green;">97</span>
<input type="checkbox"/>	JSch dependency 0.2.16-95.v3eeccb_55fa_b_78 Library plugins (for use by other plugins) / Miscellaneous Jenkins plugin that brings the JSch library as a plugin dependency, and provides an SSHAuthenticatorFactory for using JSch with the ssh-credentials plugin.	10 mo ago	<span style="color: green;">93</span>
<input type="checkbox"/>	Authentication Tokens API 1.144.v5ff4a_5ec5c3 This plugin provides an API for converting credentials into authentication tokens in Jenkins.	5 mo 2 days ago	<span style="color: green;">100</span>
<input type="checkbox"/>	Javadoc 354.vee1a_660b_4990 This plugin adds Javadoc support to Jenkins.	5 mo 5 days ago	<span style="color: green;">100</span>

## Search for “Sonar”

The screenshot shows the Jenkins Manage Jenkins interface with the Plugins section selected. A search bar at the top right contains the term 'sonar'. Below the search bar is an 'Install' button. The main area displays a table of available plugins related to Sonar, each with an 'Install' checkbox, the plugin name, its version, release date, and a green circular badge indicating its health status. One plugin, 'Quality Gates 2.5', has a red warning box over it, stating: 'Warning: This plugin version may not be safe to use. Please review the following security notices: • Credentials transmitted in plain text'.

Install	Name ↓	Released	Health
<input type="checkbox"/>	SonarQube Scanner 2.18.2 External Site/Tool Integrations / Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	24 days ago	<span style="color: green;">84</span>
<input type="checkbox"/>	Sonar Quality Gates 364.v67a_f255f340f Library plugins (for use by other plugins) / analysis / Other Post-Build Actions Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")	18 days ago	<span style="color: green;">100</span>
<input type="checkbox"/>	Quality Gates 2.5 Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed") Warning: This plugin version may not be safe to use. Please review the following security notices: • Credentials transmitted in plain text	9 yr 7 mo ago	<span style="color: red;">42</span>
<input type="checkbox"/>	Sonargraph Integration 5.0.2 External Site/Tool Integrations / Build Reports / Other Post-Build Actions This plugin integrates Sonargraph functionality into Jenkins, for Sonargraph versions 9 and 10	2 yr 6 mo ago	<span style="color: green;">100</span>
<input type="checkbox"/>	CodeSonar 3.6.0 DevOps / Build Notifiers / Build Reports / Other Post-Build Actions A plugin that integrates with the CodeSonar static analyzer.	7 mo 22 days ago	<span style="color: green;">97</span>

Select “SonarQube Scanner”, this is a tool we are going to use for analysis.

Jenkins / Manage Jenkins / Plugins

Plugins

Updates Available plugins (selected) Installed plugins Advanced settings Download progress

Search: sonar

Install	Name	Released	Health
<input checked="" type="checkbox"/>	SonarQube Scanner 2.18.2 External Site/Tool Integrations Build Reports	24 days ago	<span>84</span>
<input type="checkbox"/>	Sonar Quality Gates 364.v67a_f255f340f Library plugins (for use by other plugins) analysis Other Post-Build Actions	18 days ago	<span>100</span>
<input type="checkbox"/>	Quality Gates 2.5 Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")  Warning: This plugin version may not be safe to use. Please review the following security notices: <ul style="list-style-type: none"><li>Credentials transmitted in plain text</li></ul>	9 yr 7 mo ago	<span>42</span>
<input type="checkbox"/>	Sonargraph Integration 5.0.2 External Site/Tool Integrations Build Reports Other Post-Build Actions	2 yr 6 mo ago	<span>100</span>
<input type="checkbox"/>	CodeSonar 3.6.0 DevOps Build Notifiers Build Reports Other Post-Build Actions	7 mo 22 days ago	<span>97</span>

Click on “Install”

Jenkins / Manage Jenkins / Plugins

Updates Available plugins (selected) Installed plugins Advanced settings Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

commons-lang3 v3.x Jenkins API	Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
commons-text API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

We have installed SonarQube plugins

## 5.1.5 Install Docker Plugins

Let us install the Docker plugins. Go to the Jenkins Page

Plugin	Status
commons-lang3 v3.x Jenkins API	Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
commons-text API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

Click on “Available Plugins”

Install	Name ↓	Released	Health
<input type="checkbox"/>	PAM Authentication 1.12 Security Adds Unix Pluggable Authentication Module (PAM) support to Jenkins	10 mo ago	91
<input type="checkbox"/>	JavaMail API 1.6.2-11 Library plugins (for use by other plugins) This plugin provides the JavaMail API for other plugins.	10 mo ago	96
<input type="checkbox"/>	Command Agent Launcher 123.v37cfcd92ef67 Agent Management Allows agents to be launched using a specified command.	8 mo 29 days ago	93
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 83.v417146707a_3d Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	11 mo ago	96
<input type="checkbox"/>	SSH server 3.374.v19b_d59ce6610 Adds SSH server functionality to Jenkins, exposing CLI commands through it.	4 mo 27 days ago	97
<input type="checkbox"/>	JSch dependency 0.2.16-95.v3eeeb_55fa_b_78 Library plugins (for use by other plugins) · Miscellaneous Jenkins plugin that brings the JSch library as a plugin dependency, and provides an SSHAuthenticatorFactory for using JSch with the ssh-credentials plugin.	10 mo ago	93
<input type="checkbox"/>	Authentication Tokens API 1.144.v5ff4a_5ec5c33 This plugin provides an API for converting credentials into authentication tokens in Jenkins.	5 mo 2 days ago	100
<input type="checkbox"/>	Javadoc 354.vee1a_660b_4990 This plugin adds Javadoc support to Jenkins.	5 mo 5 days ago	100

## Search for “Docker”

The screenshot shows the Jenkins plugin manager interface. A search bar at the top contains the text "Docker". Below the search bar, there is a table of available plugins. The first three plugins listed are:

- Docker** 1308.vff6e33248305  
Cloud Providers Cluster Management docker  
This plugin integrates Jenkins with Docker
- Docker Commons** 457.v0f62a\_94f11a\_3  
Library plugins (for use by other plugins) docker  
Provides the common shared functionality for various Docker-related plugins.
- Docker Pipeline** 634.vedc7242b\_ed\_a\_7  
pipeline DevOps Deployment docker  
Build and use Docker containers from pipelines.  
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.

Three orange arrows point to the "Available plugins" button in the sidebar, the checkbox for the Docker plugin, and the checkbox for the Docker Pipeline plugin.

Select “Docker”, “Docker Pipeline” and “Docker API”

The screenshot shows the Jenkins plugin manager interface after selecting the "Docker", "Docker Pipeline", and "Docker API" plugins. The sidebar shows the "Available plugins" button is selected. The table now has three checkboxes checked:

- Docker** 1308.vff6e33248305  
Cloud Providers Cluster Management docker  
This plugin integrates Jenkins with Docker
- Docker Pipeline** 634.vedc7242b\_ed\_a\_7  
pipeline DevOps Deployment docker  
Build and use Docker containers from pipelines.  
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
- Docker API** 3.7.0-133.v93b\_8fb\_c17a\_77  
Library plugins (for use by other plugins) docker  
This plugin provides docker-java API for other plugins.

An orange arrow points to the "Install" button at the top right of the table.

Click on “Install”

**Download progress**

Preparation	
• Checking internet connectivity	Success
• Checking update center connectivity	Success
• Success	Success
commons-lang3 v3.x Jenkins API	Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
commons-text API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

## 5.1.6 Install Kubernetes Plugins

Let us install the Kubernetes plugins. Go to the Jenkins Page

**Download progress**

Preparation	
• Checking internet connectivity	Success
• Checking update center connectivity	Success
• Success	Success
commons-lang3 v3.x Jenkins API	Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
commons-text API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

Click on “Available Plugins”

Not secure 35.172.225.123:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

### Plugins

- Updates
- Available plugins**
- Installed plugins
- Advanced settings
- Download progress

Search available plugins

Install	Name ↓	Released	Health
<input type="checkbox"/>	PAM Authentication 1.12 Security Adds Unix Pluggable Authentication Module (PAM) support to Jenkins	10 mo ago	<span style="color: green;">91</span>
<input type="checkbox"/>	JavaMail API 1.6.2-11 Library plugins (for use by other plugins) This plugin provides the JavaMail API for other plugins.	10 mo ago	<span style="color: green;">96</span>
<input type="checkbox"/>	Command Agent Launcher 123.v37cfdc92ef67 Agent Management Allows agents to be launched using a specified command.	8 mo 29 days ago	<span style="color: green;">93</span>
<input type="checkbox"/>	Oracle Java SE Development Kit Installer 83.v417146707a_3d Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	11 mo ago	<span style="color: green;">96</span>
<input type="checkbox"/>	SSH server 3.374.v19b_d59ce6610 Adds SSH server functionality to Jenkins, exposing CLI commands through it.	4 mo 27 days ago	<span style="color: green;">97</span>
<input type="checkbox"/>	Git server 137.v0060b_432302 git Library plugins (for use by other plugins) Allows Jenkins to act as a Git server.	9 mo 29 days ago	<span style="color: green;">93</span>
<input type="checkbox"/>	HTML Publisher 427 Build Reports This plugin publishes HTML reports.	6 mo 5 days ago	<span style="color: green;">97</span>
<input type="checkbox"/>	Lockable Resources 1438.v3c0f8c9e2060		

## Search for “Kubernetes”

Not secure 35.172.225.123:8080/manage/pluginManager/available

Jenkins / Manage Jenkins / Plugins

### Plugins

- Updates
- Available plugins**
- Installed plugins
- Advanced settings
- Download progress

Search available plugins

kubernetes

Install	Name ↓	Released	Health
<input type="checkbox"/>	Kubernetes 4398.vb_b_33d9e7fe23 Cloud Providers Cluster Management kubernetes Agent Management This plugin integrates Jenkins with Kubernetes	1 mo 2 days ago	<span style="color: green;">97</span>
<input type="checkbox"/>	Kubernetes Client API 7.3.1-256.v788a_0b_787114 kubernetes Library plugins (for use by other plugins) Kubernetes Client API plugin for use by other Jenkins plugins.	6 mo 7 days ago	<span style="color: green;">92</span>
<input type="checkbox"/>	Kubernetes Credentials 207.v492f58828b_ed kubernetes credentials Common classes for Kubernetes credentials	3 days 20 hr ago	<span style="color: green;">97</span>
<input type="checkbox"/>	Kubernetes CLI 1.364.vadef8cb8b823 kubernetes Configure kubectl for Kubernetes	9 mo 19 days ago	<span style="color: green;">96</span>
<input type="checkbox"/>	Kubernetes Credentials Provider 1.299.v610fa_e76761a_c kubernetes credentials Provides a read only credentials store backed by Kubernetes.	4 mo 3 days ago	<span style="color: green;">91</span>
<input type="checkbox"/>	Kubernetes :: Pipeline :: DevOps Steps 1.6 pipeline kubernetes Kubernetes :: Pipeline :: DevOps Steps	6 yr 11 mo ago	<span style="color: red;">55</span>
<input type="checkbox"/>	GitLab Credentials - Kubernetes Integration 538.v99cc6503c421 kubernetes gitlab Integrates gitlabToken credential type from the gitlab-branch-source-plugin with the k8s credential	1 mo 3 days ago	<span style="color: green;">100</span>

Select “Kubernetes”, “Kubernetes Client API”, “Kubernetes Credentials” and “Kubernetes CLI”

Jenkins / Manage Jenkins / Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings Download progress

Search: kubernetes

Install	Name	Released	Health
<input checked="" type="checkbox"/>	Kubernetes 4398.vb_b_33d9e7fe23	Cloud Providers Cluster Management kubernetes Agent Management	1 mo 2 days ago 97
<input checked="" type="checkbox"/>	Kubernetes Client API 7.3.1-256.v788a_0b_787114	kubernetes Library plugins (for use by other plugins)	6 mo 7 days ago 92
<input checked="" type="checkbox"/>	Kubernetes Credentials 207.v49f58828b_ed	kubernetes credentials	3 days 20 hr ago 97
<input checked="" type="checkbox"/>	Kubernetes CLI 1.364.vadef8cb8b823	kubernetes	9 mo 19 days ago 96
<input type="checkbox"/>	Kubernetes Credentials Provider 1.299.v610fa_e76761a_	kubernetes credentials	4 mo 3 days ago 91
<input type="checkbox"/>	Kubernetes : Pipeline : DevOps Steps 1.6	pipeline kubernetes	6 yr 11 mo ago 55
<input type="checkbox"/>	GitLab Credentials - Kubernetes Integration 538.v99cc6503c421	kubernetes gitlab	1 mo 3 days ago 100

Click on “Install”

Jenkins / Manage Jenkins / Plugins

Plugins Available plugins Installed plugins Advanced settings Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

commons-lang3 v3.x Jenkins API	Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
commons-text API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

We have installed all the needed plugins.

## 5.2 Configure the tools in Jenkins

In this part, we are going to configure the tools / plugins in Jenkins

### 5.2.1 Configure JDK Tool

We will configure the JDK plugin. Go to Jenkins page

The screenshot shows the Jenkins Manage Jenkins interface under the Plugins section. A dropdown menu is open next to the 'Manage Jenkins' link in the top left. An orange arrow points from the text 'Click on the drop down on "Manage Jenkins"' to this dropdown. The 'Download progress' tab is selected. The page displays a list of Jenkins API plugins with their download status:

Plugin	Status
commons-lang3 v3.x Jenkins API	Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
commons-text API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
Timestamper	Success
Caffeine API	Success
Script Security	Success

Click on the drop down on “**Manage Jenkins**”

Select “Tools”

Not secure 35.172.225.123:8080/manage/configureTools/ Summarize ⚡ ⚡ ⚡ ⚡ Chat

Jenkins / Manage Jenkins / Tools

## Tools

Maven Configuration

Default settings provider

Use default maven settings

Default global settings provider

Use default maven global settings

Pipeline Maven Configuration

DAO class

Pipeline Maven Plugin no storage mode

DAO Diagnostics

Dao Not Ready yet

Database Configuration

JDBC URL ?

JDBC URL. For production workloads, use MySQL (incl. Amazon Aurora for MySQL, MariaDB...) or PostgreSQL.

Save Apply

Scroll down to “**JDK Installations**”

JDK installations

+ Add JDK

Git installations

≡ Git

Name: Default

Path to Git executable: git

Install automatically

+ Add Git

Save Apply

Click on “Add JDK”

Options

+ Add Publisher Options

JDK installations

+ Add JDK

≡ JDK

Name: Required

JAVA\_HOME

Install automatically

+ Add JDK

Save Apply

For the “Name”, enter “jdk17”

Jenkins / Manage Jenkins / Tools

Options

+ Add Publisher Options

JDK installations

+ Add JDK

JDK

Name: jdk17

JAVA\_HOME:

Install automatically ?

+ Add JDK

Save Apply

Then, check the box “Install Automatically”

Jenkins / Manage Jenkins / Tools

Options

+ Add Publisher Options

JDK installations

+ Add JDK

JDK

Name: jdk17

Install automatically ?

+ Add Installer

+ Add JDK

Git installations

Save Apply

Click on “Add Installer”

The screenshot shows the Jenkins 'Tools' configuration page under 'Manage Jenkins'. In the 'JDK installations' section, there is a 'JDK' entry named 'jdk17'. A context menu is open over this entry, with the 'Install from adoptium.net' option highlighted by an orange arrow. Other options in the menu include 'Extract \*.zip/\*.tar.gz', '+ Add Installer', 'Filter', 'Run Batch Command', 'Run Shell Command', and 'Git inst...'. Below the 'JDK' section, there is a 'Git' section with a 'Default' entry. At the bottom of the page are 'Save' and 'Apply' buttons.

Select “Install from adoptium.net”

This screenshot shows the same Jenkins configuration page as the first one, but the 'Install from adoptium.net' option has been selected. A context menu is now open over the 'Version' dropdown, which contains the value 'jdk8u472-b08'. An orange arrow points to this dropdown. The rest of the interface is identical to the first screenshot, showing the 'JDK' and 'Git' sections and the 'Save' and 'Apply' buttons at the bottom.

Click on the drop down to select the version

The screenshot shows the Jenkins configuration interface for managing tools. On the left, under 'JDK installations', there is a list of available Java Development Kits (JDKs). A specific entry, 'OpenJDK 17 - HotSpot jdk17' (version jdk-17.0.17+10), is selected and highlighted with a blue background. Below this list, there is a dropdown menu labeled 'Install from adoptium.net' with the version 'jdk-17.0.17+10' selected. At the bottom of the page, there are two buttons: 'Save' and 'Apply'. An orange arrow points from the text 'Select "jdk-17.0.17+10"' to the 'Apply' button.

Select “**jdk-17.0.17+10**”

The screenshot shows the Jenkins configuration interface for managing tools. On the left, under 'JDK installations', there is a list of available Java Development Kits (JDKs). A specific entry, 'OpenJDK 17 - HotSpot jdk17' (version jdk-17.0.17+10), is selected and highlighted with a blue background. Below this list, there is a dropdown menu labeled 'Install from adoptium.net' with the version 'jdk-17.0.17+10' selected. At the bottom of the page, there are two buttons: 'Save' and 'Apply'. An orange arrow points from the text 'Click on “Apply”, followed by clicking on “Save”' to the 'Apply' button.

Click on “**Apply**”, followed by clicking on “**Save**”

Not secure 35.172.225.123:8080/manage/ Summarize ⚡ ⚡ ⚡ ⚡ Chat

Jenkins / Manage Jenkins

System Configuration

- System Configure global settings and paths.
- Tools Configure tools, their locations and automatic installers.
- Nodes Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Docker Plugin for launching build Agents as Docker containers
- Appearance Configure the look and feel of Jenkins
- Managed files e.g. settings.xml for maven, central managed scripts, custom files, ...
- Plugins Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Clouds Add, remove, and configure cloud instances to provision agents on-demand.

Security

- Security Secure Jenkins; define who is allowed to access/use the system.
- Credentials Configure credentials
- Users Create/delete/modify users that can log in to this Jenkins.
- Credential Providers Configure the credential providers and types

Status Information

- System Information Displays various environmental information to assist
- System Log System log captures output from java.util.logging
- Load Statistics Check your resource utilization and see if you need more

We have completed the configuration of Java JDK tool.

## 5.2.2 Configure SonarQube Scanner Tool

We will configure the SonarQube Scanner plugin. Go to Jenkins page

Not secure 35.172.225.123:8080/manage/ Summarize ⚡ ⚡ ⚡ ⚡ Chat

Jenkins / Manage Jenkins

System Configuration

- System Configure global settings and paths.
- Tools Configure tools, their locations and automatic installers.
- Nodes Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Docker Plugin for launching build Agents as Docker containers
- Appearance Configure the look and feel of Jenkins
- Managed files e.g. settings.xml for maven, central managed scripts, custom files, ...
- Plugins Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Clouds Add, remove, and configure cloud instances to provision agents on-demand.

Security

- Security Secure Jenkins; define who is allowed to access/use the system.
- Credentials Configure credentials
- Users Create/delete/modify users that can log in to this Jenkins.
- Credential Providers Configure the credential providers and types

Status Information

- System Information Displays various environmental information to assist
- System Log System log captures output from java.util.logging
- Load Statistics Check your resource utilization and see if you need more

**Click on “Tools”**

The screenshot shows the Jenkins 'Tools' configuration page. At the top, there are sections for 'Maven Configuration' and 'Pipeline Maven Configuration'. Under 'Maven Configuration', there are fields for 'Default settings provider' (set to 'Use default maven settings') and 'Default global settings provider' (set to 'Use default maven global settings'). Under 'Pipeline Maven Configuration', there is a 'DAO class' dropdown set to 'Pipeline Maven Plugin no storage mode'. Below these, there is a 'DAO Diagnostics' section showing 'Dao Not Ready yet'. In the 'Database Configuration' section, there is a 'JDBC URL' field with a note: 'JDBC URL For production workloads, use MySQL (incl. Amazon Aurora for MySQL, MariaDB...) or PostgreSQL.' At the bottom are 'Save' and 'Apply' buttons.

**Scroll down to “SonarQube Scanner installations”**

The screenshot shows the Jenkins 'Tools' configuration page with the 'SonarQube Scanner installations' section highlighted. An orange arrow points to the '+ Add SonarQube Scanner' button. Other sections visible include 'Gradle installations', 'SonarScanner for MSBuild installations', 'Ant installations', and 'Maven installations'. At the bottom are 'Save' and 'Apply' buttons.

**Click on “Add SonarQube Scanner”**

SonarQube Scanner installations

+ Add SonarQube Scanner

**SonarQube Scanner**

Name  Required

Install automatically ?

**Install from Maven Central**

Version

+ Add Installer

+ Add SonarQube Scanner

Save Apply

Let us give it a name, we will call it “sonar-scanner”

SonarQube Scanner installations

+ Add SonarQube Scanner

**SonarQube Scanner**

Name

Install automatically ?

**Install from Maven Central**

Version

+ Add Installer

+ Add SonarQube Scanner

Ant installations

Save Apply

Click on “Apply” followed by “Save”

The screenshot shows the Jenkins Manage Jenkins interface. At the top, there are several warning messages about installed components. Below this, the 'System Configuration' section is visible, featuring links for System, Tools, Plugins, Nodes, Docker, Appearance, Managed files, Security, Credentials, and Credential Providers. A red arrow points to the 'Tools' link.

We have configured “SonarQube Scanner”.

### 5.2.3 Configure Maven Tool

We will configure the Maven plugin. Go to Jenkins page

The screenshot shows the Jenkins Manage Jenkins interface. At the top, there are several warning messages about installed components. Below this, the 'System Configuration' section is visible, featuring links for System, Tools, Plugins, Nodes, Docker, Appearance, Managed files, Security, Credentials, and Credential Providers. A red arrow points to the 'Tools' link.

**Click on “Tools”**

The screenshot shows the Jenkins 'Tools' configuration page. At the top, there are sections for 'Maven Configuration' and 'Pipeline Maven Configuration'. Under 'Maven Configuration', there are dropdowns for 'Default settings provider' (set to 'Use default maven settings') and 'Default global settings provider' (set to 'Use default maven global settings'). Under 'Pipeline Maven Configuration', there is a dropdown for 'DAO class' set to 'Pipeline Maven Plugin no storage mode'. Below these, there is a 'DAO Diagnostics' section showing 'Dao Not Ready yet'. A 'Database Configuration' section follows, with a 'JDBC URL' dropdown set to 'MySQL (incl. Amazon Aurora for MySQL, MariaDB...) or PostgreSQL'. At the bottom are 'Save' and 'Apply' buttons.

**Scroll down to “Maven Installations”**

The screenshot shows the Jenkins 'Tools' configuration page with the 'Maven installations' section highlighted. An orange arrow points to the '+ Add Maven' button. Other sections visible include 'SonarQube Scanner installations', 'Ant installations', and 'Docker installations'. At the bottom are 'Save' and 'Apply' buttons.

**Click on “Add Maven”**

Maven installations

+ Add Maven

**Maven**

Name

! Required

Install automatically ?

**Install from Apache**

Version

+ Add Installer

+ Add Maven

Save Apply

Let us give it a name. I will call it “**maven3**”

Maven installations

+ Add Maven

**Maven**

Name

Install automatically ?

**Install from Apache**

Version

+ Add Installer

+ Add Maven

Docker installations

Save Apply

Click on the drop down and select the version. Let us use “**3.6.1**”

The screenshot shows the Jenkins 'Tools' configuration page. Under 'Maven installations', there is a section for 'maven3' with 'Install automatically' checked. Below it, under 'Install from Apache', the 'Version' is set to '3.6.1'. There are buttons for '+ Add Maven' and '+ Add Installer'. At the bottom, there are 'Save' and 'Apply' buttons.

Then, click on “apply” followed by “Save”

The screenshot shows the Jenkins 'Manage Jenkins' page. A red box highlights a warning message: 'Warnings have been published for the following currently installed components: docker-build-step 2.12: CSRF vulnerability and missing permission check (no fix available). No fixes for these issues are available. It is recommended that you review the security advisory and apply mitigations if possible, or uninstall this plugin.' Below this, there are sections for 'System Configuration' and 'Security'. The 'Tools' section is highlighted, showing 'Configure tools, their locations and automatic installers.' Other sections include 'Nodes', 'Appearance', 'Managed files', 'Plugins', 'Clouds', 'Credentials', and 'Credential Providers'.

We have configured Maven tool.

## 5.2.4 Configure Docker Tool

We will configure the Docker plugin. Go to Jenkins page

The screenshot shows the Jenkins 'Manage Jenkins' page. At the top, there are two buttons: 'Go to plugin manager' and 'Configure which of these warnings are shown'. A red arrow points from the text 'Click on "Tools"' to the 'Tools' button. Below the buttons, there is a warning message about a CSRF vulnerability and missing permission check. The 'System Configuration' section contains links for System, Nodes, Appearance, Docker, and Managed files. The 'Security' section contains links for Security, Credentials, and Credential Providers. The URL in the address bar is 35.172.225.123:8080/manage/docker-plugin.

Click on “Tools”

The screenshot shows the Jenkins 'Tools' configuration page. It includes sections for Maven Configuration, Pipeline Maven Configuration, DAO Diagnostics, and Database Configuration. Under Maven Configuration, there are dropdowns for 'Default settings provider' (set to 'Use default maven settings') and 'Default global settings provider' (set to 'Use default maven global settings'). Under Pipeline Maven Configuration, there is a dropdown for 'DAO class' (set to 'Pipeline Maven Plugin no storage mode'). Under DAO Diagnostics, it says 'Dao Not Ready yet'. Under Database Configuration, there is a 'JDBC URL' field with a note: 'JDBC URL. For production workloads, use MySQL (incl. Amazon Aurora for MySQL, MariaDB...) or PostgreSQL.' At the bottom, there are 'Save' and 'Apply' buttons.

Scroll down to “Docker Installations”

The screenshot shows the Jenkins 'Tools' configuration page. It includes sections for SonarQube Scanner, Ant, Maven, and Docker installations. A red arrow points to the '+ Add Docker' button under the Docker installations section. At the bottom are 'Save' and 'Apply' buttons.

Click on “Add Docker”

The screenshot shows the 'Docker' configuration dialog box. It has fields for 'Name' (with a required indicator), 'Installation root', and an 'Install automatically' checkbox. The 'Name' field is highlighted with a red border. At the bottom are 'Save' and 'Apply' buttons.

Let us give it a name, we will call it “**docker**”

The screenshot shows the Jenkins 'Tools' configuration page. Under 'Docker installations', there is a single entry named 'docker'. The 'Install automatically' checkbox is unchecked. A red arrow points to this checkbox. Below the entry are 'Save' and 'Apply' buttons.

Check the box “Install Automatically”

The screenshot shows the same Jenkins 'Tools' configuration page as before, but now the 'Install automatically' checkbox for the 'docker' entry is checked. A red arrow points to this checked checkbox. Below the entry is a '+ Add Installer' button, which is also highlighted with a red arrow. The page includes 'Save' and 'Apply' buttons at the bottom.

Click on “Add Installer”

The screenshot shows the Jenkins 'Tools' configuration page. Under 'Docker installations', a 'Docker' entry is selected with the name 'docker'. A context menu is open over the 'Install automatically?' checkbox, with 'Download from docker.com' highlighted.

Maven installations

Docker installations

+ Add Docker

Docker

Name: docker

Install automatically?

+ Add Installer

Filter

- + Download from docker.com
- + Extract \*.zip/\*.tar.gz
- + Run Batch Command
- + Run Shell Command

Save Apply

Jenkins 2.528.3

Select “Download from docker.com”

The screenshot shows the Jenkins 'Tools' configuration page. Under 'Docker installations', a 'Docker' entry is selected with the name 'docker'. The 'Install automatically?' checkbox is checked. The 'Download from docker.com' section is expanded, showing 'Docker version: latest'.

Maven installations

Docker installations

+ Add Docker

Docker

Name: docker

Install automatically?

Download from docker.com

Docker version: latest

+ Add Installer

+ Add Docker

Save Apply

Click on “Apply”, followed by “Save”

The screenshot shows the Jenkins Manage Jenkins interface. At the top, there is a warning message about installed components:

Warnings have been published for the following currently installed components:

- [docker-build-step 2.12](#): CSRF vulnerability and missing permission check (no fix available)

No fixes for these issues are available. It is recommended that you review the security advisory and apply mitigations if possible, or uninstall this plugin.

[Go to plugin manager](#) [Configure which of these warnings are shown](#)

**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.
- [\*\*Docker\*\*](#) Plugin for launching build Agents as Docker containers
- [\*\*Clouds\*\*](#) Add, remove, and configure cloud instances to provision agents on-demand.
- [\*\*Appearance\*\*](#) Configure the look and feel of Jenkins
- [\*\*Managed files\*\*](#) e.g. settings.xml for maven, central managed scripts, custom files, ...

**Security**

- [\*\*Security\*\*](#) Secure Jenkins; define who is allowed to access/use the system.
- [\*\*Credentials\*\*](#) Configure credentials
- [\*\*Credential Providers\*\*](#) Configure the credential providers and types

We have configured docker.

## 5.3 Create Credentials

In this part, we are going to create credentials for Java JDK,

### 5.3.1 Create Credentials for Java JDK

We are going to create the credentials for Java JDK. Go to Jenkins page and navigate to “**Manage Jenkins**”

Warnings have been published for the following currently installed components:

- docker-build-step 2.12: CSRF vulnerability and missing permission check (no fix available)

No fixes for these issues are available. It is recommended that you review the security advisory and apply mitigations if possible, or uninstall this plugin.

**System Configuration**

- System**: Configure global settings and paths.
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Appearance**: Configure the look and feel of Jenkins.
- Tools**: Configure tools, their locations and automatic installers.
- Docker**: Plugin for launching build Agents as Docker containers.
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Managed files**: e.g. settings.xml for maven, central managed scripts, custom files, ...

**Security**

- Security**: Secure Jenkins; define who is allowed to access/use the system.
- Credentials**: Configure credentials
- Credential Providers**: Configure the credential providers and types

Click on “**Credentials**”

Stores scoped to Jenkins

System Domains: Global

Click on “**Global**”

This screenshot shows the Jenkins Global credentials (unrestricted) page. The URL is 35.172.225.123:8080/manage/credentials/store/system/domain/\_/. The page displays a message: "This credentials domain is empty" and "How about adding some credentials?". An orange arrow points to the text "adding some credentials?".

Click on “Adding some credentials”

This screenshot shows the Jenkins New credentials page for a "Username with password" credential. The URL is 35.172.225.123:8080/manage/credentials/store/system/domain/\_/newCredentials. The "Scope" is set to "Global (Jenkins, nodes, items, all child items, etc)". The "Username" field is highlighted with an orange arrow. Below it, there is a note: "Blank username; did you mean to use secret text credentials instead?" and a checkbox "Treat username as secret". The "Password", "ID", and "Description" fields are also present. A "Create" button is at the bottom.

For “username”, I am going to provide my GitHub username that is “ebotsidneysmith”

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...)

## New credentials

Kind: Username with password

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

Username: ebotsidneysmith

Treat username as secret

Password: (Field highlighted by an orange arrow)

ID: (Field highlighted by an orange arrow)

Description:

Create

Then for the password, we will enter the token we created before on GitHub. That is

ghp\_jG2b5rXHd7U6Kd1scwUeRKPpaw9Wlf1zjGGf

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...)

## New credentials

Kind: Username with password

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

Username: ebotsidneysmith

Treat username as secret

Password: (Field containing '.....')

ID: (Field highlighted by an orange arrow)

Description:

Create

REST API Jenkins 2.528.3

For "ID", we will enter "**git-credential**"

Not secure 35.172.225.123:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... Chat

Username with password

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

Username  
ebotsidneysmith

Treat username as secret ?

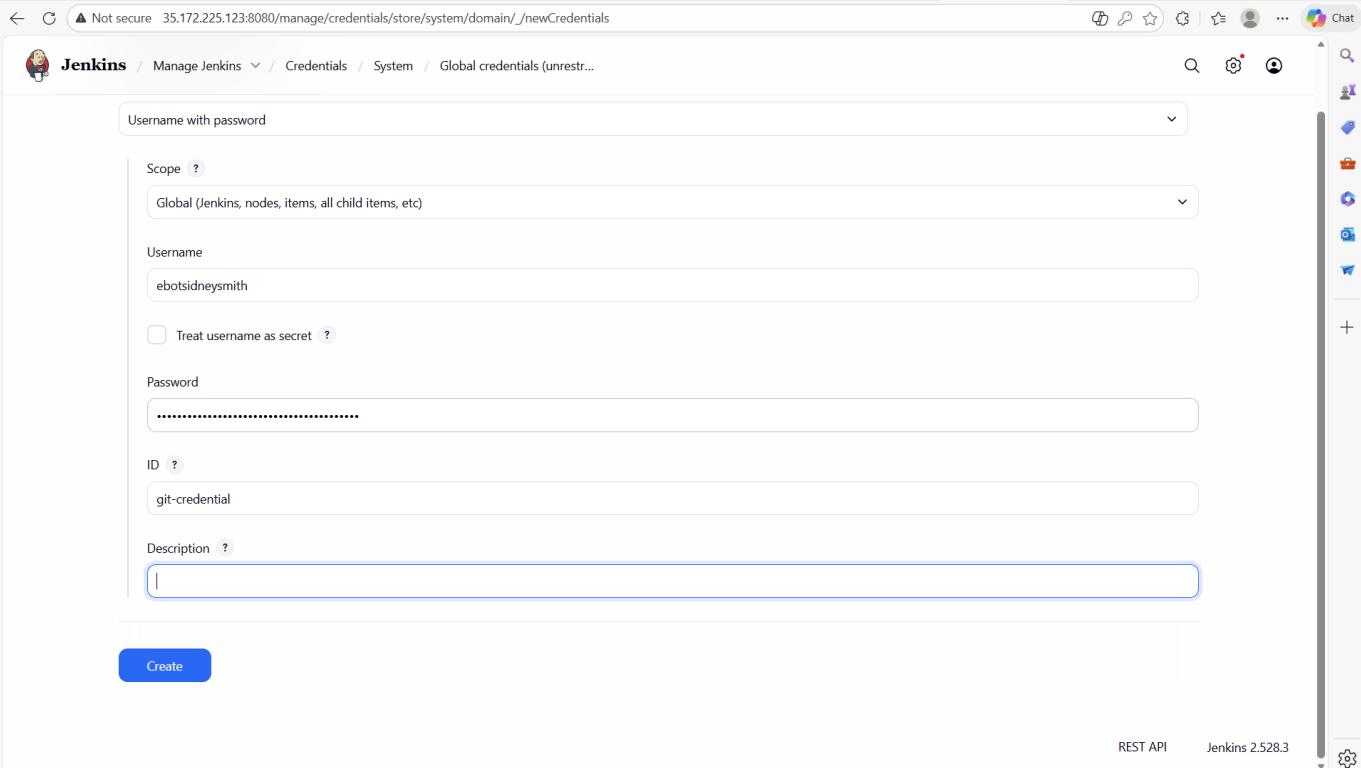
Password  
.....

ID ?  
git-credential

Description ?  
|

Create

REST API Jenkins 2.528.3



For “Description”, we will enter “git-credential”

Not secure 35.172.225.123:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... Chat

Username with password

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

Username  
ebotsidneysmith

Treat username as secret ?

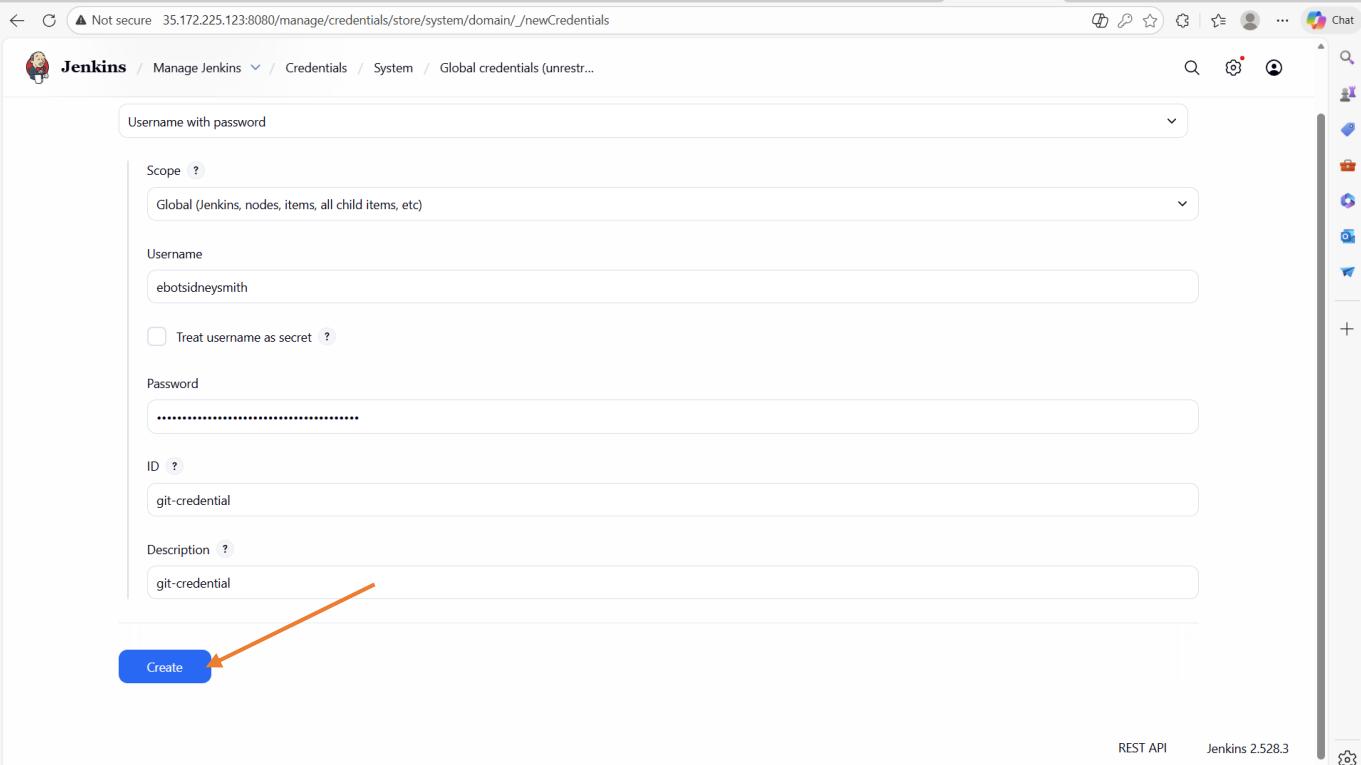
Password  
.....

ID ?  
git-credential

Description ?  
git-credential

**Create**

REST API Jenkins 2.528.3



Click on “Create”

The screenshot shows the Jenkins Global credentials (unrestricted) page. At the top, there is a navigation bar with links for Manage Jenkins, Credentials, System, and Global credentials (unrestricted). A search icon and a gear icon are also present. On the right side, there is a sidebar with various icons for Jenkins features like Pipeline, Build, Configuration, and Dashboards. The main content area is titled "Global credentials (unrestricted)" and contains a message: "Credentials that should be available irrespective of domain specification to requirements matching." Below this, there is a single credential entry: "ebotsidneysmith/\*\*\*\*\*\*\*\* (git-credential)" with the subtitle "git-credential - git-credential". To the right of the entry are a trash can icon and an ellipsis menu icon. At the bottom right of the page, there are links for REST API and Jenkins 2.528.3, along with a gear icon.

We have created the credential for GitHub

### 5.3.2 Create Credentials for SonarQube

We are going to create the credentials for SonarQube. Go to Jenkins page and navigate to “**Manage Jenkins**”

The screenshot shows the Jenkins Manage Jenkins interface. In the top right corner, there are several icons: Summarize, Chat, and a search bar labeled "Search settings". Below the header, a red box highlights a warning message: "Warnings have been published for the following currently installed components: docker-build-step 2.12: CSRF vulnerability and missing permission check (no fix available). No fixes for these issues are available. It is recommended that you review the security advisory and apply mitigations if possible, or uninstall this plugin." To the right of the warning are two buttons: "Go to plugin manager" and "Configure which of these warnings are shown".  
  
The main content area is titled "System Configuration" and contains several sections:

- 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.
- Docker**: Plugin for launching build Agents as Docker containers.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Appearance**: Configure the look and feel of Jenkins.
- Managed files**: e.g. settings.xml for maven, central managed scripts, custom files, ... (this item has a blue arrow pointing to it from the warning message above).
- Security**: Secure Jenkins; define who is allowed to access/use the system.
- Credentials**: Configure credentials (this item has a blue arrow pointing to it from the warning message above).
- Credential Providers**: Configure the credential providers and types.

At the bottom left of the main content area, the URL "35.172.225.123:8080/manage/configfiles" is visible.

**Click on “Credentials”**

The screenshot shows the Jenkins Credentials page. At the top, there is a breadcrumb navigation: Jenkins / Manage Jenkins / Credentials. The main title is "Credentials".  
  
A single credential entry is listed:

- ebotsidneysmith/\*\*\*\*\* (git-credential)**
- System - Global - git-credential - git-credential

Below the list, a note says "Stores scoped to Jenkins".  
  
At the bottom, there is a navigation bar with "System" and "Domains: Global". A blue arrow points from the "Global" text in the navigation bar towards the "Global" text in the note above.  
  
The URL at the top is "35.172.225.123:8080/manage/credentials/credentials/store/system/domain/\_/".

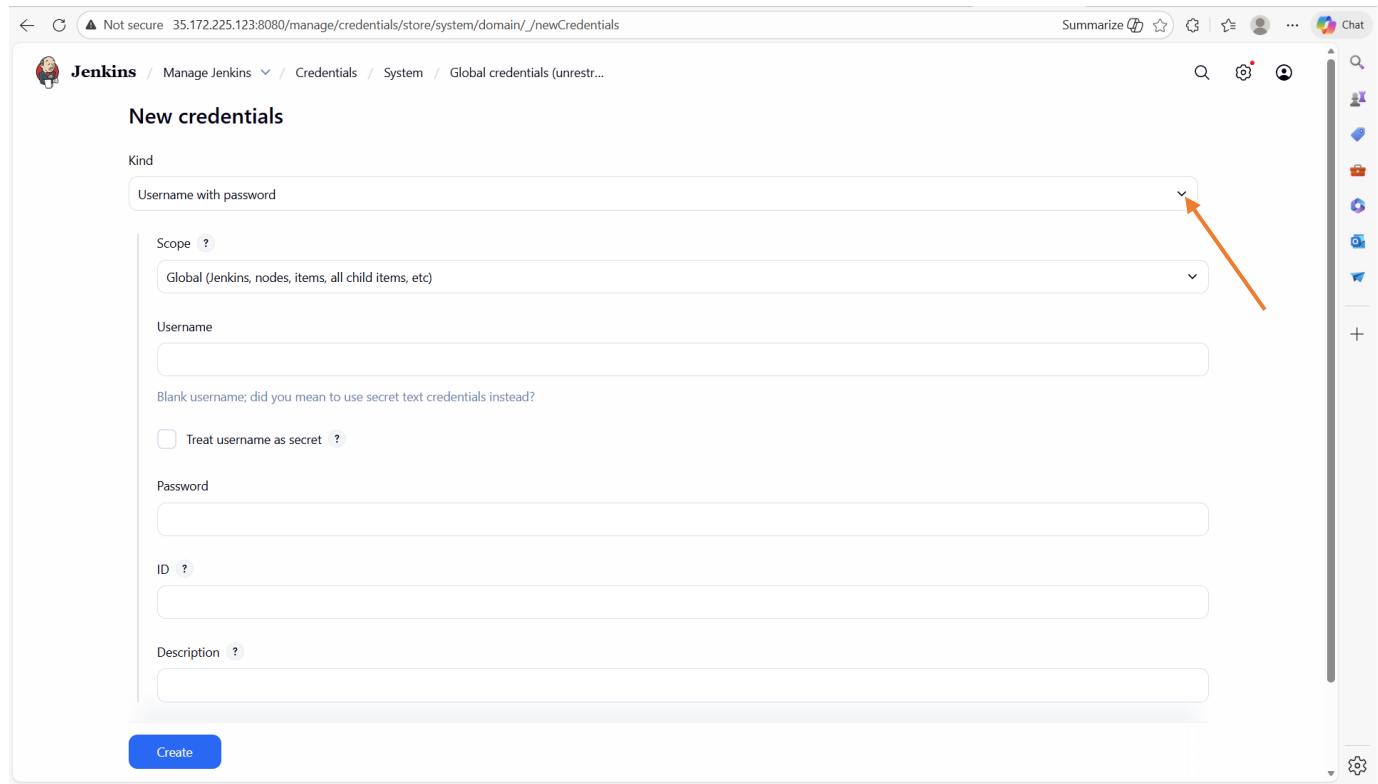
**Click on “Global”**

The screenshot shows the Jenkins Global credentials page. The URL at the top is "35.172.225.123:8080/manage/credentials/credentials/store/system/domain/\_/".  
  
The main title is "Global credentials (unrestricted)".  
  
A single credential entry is listed:

- ebotsidneysmith/\*\*\*\*\* (git-credential)**
- git-credential - git-credential

To the right of the credential list, there is a blue button labeled "+ Add Credentials" with a blue arrow pointing to it.

**Click on “Add Credentials”**



New credentials

Kind

Username with password

Scope ?

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

Username

Blank username; did you mean to use secret text credentials instead?

Treat username as secret ?

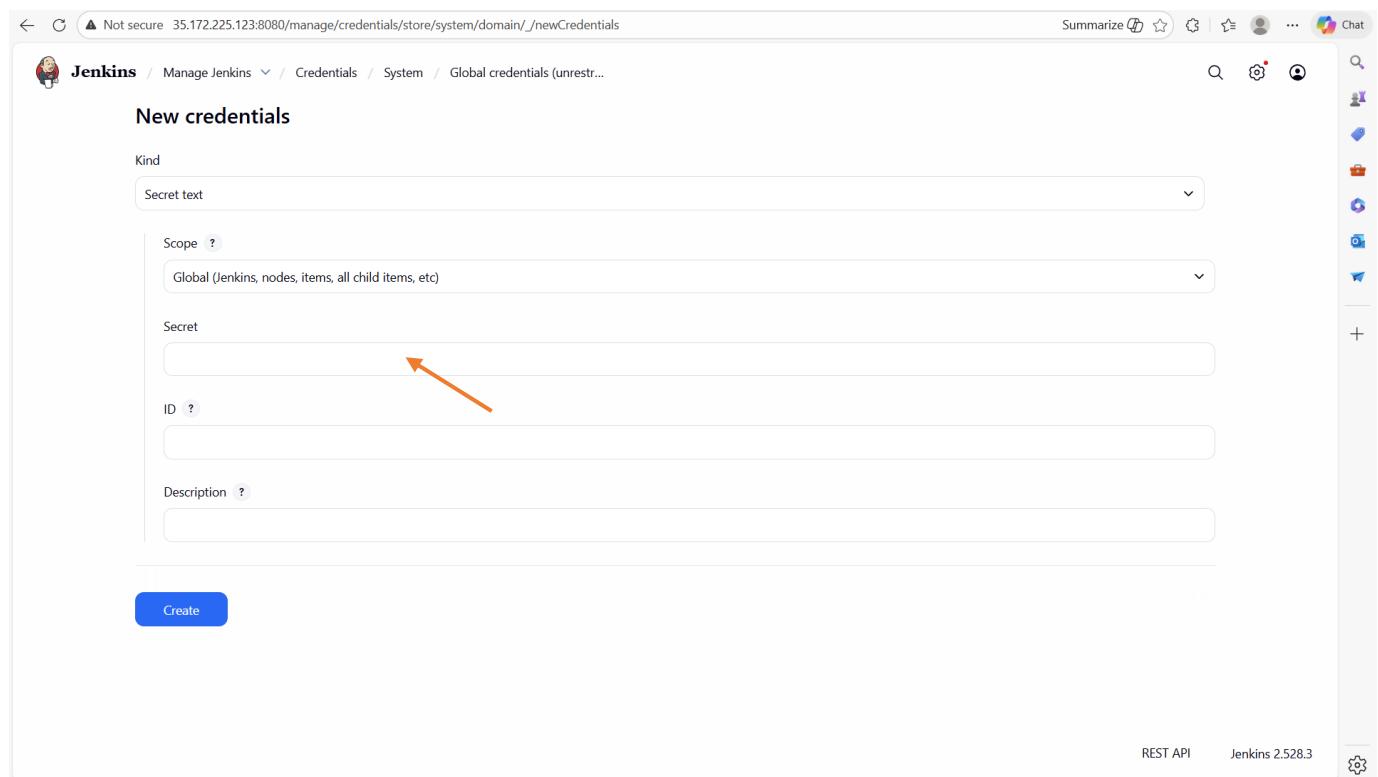
Password

ID ?

Description ?

Create

Click on the drop down on “Kind” and select “Secret Text”



New credentials

Kind

Secret text

Scope ?

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

Secret

ID ?

Description ?

Create

Copy the Token we generated on SonarQube and paste on “Secret”

Not secure 35.172.225.123:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... ... Chat

### New credentials

Kind Secret text

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

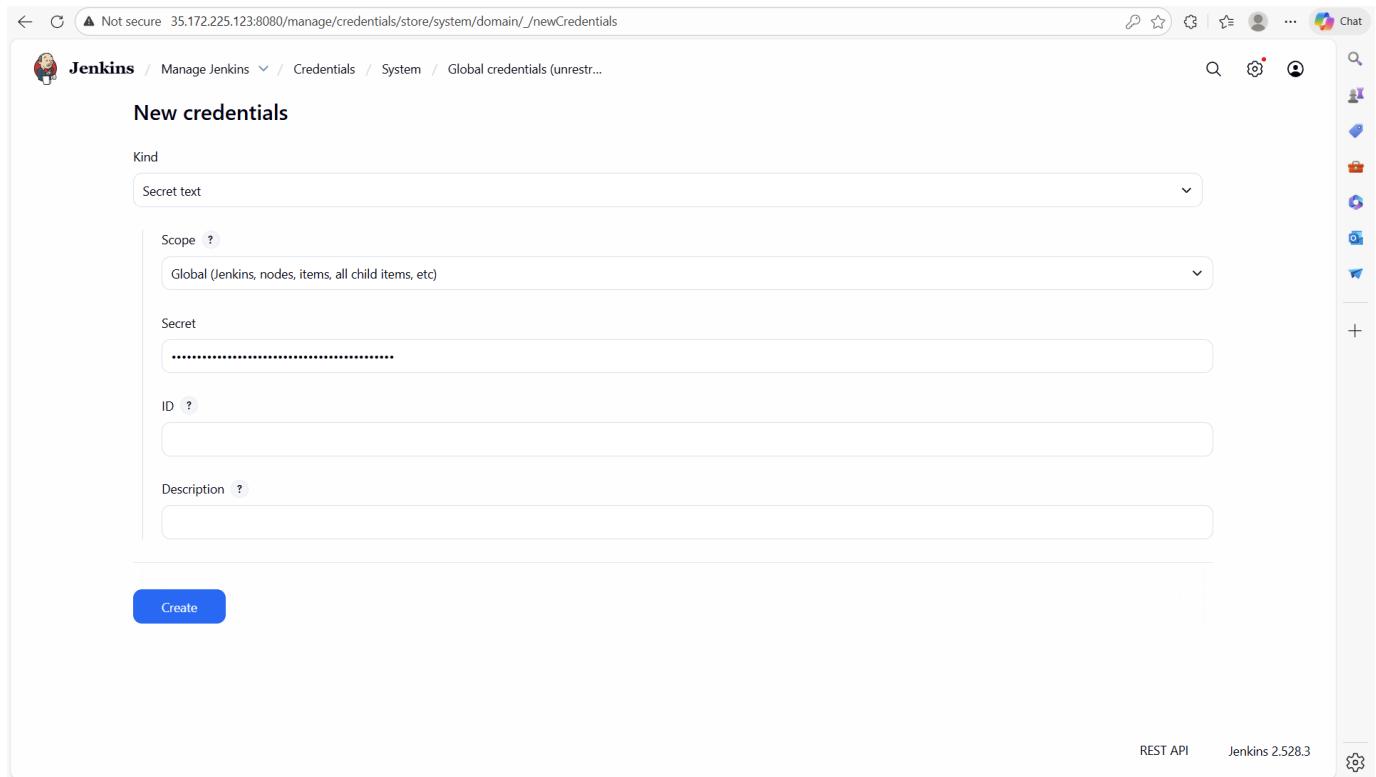
Secret .....

ID ?

Description ?

Create

REST API Jenkins 2.528.3



Then for "ID", we will put "**sonar-token**"

Not secure 35.172.225.123:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... ... Chat

### New credentials

Kind Secret text

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

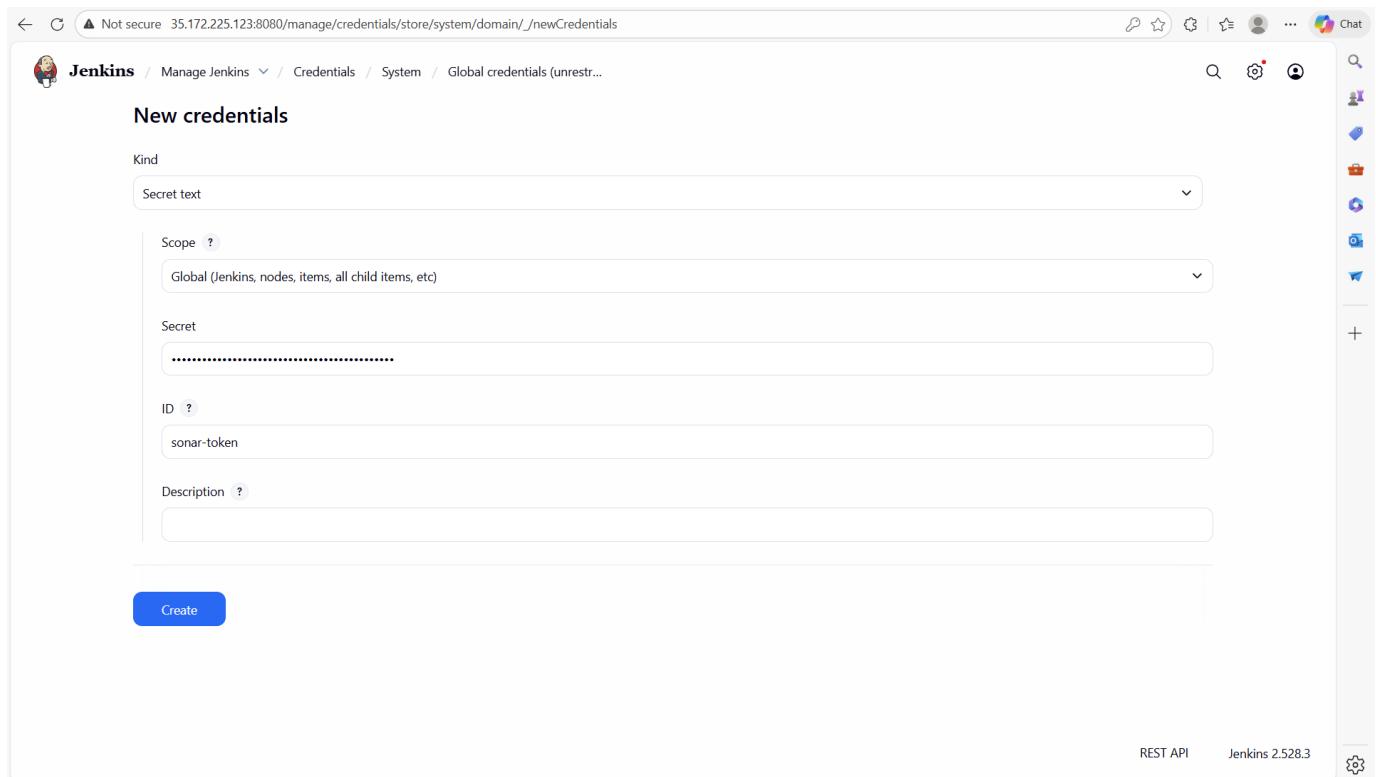
Secret .....

ID ? sonar-token

Description ?

Create

REST API Jenkins 2.528.3



We will also give the "**Description**" as "**sonar-token**"

New credentials

Kind: Secret text

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

Secret: [REDACTED]

ID: sonar-token

Description: sonar-token

Create

Click on “Create”

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

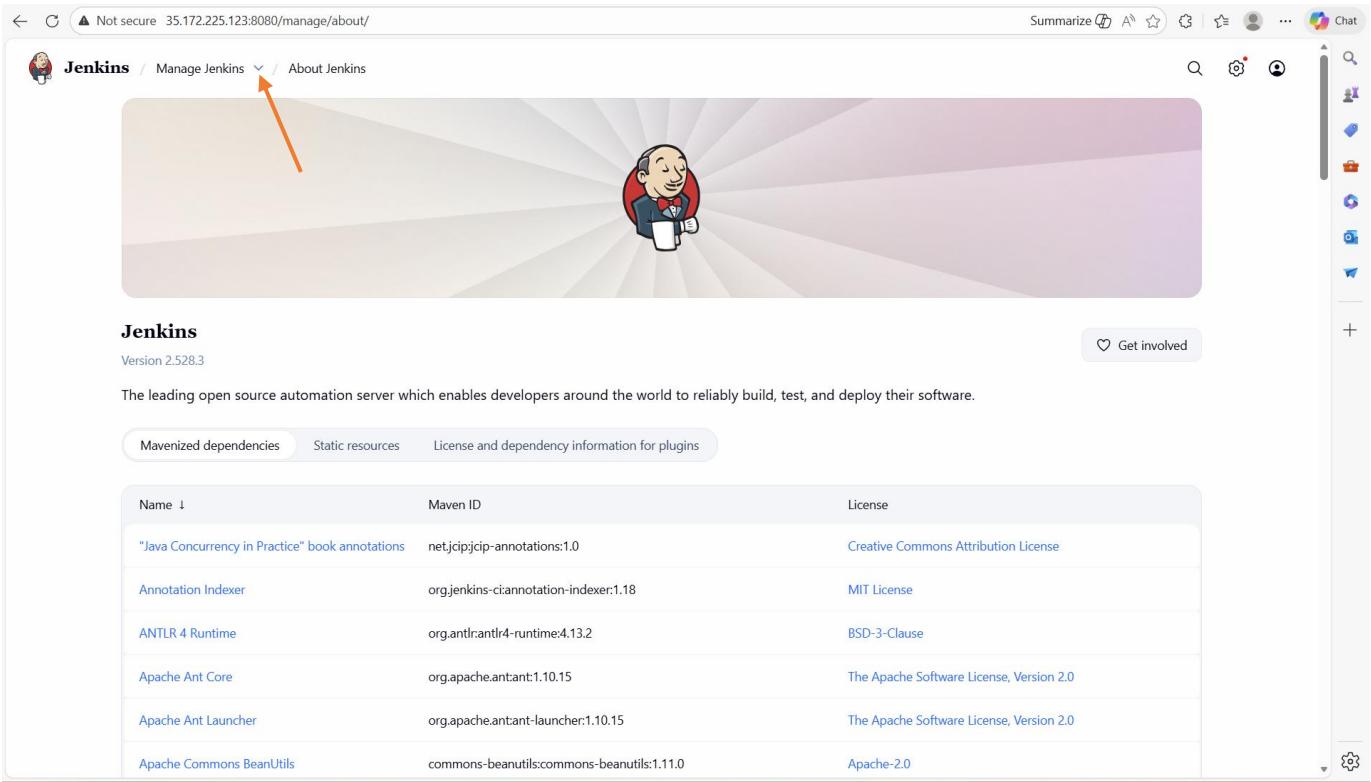
ebotsidneysmith/******/ (git-credential)	... git-credential - git-credential
sonar-token	... sonar-token - sonar-token

+ Add Credentials

We have created the SonarQube credential.

### 5.2.3 Configure SonarQube Server

We will configure the SonarQube server. Go to Jenkins page and navigate to “Manage Jenkins”



Jenkins / Manage Jenkins / About Jenkins

**Jenkins**

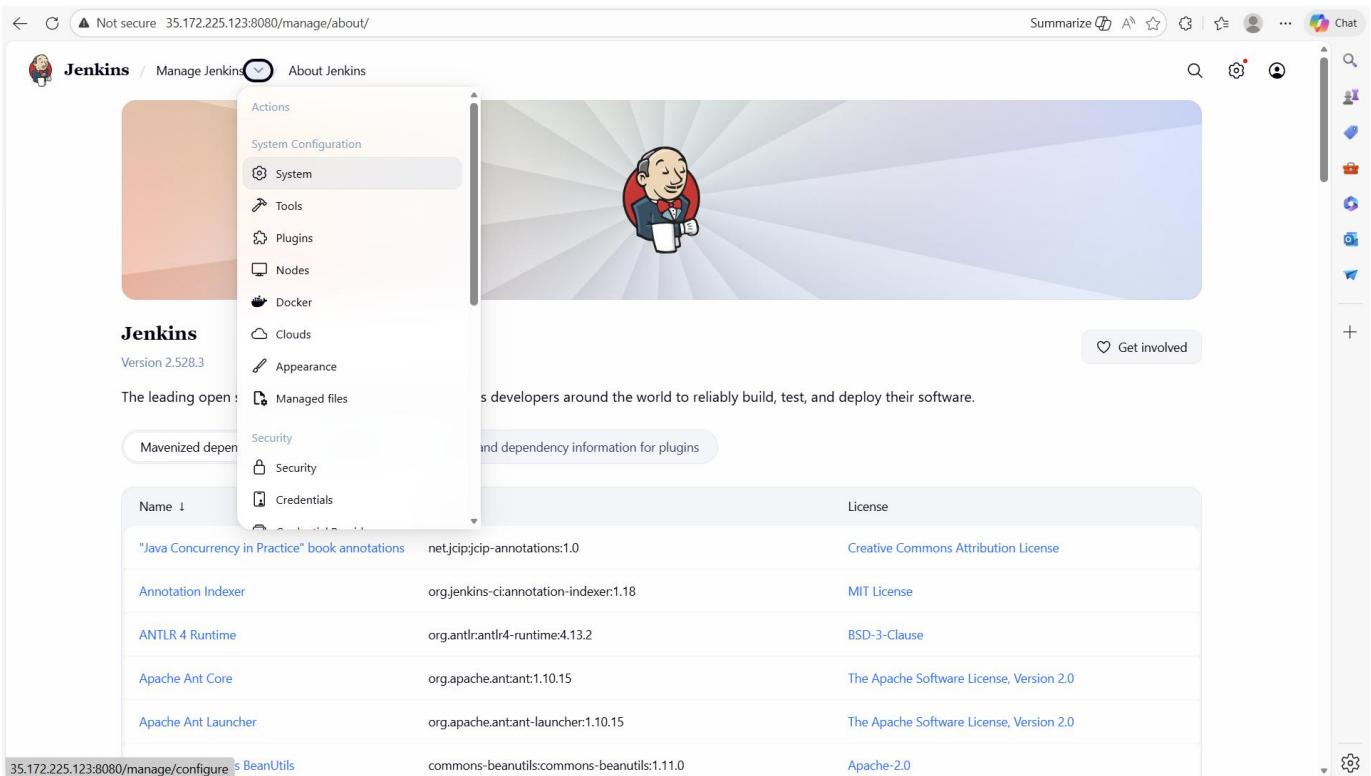
Version 2.528.3

The leading open source automation server which enables developers around the world to reliably build, test, and deploy their software.

Mavenized dependencies Static resources License and dependency information for plugins

Name ↓	Maven ID	License
"Java Concurrency in Practice" book annotations	net.jcip:jcip-annotations:1.0	Creative Commons Attribution License
Annotation Indexer	org.jenkins-ci:annotation-indexer:1.18	MIT License
ANTLR 4 Runtime	org.antlr:antlr4-runtime:4.13.2	BSD-3-Clause
Apache Ant Core	org.apache.ant:ant:1.10.15	The Apache Software License, Version 2.0
Apache Ant Launcher	org.apache.ant:ant-launcher:1.10.15	The Apache Software License, Version 2.0
Apache Commons BeanUtils	commons-beanutils:commons-beanutils:1.11.0	Apache-2.0

Click on the drop down on “Manage Jenkins”



Jenkins / Manage Jenkins / About Jenkins

**Jenkins**

Version 2.528.3

The leading open : s developers around the world to reliably build, test, and deploy their software.

Mavenized dependencies Static resources License and dependency information for plugins

Actions

- System Configuration
- System**
- Tools
- Plugins
- Nodes
- Docker
- Clouds
- Appearance
- Managed files
- Security
- Credentials

Name ↓	Maven ID	License
"Java Concurrency in Practice" book annotations	net.jcip:jcip-annotations:1.0	Creative Commons Attribution License
Annotation Indexer	org.jenkins-ci:annotation-indexer:1.18	MIT License
ANTLR 4 Runtime	org.antlr:antlr4-runtime:4.13.2	BSD-3-Clause
Apache Ant Core	org.apache.ant:ant:1.10.15	The Apache Software License, Version 2.0
Apache Ant Launcher	org.apache.ant:ant-launcher:1.10.15	The Apache Software License, Version 2.0
Apache Commons BeanUtils	commons-beanutils:commons-beanutils:1.11.0	Apache-2.0

Select “System”

The screenshot shows the Jenkins System configuration page. It includes fields for the Home directory (set to /var/lib/jenkins), a System Message area (empty), and sections for Maven Project Configuration, Global MAVEN\_OPTS, Local Maven Repository, and the number of executors (set to 1). At the bottom are Save and Apply buttons.

## Scroll down to “SonarQube Servers”

The screenshot shows the Jenkins System configuration page with the "SonarQube servers" section expanded. It contains fields for Environment variables, SonarQube installations (with a "+ Add SonarQube" button highlighted by an orange arrow), Metrics, Access keys, Pipeline Speed / Durability, and Default Speed / Durability Level. At the bottom are Save and Apply buttons.

Click on “Add SonarQube”

SonarQube servers

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

Environment variables

SonarQube installations

List of SonarQube installations

Name

**This property is mandatory.**

Server URL  
Default is http://localhost:9000

Server authentication token  
SonarQube authentication token. Mandatory when anonymous access is disabled.  
- none -

Advanced

For the “**Name**”, let us call it “**sonar**”

SonarQube servers

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

Environment variables

SonarQube installations

List of SonarQube installations

Name

Server URL  
Default is http://localhost:9000

Server authentication token  
SonarQube authentication token. Mandatory when anonymous access is disabled.  
- none -

Advanced

For the “**Server URL**”, go back to the SonarQube browser

The screenshot shows the SonarQube Administration interface. The URL in the address bar is `http://3.80.70.234:9000/admin/users`. The main content area is titled "Tokens of Administrator". It includes a "Generate Tokens" form and a table listing existing tokens. A message at the top of the table says "New token 'sonar-token' has been created. Make sure you copy it now, you won't be able to see it again!" with a "Copy" button. The table has columns: Name, Type, Project, Last use, Created, and Expiration. One row is shown: "sonar-token", "User", "", "Never", "January 4, 2026", "February 2, 2026". A "Revoke" button is next to the expiration date. At the bottom right of the table is a "Done" button. A note at the bottom left says "Embedded database should be used for evaluation purposes only". The footer contains the SonarQube logo and links to Community Edition v9.8, NO LONGER ACTIVE, LGPL v3, Documentation, Plugins, and Web API.

Copy the highlighted part:

`http://3.80.70.234:9000/`

And paste on the Jenkins browser

The screenshot shows the Jenkins System configuration page under "Manage Jenkins / System". The URL in the address bar is `http://35.172.225.123:8080/manage/configure`. The "SonarQube servers" section is visible. Under "SonarQube installations", there is a table with one row. The "Name" column has a text input field containing "sonar". The "Server URL" column has a text input field containing "Default is http://localhost:9000" followed by "http://3.80.70.234:9000/". The "Server authentication token" section shows a dropdown menu with "- none -" and a "+ Add" button. At the bottom are "Save" and "Apply" buttons.

Remove the slash “/”

The screenshot shows the Jenkins System configuration page under the 'SonarQube servers' section. It includes fields for 'Name' (set to 'sonar'), 'Server URL' (set to 'http://3.80.70.234:9000'), and 'Server authentication token' (a dropdown menu currently showing '- none -'). A red arrow points to the '+ Add' button next to the token dropdown. At the bottom are 'Save' and 'Apply' buttons.

Click on the drop down on “Server Authentication Token” and select our token

The screenshot shows the same Jenkins System configuration page after selecting a token. The 'Server authentication token' dropdown now contains 'sonar-token'. The rest of the page remains the same, with 'Save' and 'Apply' buttons at the bottom.

Click on “Apply”, then click on “Save”

The screenshot shows the Jenkins home page. At the top, there's a header with a back arrow, forward arrow, and a security warning icon. The URL is 13.217.42.186:8080. On the left sidebar, there are links for 'New Item' and 'Build History'. Below the sidebar, there are two sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). The main content area has a heading 'Welcome to Jenkins!' followed by a sub-section 'Start building your software project' with a 'Create a job' button and a '+' sign. Another section titled 'Set up a distributed build' includes links for 'Set up an agent' (with a monitor icon), 'Configure a cloud' (with a cloud icon), and 'Learn more about distributed builds' (with a question mark icon). At the bottom right, there are links for 'REST API' and 'Jenkins 2.528.3'. A vertical sidebar on the right contains icons for various Jenkins features like users, groups, and clouds.

### 5.3.4 Create Credentials for Nexus

We are going to create the credentials for Nexus. Go to Jenkins page and navigate to “Manage Jenkins”

The screenshot shows the 'Manage Jenkins' page. At the top, there's a header with a back arrow, forward arrow, and a security warning icon. The URL is 35.172.225.123:8080/manage/. The main content area has a heading 'Manage Jenkins' and a search bar. A red box highlights a warning message: 'Warnings have been published for the following currently installed components: docker-build-step 2.12: CSRF vulnerability and missing permission check (no fix available). No fixes for these issues are available. It is recommended that you review the security advisory and apply mitigations if possible, or uninstall this plugin.' Below this, there are several configuration sections: 'System Configuration' (System, Nodes, Appearance, Security), 'Tools' (Tools, Docker), 'Plugins' (Plugins, Clouds), 'Managed files' (with a red arrow pointing to it), 'Credentials' (Credentials), and 'Credential Providers'.

Click on “Managed Files”

Jenkins / Manage Jenkins / Managed files

## Config File Management

You can edit or remove your configuration files

+ Add a new Config

Click on “Add a new config”

New configuration

Type

- Global Maven settings.xml**  
A global maven settings.xml which can be referenced within Apache Maven jobs.  
Use it within maven projects or maven builder and reference credentials for a server authentication from here: [credentials](#)
- Maven settings.xml**  
A settings.xml which can be referenced within Apache Maven jobs.  
Use it within maven projects or maven builder and reference credentials for a server authentication from here: [credentials](#)
- Properties file**  
a Properties file [credentials](#)
- Json file**  
a Json file
- Maven toolchains.xml**  
a toolchains.xml which can be referenced within Apache Maven jobs
- Simple XML file**  
a general xml file
- Groovy file**  
a reusable groovy script
- Custom file**  
a custom file (e.g. text or any other not yet available format)
- Extended Email Publisher Groovy Template**  
A Groovy template used by the Extended Email Publisher plugin to generate emails.
- Extended Email Publisher Jelly Template**  
A Jelly template used by the Extended Email Publisher plugin to generate emails.

ID

Select “**Global Maven settings.xml**” since we are using Java.

Jenkins / Manage Jenkins / Managed files

+ Add a new Config

Global Maven settings.xml  
A global maven settings.xml which can be referenced within Apache Maven jobs.  
Use it within maven projects or maven builder and reference credentials for a server authentication from here: [credentials](#)

Maven settings.xml  
A settings.xml which can be referenced within Apache Maven jobs.  
Use it within maven projects or maven builder and reference credentials for a server authentication from here: [credentials](#)

Properties file  
a Properties file [credentials](#)

Json file  
a Json file

Maven toolchains.xml  
a toolchains.xml which can be referenced within Apache Maven jobs

Simple XML file  
a general xml file

Groovy file  
a reusable groovy script

Custom file  
a custom file (e.g. text or any other not yet available format)

Extended Email Publisher Groovy Template  
A Groovy template used by the Extended Email Publisher plugin to generate emails.

Extended Email Publisher Jelly Template  
A Jelly template used by the Extended Email Publisher plugin to generate emails.

ID  
ID of the config file  
672f2163-8b72-4c77-a5d7-e1225c807094

Then, provide the ID. Make sure it is something you can remember. In this case we will call it “**global-settings**”

Jenkins / Manage Jenkins / Managed files

Properties file  
a Properties file [credentials](#)

Json file  
a Json file

Maven toolchains.xml  
a toolchains.xml which can be referenced within Apache Maven jobs

Simple XML file  
a general xml file

Groovy file  
a reusable groovy script

Custom file  
a custom file (e.g. text or any other not yet available format)

Extended Email Publisher Groovy Template  
A Groovy template used by the Extended Email Publisher plugin to generate emails.

Extended Email Publisher Jelly Template  
A Jelly template used by the Extended Email Publisher plugin to generate emails.

ID  
ID of the config file  
global-settings

Next ←

Jenkins 2.528.3

Click on “**Next**”

The screenshot shows the Jenkins 'Edit Configuration File' page for a configuration named 'global-settings'. The page includes fields for ID, Name, Comment, and a 'Replace All' checkbox. A 'Server Credentials' section with a '+ Add' button is present. The 'Content' section displays XML code related to server configurations.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <!--
4 Licensed to the Apache Software Foundation (ASF) under one
5 or more contributor license agreements. See the NOTICE file
6 distributed with this work for additional information
7 regarding copyright ownership. The ASF licenses this file
```

Scroll down to content and go to “server” in the content

The screenshot shows the Jenkins 'Edit Configuration File' page with the XML content partially highlighted. The highlighted section is the <server> block, which contains deployment repository details. A 'Submit' button is visible at the bottom.

```
117 | used together.
118 |
119 <server>
120   <id>deploymentRepo</id>
121   <username>repouser</username>
122   <password>repopwd</password>
123 </server>
124 -->
125
126 <!-- Another sample, using keys to authenticate.
127 <server>
128   <id>siteServer</id>
```

We have to modify this.

Not secure 54.146.239.233:8080/manage/configfiles/addConfig

Jenkins / Manage Jenkins / Managed files

Name: MyGlobalSettings

Comment: Global settings

Replace All

Server Credentials: + Add

Content:

```
119  -->
120  <server>
121    <id>deploymentRepo</id>
122    <username>repouser</username>
123    <password>repowd</password>
124  </server>
125  -->
126
127  <!-- Another sample, using keys to authenticate.
128  <server>
129    <id>siteServer</id>
130    <privateKey>/path/to/private/key</privateKey>
```

Submit

Make sure you have (--) just before <server>

Not secure 54.146.239.233:8080/manage/configfiles/addConfig

Jenkins / Manage Jenkins / Managed files

Name: MyGlobalSettings

Comment: Global settings

Replace All

Server Credentials: + Add

Content:

```
119  -->
120  <server>
121    <id>deploymentRepo</id>
122    <username>repouser</username>
123    <password>repowd</password>
124  </server>
125  -->
126
127  <!-- Another sample, using keys to authenticate.
128  <server>
129    <id>siteServer</id>
130    <privateKey>/path/to/private/key</privateKey>
```

Submit

Then, remove the (--) after </server>

The screenshot shows the Jenkins 'Manage Jenkins' interface with the 'Managed files' configuration page selected. The 'Name' field is set to 'MyGlobalSettings'. The 'Comment' field contains the text 'Global settings'. The 'Replace All' checkbox is checked. In the 'Content' section, there is a code editor displaying the following XML code:

```
119    -->
120    <server>
121      <id>deploymentRepo</id>
122      <username>repouser</username>
123      <password>repowd</password>
124    </server>
125
126
127    <!-- Another sample, using keys to authenticate.
128    <server>
129      <id>siteserver</id>
130      <privateKey>/path/to/private/key</privateKey>
```

A red bracket highlights the first server block (lines 119-125). A blue bracket highlights the second server block (lines 127-130).

Then, copy the above part of code and duplicate it. We will use one for “**maven-releases**” and the other for “**maven-snapshots**”.

The screenshot shows the Jenkins 'Manage Jenkins' interface with the 'Managed files' configuration page selected. The 'Name' field is set to 'MyGlobalSettings'. The 'Comment' field contains the text 'Global settings'. The 'Replace All' checkbox is checked. In the 'Content' section, there are now two distinct server blocks, each enclosed in its own set of brackets:

```
119    -->
120    <server>
121      <id>deploymentRepo</id>
122      <username>repouser</username>
123      <password>repowd</password>
124    </server>
125
126    <server>
127      <id>deploymentRepo</id>
128      <username>repouser</username>
129      <password>repowd</password>
130    </server>
```

A red bracket highlights the first server block (lines 119-125). A blue bracket highlights the second server block (lines 127-130).

We have to add the repository name that is “**maven-release**”, then Nexus user name that is “**admin**” and the password that is “xxxxxxxx”.

The screenshot shows the Jenkins interface for managing configuration files. The URL is 54.146.239.233:8080/manage/configfiles/addConfig. The page title is "Jenkins / Manage Jenkins / Managed files". There is a "Comment" field with "Global settings" selected. A checkbox "Replace All" is checked. Below it is a "Server Credentials" section with a "+ Add" button. The "Content" area contains XML code for two servers:

```
119    -->
120    <server>
121      <id>maven-releases</id>
122      <username>admin</username>
123      <password>IloveBuea77!</password>
124    </server>
125
126    <server>
127      <id>maven-snapshots</id>
128      <username>admin</username>
129      <password>IloveBuea77!</password>
130    </server>
131
```

A blue "Submit" button is at the bottom.

Then, we create one for “**maven-snapshots**” as well.

This screenshot shows the same Jenkins configuration page after adding another server entry for "maven-snapshots". The "Content" area now includes both configurations:

```
119    -->
120    <server>
121      <id>maven-releases</id>
122      <username>admin</username>
123      <password>IloveBuea77!</password>
124    </server>
125
126    <server>
127      <id>maven-snapshots</id>
128      <username>admin</username>
129      <password>IloveBuea77!</password>
130    </server>
131
```

An orange arrow points from the text "Then, click on \"Submit\" to the blue "Submit" button at the bottom.

Then, click on “**Submit**”

The screenshot shows the Jenkins 'Config File Management' page. At the top, there's a navigation bar with icons for back, forward, search, and user profile. Below it, a sidebar on the left has links for 'Manage Jenkins', 'Config Files', and 'Add a new Config'. The main content area is titled 'Config File Management' with the sub-section 'Managed files'. It displays a table with one row:

E	D	Name	ID	Comment	Content Type
		MyGlobalSettings	global-settings	Global settings	

We will use this setting while writing the stage involving Nexus.

### 5.3.5 Create Credentials for Docker

We are going to create the credentials for Docker. Go to Jenkins page and navigate to “Manage Jenkins”

The screenshot shows the Jenkins 'Manage Jenkins' page. At the top, there's a header with the Jenkins logo, a search bar, and various management links. A prominent red box highlights a warning message about a CSRF vulnerability in the 'docker-build-step 2.12' plugin. Below the message are two buttons: 'Go to plugin manager' and 'Configure which of these warnings are shown'.

The main content area is divided into sections:

- System Configuration**: Includes links for 'System', 'Nodes', 'Appearance', and 'Managed files'.
- Security**: Includes links for 'Security'.
- Plugins**: Shows a list of installed plugins: 'Docker', 'Managed files', 'Clouds', and 'Credential Providers'.
- Credential Providers**: A section specifically for managing credentials.

A red arrow points from the text 'Click on "Credentials"' to the 'Configure credentials' link under the 'Credential Providers' section.

Click on “**Credentials**”

Jenkins / Manage Jenkins / Credentials

## Credentials

ebotsidneysmith/\*\*\*\*\* (git-credential)  
System - Global - git-credential - git-credential

sonar-token  
System - Global - sonar-token - sonar-token

Stores scoped to Jenkins

System Domains: Global

REST API Jenkins 2.528.3

Click on “Global”

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)

## Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ebotsidneysmith/\*\*\*\*\* (git-credential)  
git-credential - git-credential

sonar-token  
sonar-token - sonar-token

+ Add Credentials

Click on “Add Credentials”

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...)

## New credentials

Kind: Username with password

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

Username:

Treat username as secret

Password:

ID:

Description:

**Create**

Here, you have to enter the username of your Docker account. Login to the **Docker hub** on <https://hub.docker.com/>

https://hub.docker.com/repositories/ebotsidneysmith

Explore My Hub

ebotsidneysmith Docker Personal

**Repositories**  
All repositories within the ebotsidneysmith namespace.

Search by repository name All content

Create a repository

Name	Last Pushed	Contains	Visibility	Scout
ebotsidneysmith/testrepo	3 months ago	IMAGE	Public	Inactive
ebotsidneysmith/sosorepo	3 months ago	IMAGE	Public	Inactive
ebotsidneysmith/nodejswebapp	3 months ago	IMAGE	Public	Inactive
ebotsidneysmith/demoapp	3 months ago	IMAGE	Public	Inactive
ebotsidneysmith/demorepo	3 months ago	IMAGE	Public	Inactive

1–5 of 5

Click on your account icon

The screenshot shows the Docker Hub 'My Hub' interface. On the left, there's a sidebar with options like 'Repositories', 'Hardened Images', 'Collaborations', 'Settings', 'Default privacy', 'Notifications', 'Billing', 'Usage', 'Pulls', and 'Storage'. The main area is titled 'Repositories' and shows a list of repositories under the namespace 'ebotsidneysmith'. The repositories listed are: 'ebotsidneysmith/testrepo', 'ebotsidneysmith/sosrepo', 'ebotsidneysmith/nodejswebapp', 'ebotsidneysmith/demoapp', and 'ebotsidneysmith/demorepo'. Each repository entry includes the name, last pushed date (3 months ago), and a link to the image. A modal window is open on the right, showing the user profile 'ebotsidneysmith' with options like 'What's new', 'My profile', 'Account settings', and a 'Sign out' button. The Docker Hub URL is https://hub.docker.com/repositories/ebotsidneysmith.

Copy your username: **ebotsidneysmith**. Paste this user name in the Jenkins GUI and add the Docker Hub password.

The screenshot shows the Jenkins 'New credentials' screen. It's a 'Username with password' credential type. The fields filled are 'Scope' (Global), 'Username' (ebotsidneysmith), and 'Password'. An orange arrow points to the 'Password' input field. Below the fields are 'ID' and 'Description' fields, both empty. At the bottom is a 'Create' button. The URL in the browser is 35.172.225.123:8080/manage/credentials/store/system/domain/\_/newCredentials. The Jenkins UI has a sidebar with icons for Manage Jenkins, Credentials, System, and Global credentials.

On the “**Password**”, enter the password of your Docker hub account.

Not secure 35.172.225.123:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...)

### New credentials

Kind: Username with password

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

Username: ebotsidneysmith

Treat username as secret

Password:

ID:  (arrow points here)

Description:

Create

Then on "ID", enter "**docker-cred**"

Not secure 35.172.225.123:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...)

### New credentials

Kind: Username with password

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

Username: ebotsidneysmith

Treat username as secret

Password:

ID:  (arrow points here)

Description:

Create

Then, for "Description", also enter "**docker-cred**"

New credentials

Kind

Username with password

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

Username ebotsidneysmith

Treat username as secret ?

Password .....  
.....

ID ? docker-cred

Description ? docker-cred

Create

Then, click on “Create”

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ebotsidneysmith***** (git-credential) git-credential - git-credential	⋮
sonar-token sonar-token - sonar-token	⋮
ebotsidneysmith***** (docker-cred) docker-cred - docker-cred	⋮

+ Add Credentials

The Docker credential has been created. We will use this on the Docker Stage in the Pipeline.

### 5.3.6 Create Credentials for Kubernetes

We will have to connect Jenkins to the Kubernetes cluster. To do this, we will have to generate a token that will be used for authentication.

For deploying the application to Kubernetes cluster in a proper way, we can create a service account or we can use *Role-Based Access Control (RBAC)*.

For the best practice, we cannot give complete access to everyone. To do this, we have to create “**Roles**” that we can assign to specific “**users**”. This is what we are going to do for our deployment.

**Create Service Account, Role & Assign that role, and create a secret for Service Account and generate a Token**

Below are the codes to create a service account

### **Creating Service Account**

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins
  namespace: webapps
```

### **Create Role**

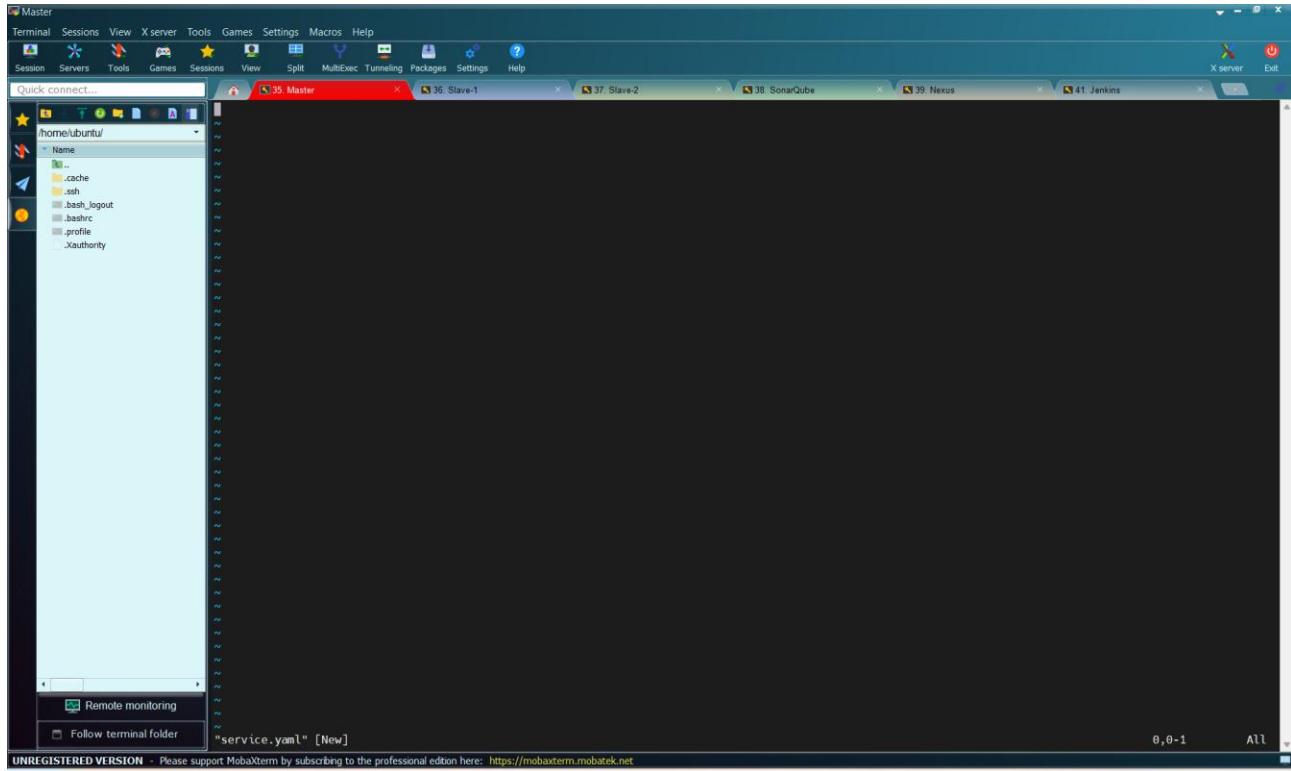
```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: app-role
  namespace: webapps
rules:
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - extensions
  - policy
  - rbac.authorization.k8s.io
resources:
- pods
- secrets
- componentstatuses
- configmaps
- daemonsets
- deployments
- events
- endpoints
- horizontalpodautoscalers
- ingress
- jobs
- limitranges
- namespaces
- nodes
- pods
- persistentvolumes
- persistentvolumeclaims
- resourcequotas
- replicasesets
- replicationcontrollers
- serviceaccounts
- services
verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]
```

## Bind the role to service account

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

Let us start by creating a service account. Go to the “Master” server and create the file called “**service.yaml**” using the command:

```
vi service.yaml
```



Then paste the code below to create the service account:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins
  namespace: webapps
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins
  namespace: webapps
```

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

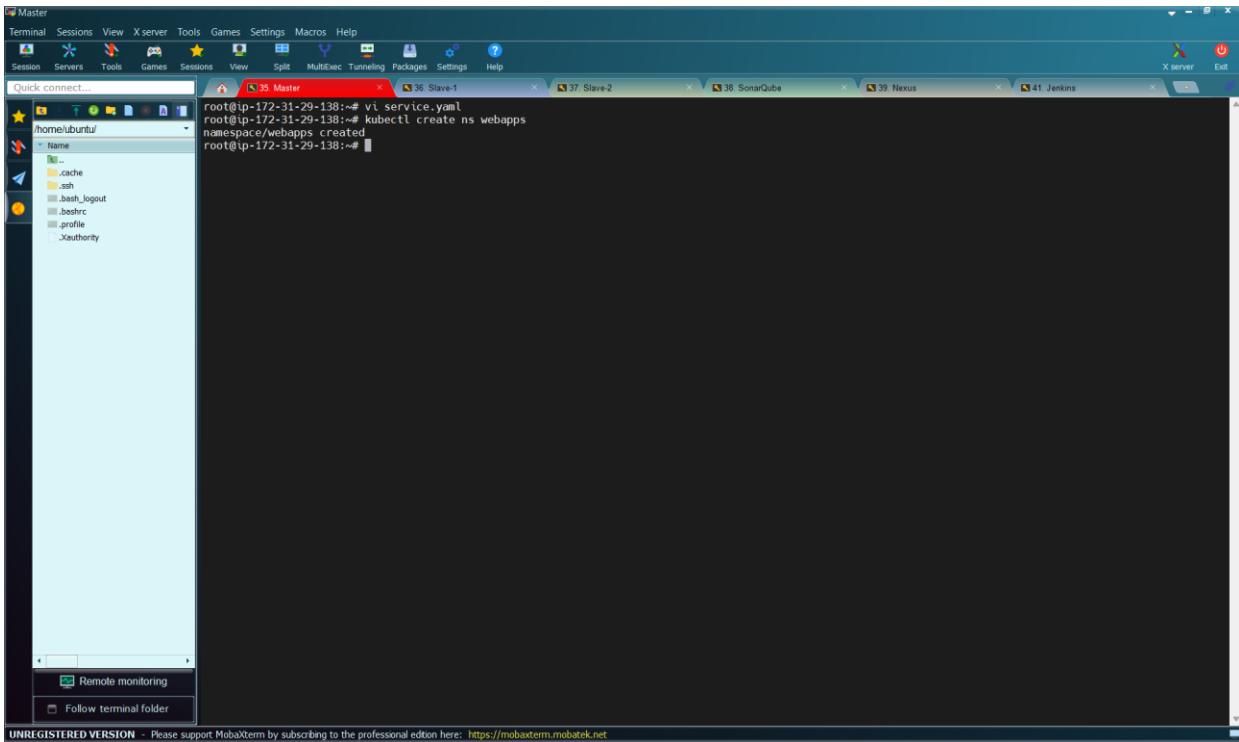
Save the file by pressing “**ESC**” followed by “**:wq**” and press “**Enter**”

```
root@ip-172-31-29-138:~# vi service.yaml
```

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

Now, let us create the namespace “**webapps**” using the command:

```
kubectl create ns webapps
```



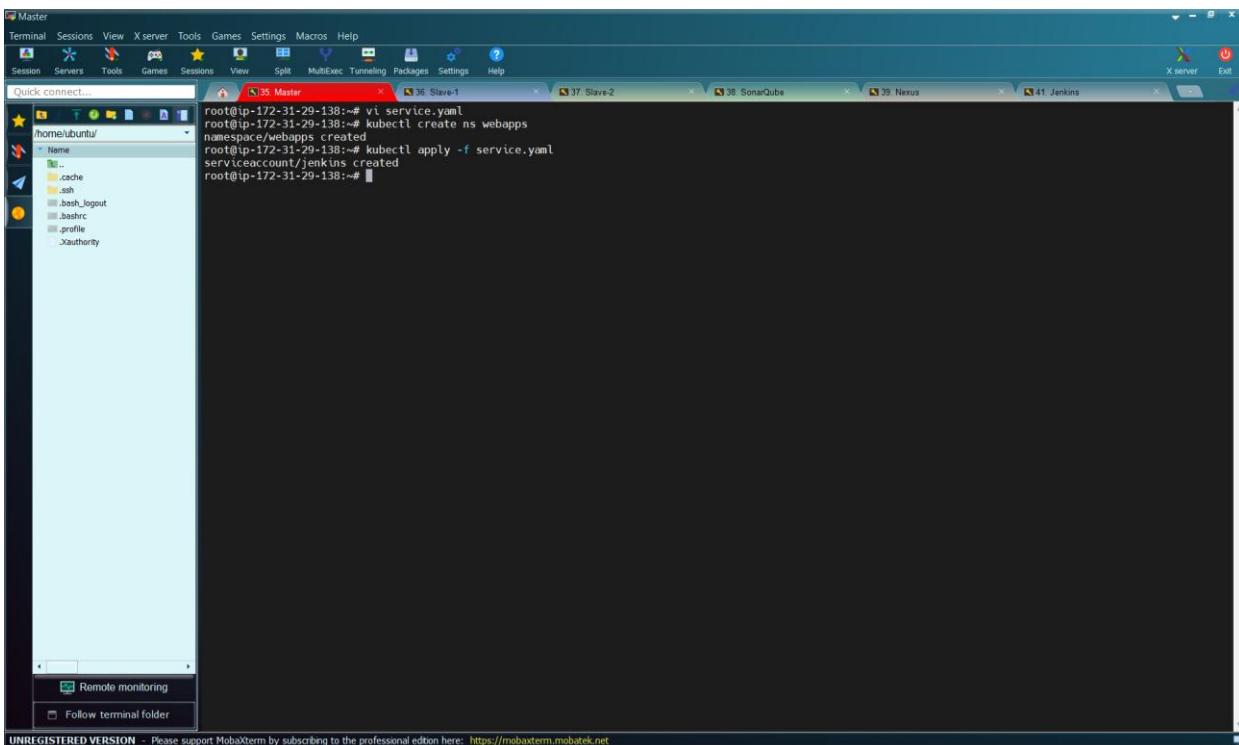
The screenshot shows a terminal window in MobaXterm titled 'Master'. The user is root and has run the command 'vi service.yaml' to edit a configuration file. They then run 'kubectl create ns webapps', which creates a new namespace named 'webapps'. The terminal window also shows other open sessions like 'Slave-1', 'Slave-2', 'SonarQube', 'Nexus', and 'Jenkins'.

```
root@ip-172-31-29-138:~# vi service.yaml
root@ip-172-31-29-138:~# kubectl create ns webapps
namespace/webapps created
root@ip-172-31-29-138:~#
```

The namespace has been created. Now, let us execute the Service Account command to create the service account:

```
kubectl apply -f service.yaml
```

This will create the service account inside the namespace



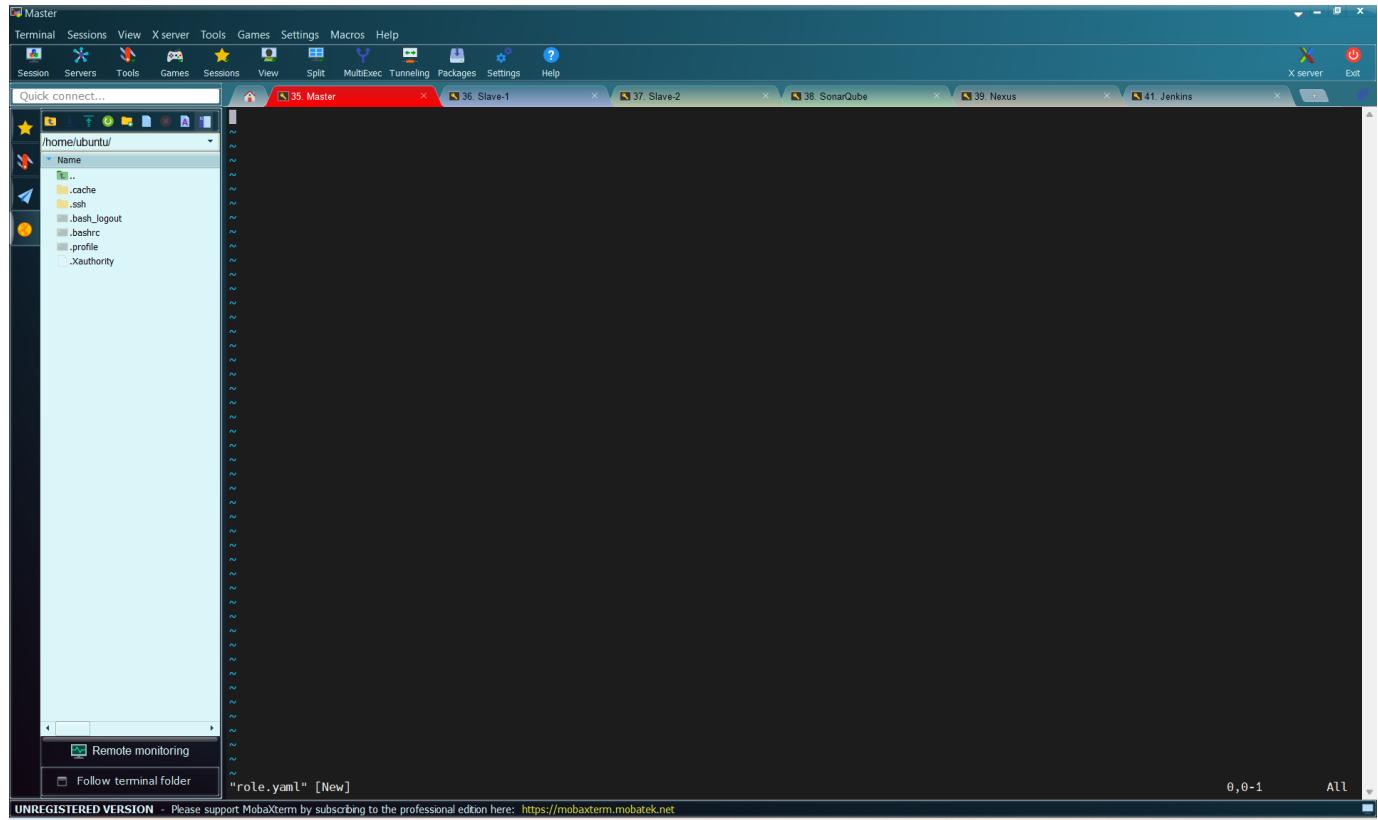
The screenshot shows a terminal window in MobaXterm titled 'Master'. The user is root and has run 'vi service.yaml' to edit a configuration file. They then run 'kubectl apply -f service.yaml', which creates a service account named 'jenkins' inside the 'webapps' namespace. The terminal window also shows other open sessions like 'Slave-1', 'Slave-2', 'SonarQube', 'Nexus', and 'Jenkins'.

```
root@ip-172-31-29-138:~# vi service.yaml
root@ip-172-31-29-138:~# kubectl create ns webapps
namespace/webapps created
root@ip-172-31-29-138:~# kubectl apply -f service.yaml
serviceaccount/jenkins created
root@ip-172-31-29-138:~#
```

The service account has been created.

The next thing is to create the role. This role has complete permissions. To do this, we will create a file called “**role.sh**”. Let us create the file using the command on the “**Master**” server:

```
vi role.yaml
```



Paste the code below:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: app-role
  namespace: webapps
rules:
  - apiGroups:
      - ""
      - apps
      - autoscaling
      - batch
      - extensions
      - policy
      - rbac.authorization.k8s.io
    resources:
      - pods
      - secrets
      - componentstatuses
      - configmaps
      - daemonsets
```

```

- deployments
- events
- endpoints
- horizontalpodautoscalers
- ingress
- jobs
- limitranges
- namespaces
- nodes
- pods
- persistentvolumes
- persistentvolumeclaims
- resourcequotas
- replicases
- replicationcontrollers
- serviceaccounts
- services

verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]

```

```

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: app-role
  namespace: webapps
rules:
  - apiGroups:
      - ""
      - apps
      - autoscaling
      - batch
      - extensions
      - policy
      - rbac.authorization.k8s.io
    resources:
      - pods
      - secrets
      - componentstatuses
      - configmaps
      - daemonsets
      - deployments
      - events
      - endpoints
      - horizontalpodautoscalers
      - ingress
      - jobs
      - limitranges
      - namespaces
      - nodes
      - pods
      - persistentvolumes
      - persistentvolumeclaims
      - resourcequotas
      - replicases
      - replicationcontrollers
      - serviceaccounts
      - services
  verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]

```

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

Save and exit the file by pressing “**ESC**” followed by “**:wq**” and press “**Enter**”

```
root@ip-172-31-29-138:~# vi role.yaml
```

Then, run the command to create the role:

```
kubectl apply -f role.yaml
```

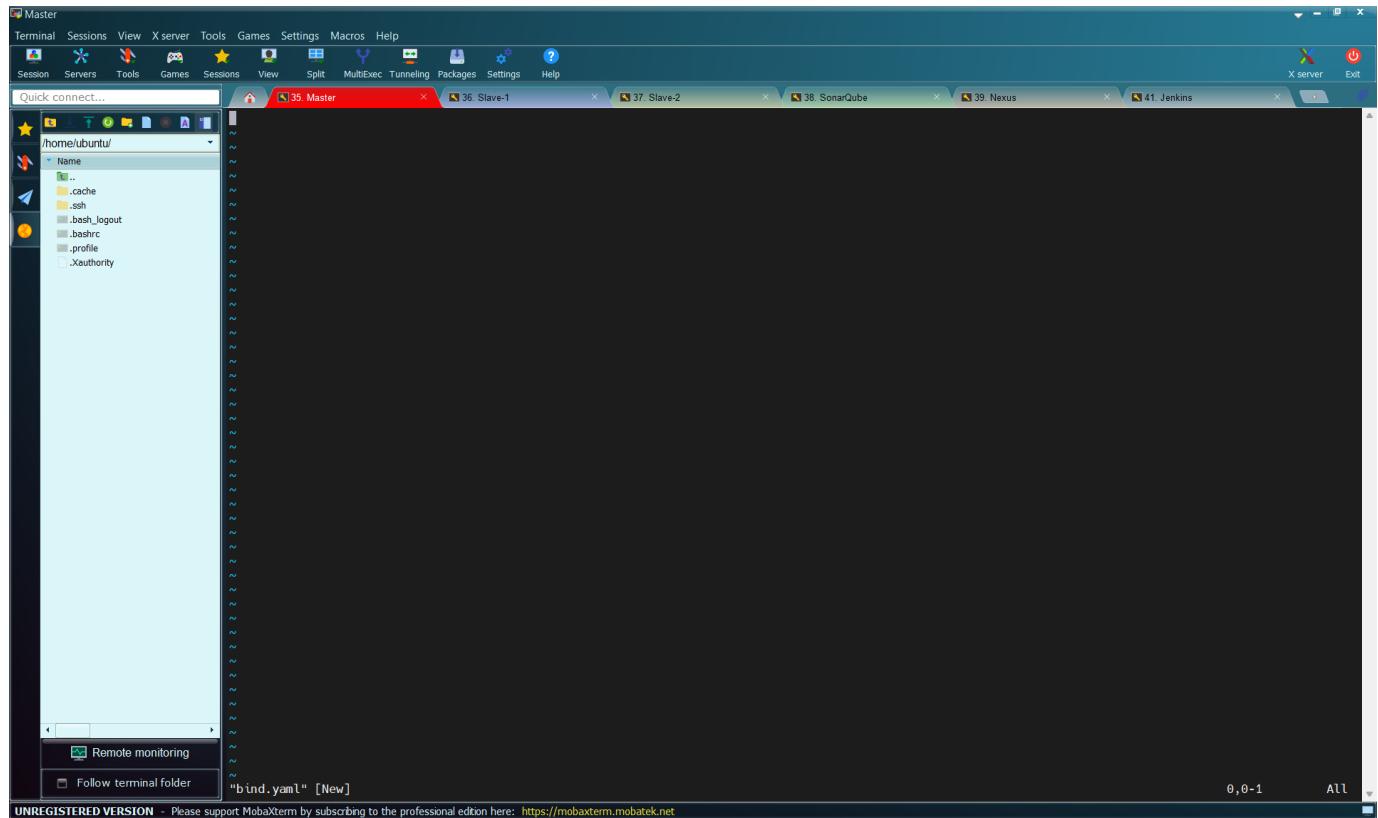
```
root@ip-172-31-29-138:~# vi role.yaml
root@ip-172-31-29-138:~# kubectl apply -f role.yaml
role.rbac.authorization.k8s.io/app-role created
root@ip-172-31-29-138:~#
```

The role has been created.

## Bind the role to service account

Now, let us assign the role to the service account. To do this, we have to create another yaml file called "**bind.yaml**". Let us run the command to create the file

```
vi bind.yaml
```



Paste the code below:

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

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

Save and exit the file by pressing “**ESC**”, followed by “**:wq**” and press “**Enter**”

```
root@ip-172-31-29-138:~# vi role.yaml
root@ip-172-31-29-138:~# kubectl apply -f role.yaml
role.rbac.authorization.k8s.io/app-role created
root@ip-172-31-29-138:~# vi bind.yaml
root@ip-172-31-29-138:~#
```

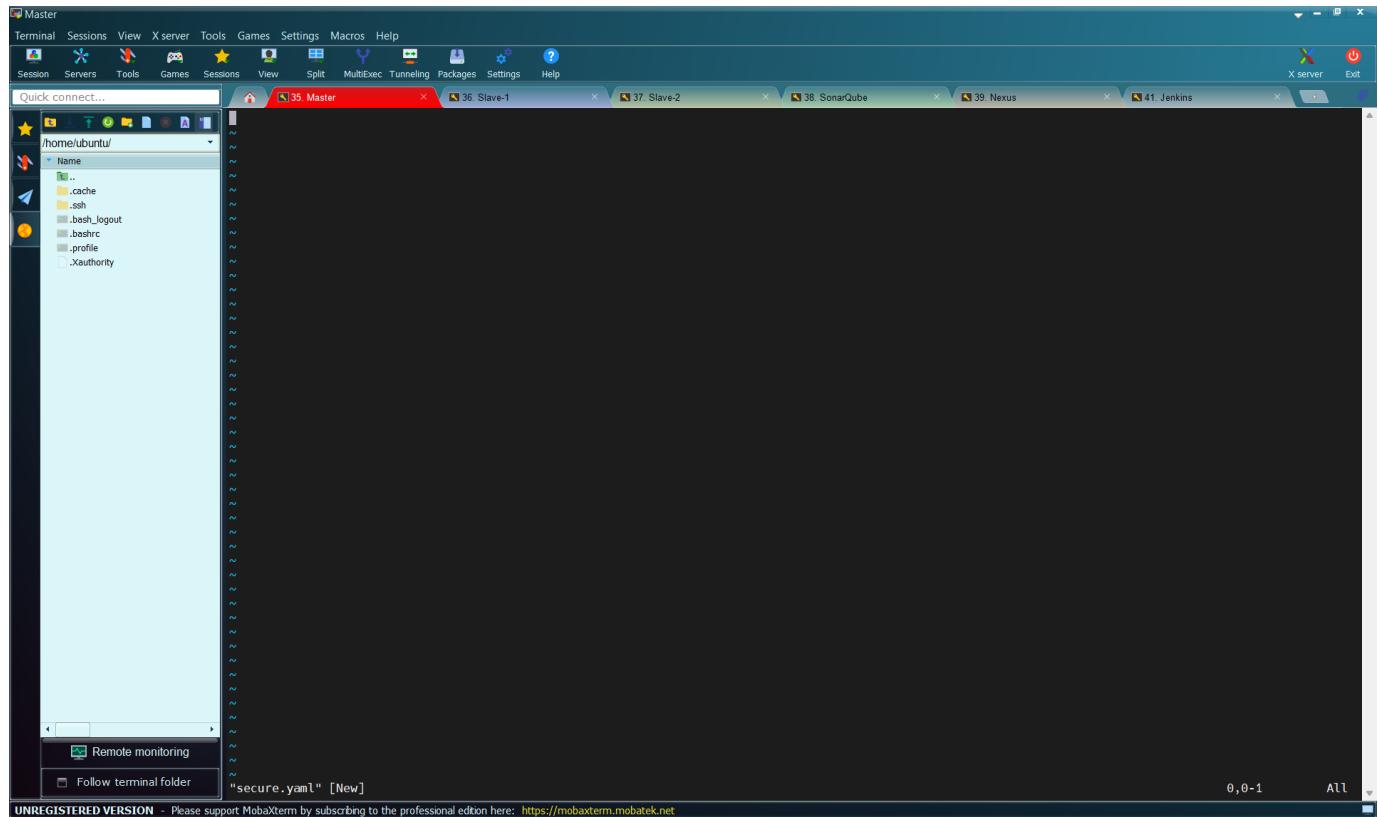
Then, let us run the command to bind them:

```
kubectl apply -f bind.yaml
```

Let us create a yaml file called “**secure.yaml**” for this.

The file is created using this command on the Master server:

```
vi secure.yaml
```



Then to get the code for this file, go to

<https://kubernetes.io/docs/reference/access-authn-authz/service-accounts-admin/>

← → ⌛ kubernetes.io/docs/reference/access-authn-authz/service-accounts-admin/ ⚙️ 🌐 🔍 ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

kubernetes
Documentation
Kubernetes Blog
Training
Careers
Partners
Community
Versions ▾
English ▾
🔍 Search this site

🔍 Search this site

- ▶ Documentation
- ▶ Getting started
- ▶ Concepts
- ▶ Tasks
- ▶ Tutorials
- ▼ Reference
  - Glossary
  - ▶ API Overview
  - ▼ API Access Control
    - Authenticating
    - Authenticating with Bootstrap Tokens
    - Authorization
    - Using RBAC
    - Authorization
    - Using Node Authorization
    - Webhook Mode
    - Using ABAC
    - Authorization
    - Admission Control

Kubernetes Documentation / Reference / API Access Control / Managing Service Accounts

## Managing Service Accounts

A *ServiceAccount* provides an identity for processes that run in a Pod.

A process inside a Pod can use the identity of its associated service account to authenticate to the cluster's API server.

For an introduction to service accounts, read [configure service accounts](#).

This task guide explains some of the concepts behind ServiceAccounts. The guide also explains how to obtain or revoke tokens that represent ServiceAccounts, and how to (optionally) bind a ServiceAccount's validity to the lifetime of an API object.

### Before you begin

You need to have a Kubernetes cluster, and the `kubectl` command-line tool must be configured to communicate with your cluster. It is recommended to run this tutorial on a cluster with at least two nodes that are not acting as control plane hosts. If you do not already have a cluster, you can create one by using [minikube](#) or you can use one of these Kubernetes playgrounds:

- [iximiuz Labs](#)
- [Killercoda](#)
- [KodeKloud](#)

[Edit this page](#)  
[Create child page](#)  
[Create an issue](#)  
[Print entire section](#)

Before you begin

User accounts versus service accounts

Bound service account tokens

Additional metadata in Pod bound tokens

Verifying and inspecting private claims

Bound service account token volume mechanism

Manual Secret management for ServiceAccounts

Auto-generated legacy ServiceAccount token clean up

Control plane details

ServiceAccount controller

Token controller

ServiceAccount admission controller

Legacy ServiceAccount token tracking controller

Legacy ServiceAccount token cleaner

TokenRequest API

Create additional API tokens

Delete/invalidate a ServiceAccount token

Delete/invalidate a long-lived/legacy ServiceAccount token

Scroll down to “Create Additional API Token”

← → ⌛ kubernetes.io/docs/reference/access-authn-authz/service-accounts-admin/ ⚙️ 🌐 🔍 ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

kubernetes
Documentation
Kubernetes Blog
Training
Careers
Partners
Community
Versions ▾
English ▾
🔍 Search this site

🔍 Search this site

- ▶ Documentation
- ▶ Getting started
- ▶ Concepts
- ▶ Tasks
- ▶ Tutorials
- ▼ Reference
  - Glossary
  - ▶ API Overview
  - ▼ API Access Control
    - Authenticating
    - Authenticating with Bootstrap Tokens
    - Authorization
    - Using RBAC
    - Authorization
    - Using Node Authorization
    - Webhook Mode
    - Using ABAC
    - Authorization
    - Admission Control
    - Dynamic Admission Control

## Create additional API tokens

**Caution:**

Only create long-lived API tokens if the [token request](#) mechanism is not suitable. The token request mechanism provides time-limited tokens; because these expire, they represent a lower risk to information security.

To create a non-expiring, persisted API token for a ServiceAccount, create a Secret of type `kubernetes.io/service-account-token` with an annotation referencing the ServiceAccount. The control plane then generates a long-lived token and updates that Secret with that generated token data.

Here is a sample manifest for such a Secret:

```
secret/serviceaccount/myserviceaccount.yaml
```

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

To create a Secret based on this example, run:

Copy this code above

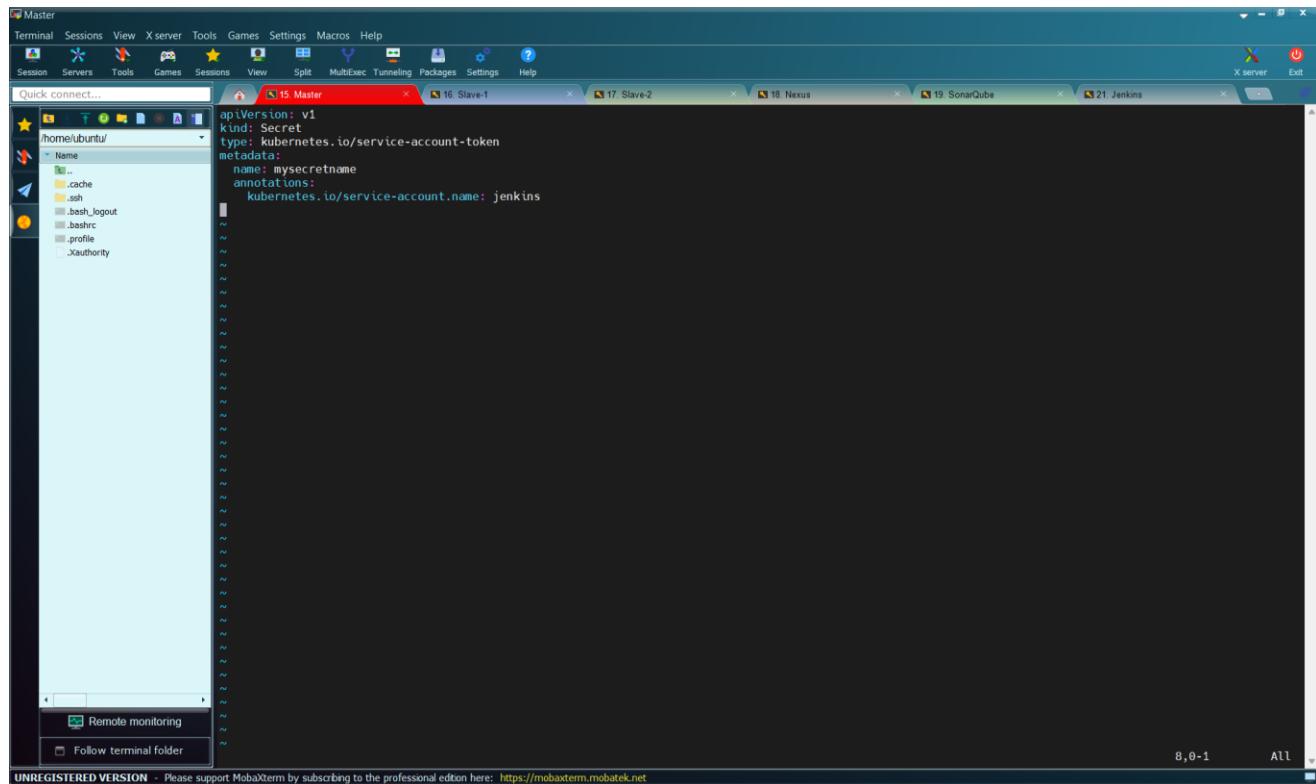
```
apiVersion: v1
kind: Secret
```

```
type: kubernetes.io/service-account-token
metadata:
  name: mysecretname
  annotations:
    kubernetes.io/service-account.name: myserviceaccount
```

Change “**myserviceaccount**” to our service account name “**jenkins**”

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

Copy and paste this in the “**secure.yaml**” file.



Then, save and exit this file by pressing “**ESC**” followed by “**:wq**” then press “**Enter**”.

```
root@ip-172-31-22-72:~# vi secure.yaml
```

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

We did not provide the namespace in this file. We can provide it while running the command to generate the token. We will run the command:

```
kubectl apply -f secure.yaml -n webapps
```

```
root@ip-172-31-29-138:~# vi secure.yaml
root@ip-172-31-29-138:~# kubectl apply -f secure.yaml -n webapps
secret/mysecretname created
root@ip-172-31-29-138:~#
```

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

We have created the token. To see the token, run the command:

```
kubectl describe secret mysecretname -n webapps
```

Master

Terminal Sessions View X server Tools Games Settings Macros Help

Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

X server Exit

Quick connect... 15. Master 16. Slave-1 17. Slave-2 18. Nexus 19. SonarQube 21. Jenkins

/home/ubuntu/

Name: mysecretname  
Namespace: webapps  
Labels: <none>  
Annotations: kubernetes.io/service-account.name: jenkins  
kubernetes.io/service-account.uid: ef316fdf-5621-a043-4a580d2b3c95  
Type: kubernetes.io/service-account-token

Data

ca.crt: 1107 bytes  
namespace: 7 bytes  
token: eyJhbGciOiJSUzI1NiIsImtpZCI6IkcvcHV6NnBYN1pWLXA4bmfMwktwhCMXL0tLR1Z003a0E1TzFVYk5jb1kif0.eyJpc3MiOiJrdWJlcmlzdGVzL3Nlcnp2Y2VhY2NvdW50Iiwia3ViZXJuXRlcypby9zZXj2awNlYWNjb3VudC9uYWh1c3BhY2U0U3ZWJhcHBzIwiav3VzXJuZXRLcy5pb9zXJ2aWNTLYNjb3VudC9zZWNyZXQubmFtZS16lm15c2VjcmVbmfFtZS1sImt1YmYvbmV0ZXMuawBvc2VydmljZWFjZ291bnvc2VydmljZS1hZ2Vndf50Lmh5bwUiOJqZw5raw5zIwiav3VzXJuZXRLcy5pb9zXJ2aWNTLYNjb3VudC9zZXJ2aWNLWFjY291bnqudWkjoiZWyZMTZmZGYtNTYyMS00NzYxLWEwNDMtNGE10DBkMnIzYzK1Iwiic3ViIjoi3LzdGvt0nNLcnZpY2VhY2NvdW50ondYmFwchM6amVua2lucyJ9\_gb-n5flZg02XuastBXfmw3jAhbVQldhBNd81cFFFE034ZUhAhPVTbxjYe0F1qGKMSG1CzQam4zG\_2qPMxeyz9hBWrJMtgTv5bmJ512WPfzyxpcuL9JMWv5Db5IC\_-8zQLf\_-xxgTyf7DNCtzjV71SA674f06VXBvh0sfshCw\_jwyF\_7022sHkvEk0ZKgk27gYFPe9taFnJDffs4MJErzLXm5tZ3tFLNg0nahizIuMDra600aw2PrFzbysV6HLEUwqs7YGH\_51Y6fbJ3dzeZVMfh3yxIyrXtdgJ2A7md2UM-asA12IrRd8ibdrZnLdn4aA

root@ip-172-31-22-72:~#

Remote monitoring  
Follow terminal folder

You can see the token. Copy this token.

eyJhbGciOiJSUzI1NlslmtpZCI6IkwcHV0NnBYN1pWLXA4bmpFMWktWnhCMXIOTIR1Z003a0E1TzFVYk5jb1kifQ.eyJpc3MiOiJrdWJlcmt5ldGVzL3NlcnpY2VhY2NvdW50liwia3ViZXJuZXRLcy5pb9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOij3ZWJhcHBzliwia3ViZXJuZXRLcy5pb9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6Im15c2VjcmV0bmFtZSlsm1t1YmVybmv0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50Lm5hbWUiOijqZW5raW5zliwia3ViZXJuZXRLcy5pb9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlWFjY291bnQudWIkljoiZWYzMTZmZGYtNTYyMS00NzYxLWEwNDMtNGE1ODBkMmlzYzk1liwiic3Viljoic3lzdGVtOnNlcnpY2VhY2NvdW50OndlYmFwcHM6amVua2lucyJ9.gb-nSfLZg02Xua6t8Xfmnw3jAhvBQldhBNd81cfFFEO34ZUhhAHpVTbxjYe0F1qGKMSG1Cz\_Qam4zG\_2qGPMxeyz9hBWrJMjgTv5BmJSI2WPfZyqXpcuLL9JMvwv5DBsIC\_-8z9LDfxXgWTyF7DNCTzjV7ISAA6714r0aVX8vhsQbfsHcW\_jWy3F\_7Q22sHkvieKQZKGk27gYFPed9taiFnJDffs4MJErzLXm5tZ3TFIN6DnaHzlluMDr6a000aw2PrYFzbsyV6HLEUwvqs7YGH\_5lY6fbJ3dzeZVMfh3yxiyrXtdgZl2AZmd2UM-asAI2lrRd8ibdrZnLdNr4aA

[Go back to “Manage Jenkins” in Jenkins](#)

The screenshot shows the Jenkins Manage Jenkins interface. At the top, there is a red banner with the text "Java 17 end of life in Jenkins" and a message about support ending on March 31, 2026. Below the banner, the "System Configuration" section is visible, containing links for System, Tools, Plugins, Nodes, Docker, Managed files, Appearance, Security, and Credential Providers. An orange arrow points to the "Credentials" link under the Security section.

**Java 17 end of life in Jenkins**

You are running Jenkins on Java 17, support for which will end on or after Mar 31, 2026. Refer to [the documentation](#) for more details.

**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.
- Docker** Plugin for launching build Agents as Docker containers
- Clouds** Add, remove, and configure cloud instances to provision agents on-demand.
- Appearance** Configure the look and feel of Jenkins
- Managed files** e.g. settings.xml for maven, central managed scripts, custom files, ...
- Credentials** Configure credentials
- Credential Providers** Configure the credential providers and types

**Security**

- Security** Secure Jenkins; define who is allowed to access/use the system.
- Users** Create accounts for users that can log in to this Jenkins instance.

## Click on Credentials

The screenshot shows the "Credentials" page under the "Manage Jenkins" menu. It displays a list of three credentials: "ebotsidneysmith/\*\*\*\*\*\*\*\* (git-credential)", "sonar-token", and "ebotsidneysmith/\*\*\*\*\*\*\*\* (docker-cred)". Below the list, there is a section titled "Stores scoped to Jenkins" with a "System" option and a "Domains: Global" dropdown. An orange arrow points to the "Global" dropdown.

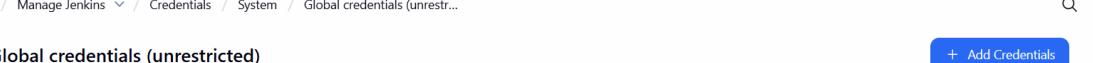
**Credentials**

- ebotsidneysmith/\*\*\*\*\*\*\*\* (git-credential)
- sonar-token
- ebotsidneysmith/\*\*\*\*\*\*\*\* (docker-cred)

Stores scoped to Jenkins

System Domains: Global

## Click on "Global"



The screenshot shows the Jenkins Global credentials page. At the top, there's a navigation bar with links for Jenkins, Manage Jenkins, Credentials, System, and Global credentials (unrestricted...). On the right side of the header are various Jenkins management icons. The main title is "Global credentials (unrestricted)". Below it, a sub-instruction reads: "Credentials that should be available irrespective of domain specification to requirements matching." A blue button labeled "+ Add Credentials" is positioned on the right. The list of credentials includes:

- ebotsidneysmith/\*\*\*\*\*\*\*\* (git-credential)  
git-credential - git-credential
- sonar-token  
sonar-token - sonar-token
- ebotsidneysmith/\*\*\*\*\*\*\*\* (docker-cred)  
docker-cred - docker-cred

**Click on “Add Credentials”**

Not secure 54.146.239.233:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...

## New credentials

Kind

Username with password

Scope ?

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

Username

Blank username; did you mean to use secret text credentials instead?

Treat username as secret ?

Password

ID ?

Description ?

Create

Then on “Kind”, click on the drop down and select “secret text”

Not secure 54.146.239.233:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...)

### New credentials

Kind

Secret text

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

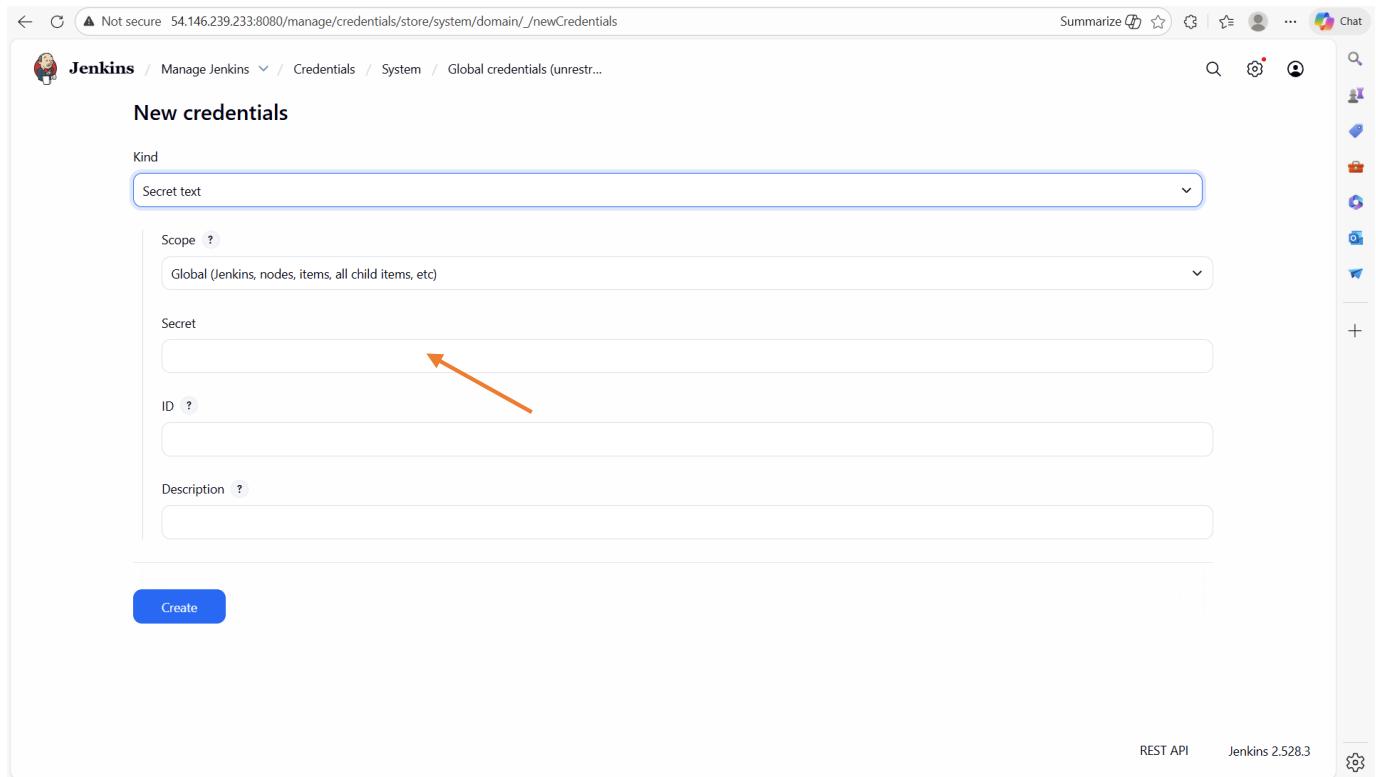
Secret

ID ?

Description ?

Create

REST API Jenkins 2.528.3



On “secret” field, paste the token we copied.

Not secure 54.146.239.233:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...)

### New credentials

Kind

Secret text

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

Secret

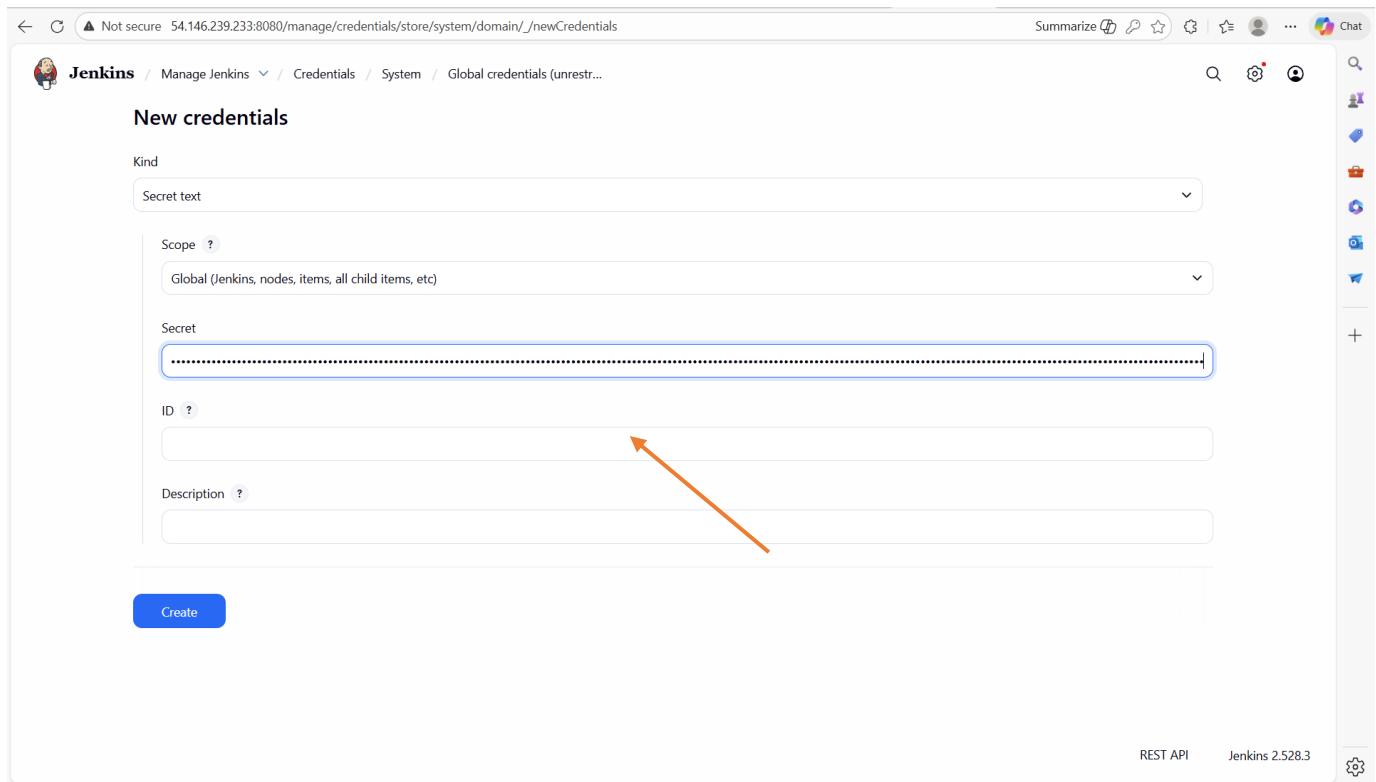
.....

ID ?

Description ?

Create

REST API Jenkins 2.528.3



Then, for “ID” and “Description”, enter “k8-cred”

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted...)

### New credentials

Kind: Secret text

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

Secret:

ID: k8-cred

Description: k8-cred

**Create**

REST API Jenkins 2.528.3

Click on “Create”

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted...)

### Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ebotsidneysmith/***** (git-credential) git-credential - git-credential	...
sonar-token sonar-token - sonar-token	...
ebotsidneysmith/***** (docker-cred) docker-cred - docker-cred	...
<b>k8-cred</b> k8-cred - k8-cred	...

+ Add Credentials

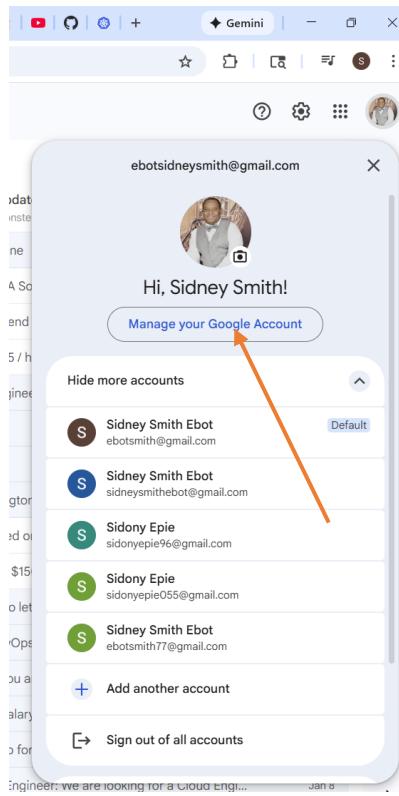
We have created the token for Kubernetes.

### 5.3.7 Create Credentials for Email Notifications

Let us now create the credentials for email notification.

#### 5.3.7.1 Create the App Password of GMail

We have to first get the credentials of the Gmail account. Let us go to Gmail account.



Click on “Manage Your Google Account”

A screenshot of the 'Google Account' settings page. The URL is 'myaccount.google.com/u/2/?hl=en&utm\_source=OGB&utm\_medium=act&gar=WzEyMjQ='. The page title is 'Google Account'. On the left, there is a sidebar with various settings categories:

- Home
- Personal info
- Security & sign-in (highlighted with a red arrow)
- Google password
- Third-party connections
- Data & privacy
- People & sharing
- Payments & subscriptions

The main area shows the user's profile picture and name: 'Sidney Smith Ebot' and 'ebotsidneysmith@gmail.com'. Below the profile is a search bar 'Search Google Account' and a row of buttons: 'My password', 'Devices', 'Password Manager', 'My Activity', and 'Email'. At the bottom, there is a note about Google keeping data private and secure, with a 'Learn more' link, and links for 'Privacy', 'Terms', 'Help', and 'About'.

Click on “Security & Sign-in”

**Security & sign-in**

Your account is protected  
The Security Checkup checked your account and found no recommended actions

Recent security activity  
No security activity or alerts in the last 28 days

How you sign in to Google  
Make sure you can always access your Google Account by keeping this information up to date

<b>2-Step Verification</b> On since Nov 20, 2025
<b>Passkeys and security keys</b> 3 passkeys
<b>Password</b> Last changed Oct 2, 2022
<b>Skip password when possible</b> On
<b>Authenticator</b> Added Nov 20, 2025

Privacy Terms Help About

# Search for “app password”

The screenshot shows the Google Account settings interface. On the left, there's a sidebar with various account management links like Home, Personal info, Security & sign-in, Google password, Third-party connectivity, Data & privacy, People & sharing, and Payments & subscriptions. The main area has a search bar at the top with the query "app password". Below it, a list of "Google Account results" is displayed, including "Password Manager Security", "Password Personal info, Security", "App passwords Security" (which is highlighted with an orange arrow), "Web & App Activity Data & privacy", and "Help Center articles". A modal window titled "Review security activity" is open over the results. At the bottom of the main area, there's a section titled "How you sign in to Google" with a sub-section "2-Step Verification" showing "On since 9:59 AM". A small pop-up in the bottom right corner asks for feedback on Google Account settings.

### Select “App passwords”

← App passwords

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.

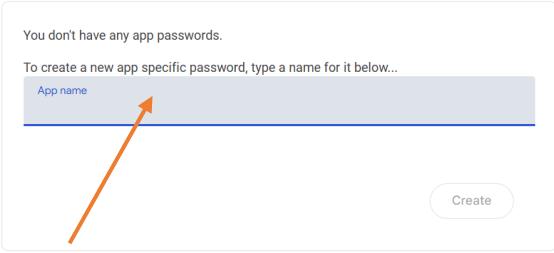
[Learn more](#)

You don't have any app passwords.

To create a new app specific password, type a name for it below...

App name

Create



Privacy Terms Help About

Enter the “App Name”, we will call it “jenkins”

← App passwords

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.

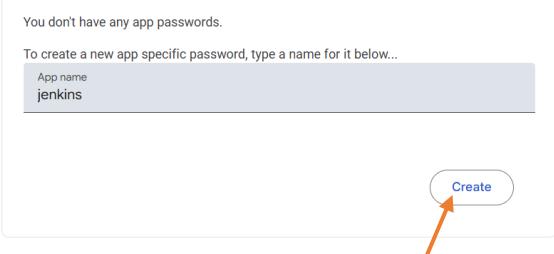
[Learn more](#)

You don't have any app passwords.

To create a new app specific password, type a name for it below...

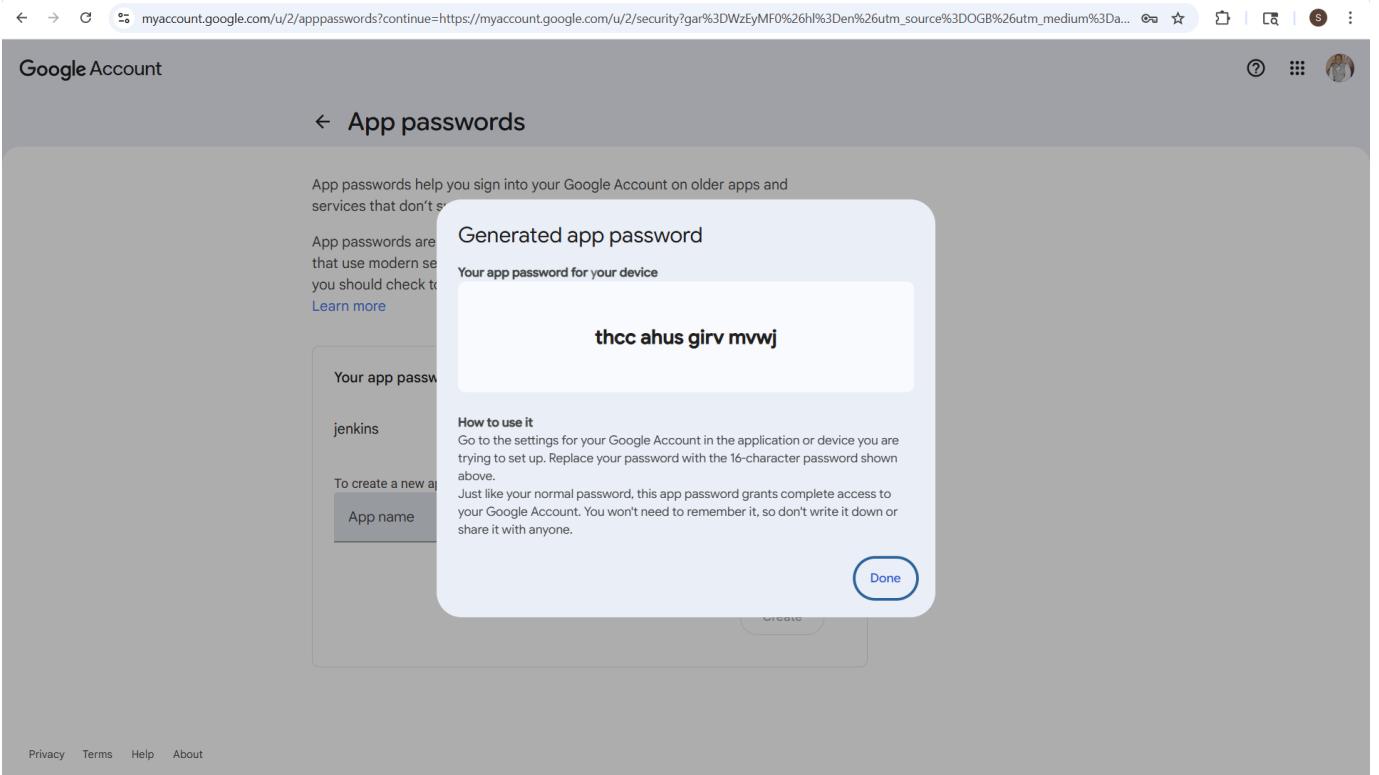
App name  
jenkins

Create



Privacy Terms Help About

Click on “Create”



The password has been created. Copy the password:

thcc ahus girv mvwj

We will use this password for sending mail notifications.

### 5.3.7.2 Create the App Password of GMail

Here, we are going to create the credentials on Jenkins. To do this, we will have to go to "**Manage Jenkins**" dashboard.

The screenshot shows the Jenkins System Configuration page. At the top, there is a red banner with the text "Java 17 end of life in Jenkins" and a message about running Jenkins on Java 17. Below the banner, the "System Configuration" section is visible, containing several configuration items: System, Tools, Plugins, Nodes, Docker, Appearance, Managed files, Security, and Credential Providers. The "Credentials" item is highlighted with an orange arrow pointing to it.

**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.
- Docker**: Plugin for launching build Agents as Docker containers.
- CLOUDS**: Add, remove, and configure cloud instances to provision agents on-demand.
- Appearance**: Configure the look and feel of Jenkins.
- Managed files**: e.g. settings.xml for maven, central managed scripts, custom files, ...
- Security**:
  - Security**: Secure Jenkins; define who is allowed to access/use the system.
  - Users**: Create/del data/modifi users that can log in to this Jenkins
- Credential Providers**: Configure the credential providers and types.

Click on “**Credentials**”

The screenshot shows the Jenkins Credentials page. It displays a list of credentials: ebotsidneysmith/\*\*\*\*\*\*\*\* (git-credential), sonar-token, ebotsidneysmith/\*\*\*\*\*\*\*\* (docker-cred), and k8-cred. Below the list, there is a section titled "Stores scoped to Jenkins" with a "System" button and a "Domains: Global" dropdown. An orange arrow points to the "Global" dropdown.

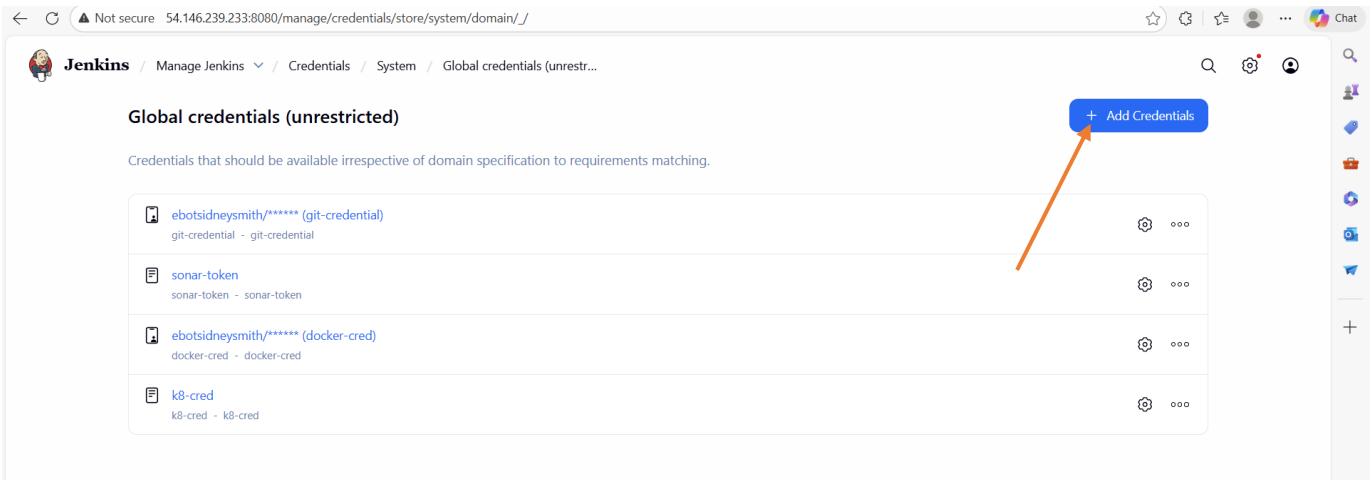
**Credentials**

- ebotsidneysmith/\*\*\*\*\*\*\*\* (git-credential)
- sonar-token
- ebotsidneysmith/\*\*\*\*\*\*\*\* (docker-cred)
- k8-cred

Stores scoped to Jenkins

System Domains: Global

Click on “**Global**”

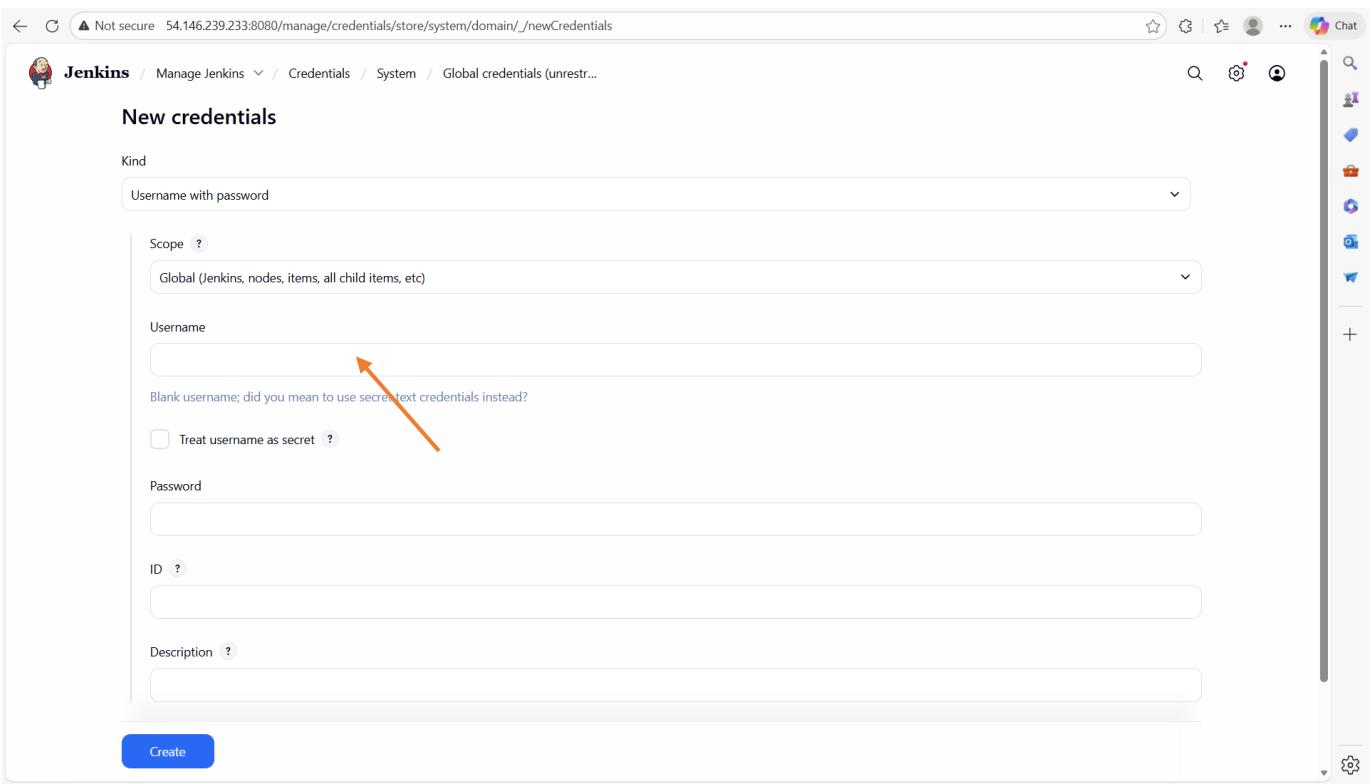


A screenshot of the Jenkins Global credentials (unrestricted) page. The page shows a list of existing credentials:

- ebotsidneysmith/\*\*\*\*\*\*\*\* (git-credential)
- sonar-token
- ebotsidneysmith/\*\*\*\*\*\*\*\* (docker-cred)
- k8-cred

A blue button labeled "+ Add Credentials" is located in the top right corner, with an orange arrow pointing towards it.

Click on “Add Credentials”



A screenshot of the Jenkins New credentials creation page. The "Kind" dropdown is set to "Username with password". The "Scope" dropdown is set to "Global (Jenkins, nodes, items, all child items, etc)". The "Username" field is highlighted with an orange arrow. The "Password" and "ID" fields are also present. A "Create" button is at the bottom.

On “username”, enter your gmail account that is “**ebotsidneysmith@gmail.com**”

Not secure 54.146.239.233:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... Summarize ⚡ ⚡ ⚡ ⚡ ⚡ ⚡ Chat

### New credentials

Kind: Username with password

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

Username: ebotsidneysmith@gmail.com

Treat username as secret

>Password: (Red arrow pointing here)

ID: (Red arrow pointing here)

Description: (Red arrow pointing here)

Create

Then, for the password, we will use the password we generated. That is “thcc ahus girv mvwj”

Not secure 54.146.239.233:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... Summarize ⚡ ⚡ ⚡ ⚡ ⚡ ⚡ Chat

### New credentials

Kind: Username with password

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

Username: ebotsidneysmith@gmail.com

Treat username as secret

Password: ..... (Red arrow pointing here)

ID: (Red arrow pointing here)

Description: (Red arrow pointing here)

Create

Then, for “ID” and “Description” enter “mail-cred”

Not secure 54.146.239.233:8080/manage/credentials/store/system/domain/\_/newCredentials

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted...)

### New credentials

Kind: Username with password

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

Username: ebotsidneysmith@gmail.com

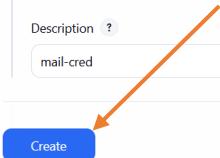
Treat username as secret

Password:

ID: mail-cred

Description: mail-cred

**Create**



Click on “Create”

Not secure 54.146.239.233:8080/manage/credentials/store/system/domain/\_/

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)

### Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ebotsidneysmith/******** (git-credential) git-credential - git-credential	
sonar-token sonar-token - sonar-token	
ebotsidneysmith/******** (docker-cred) docker-cred - docker-cred	
k8-cred k8-cred - k8-cred	
ebotsidneysmith@gmail.com/******** (mail-cred) mail-cred - mail-cred	



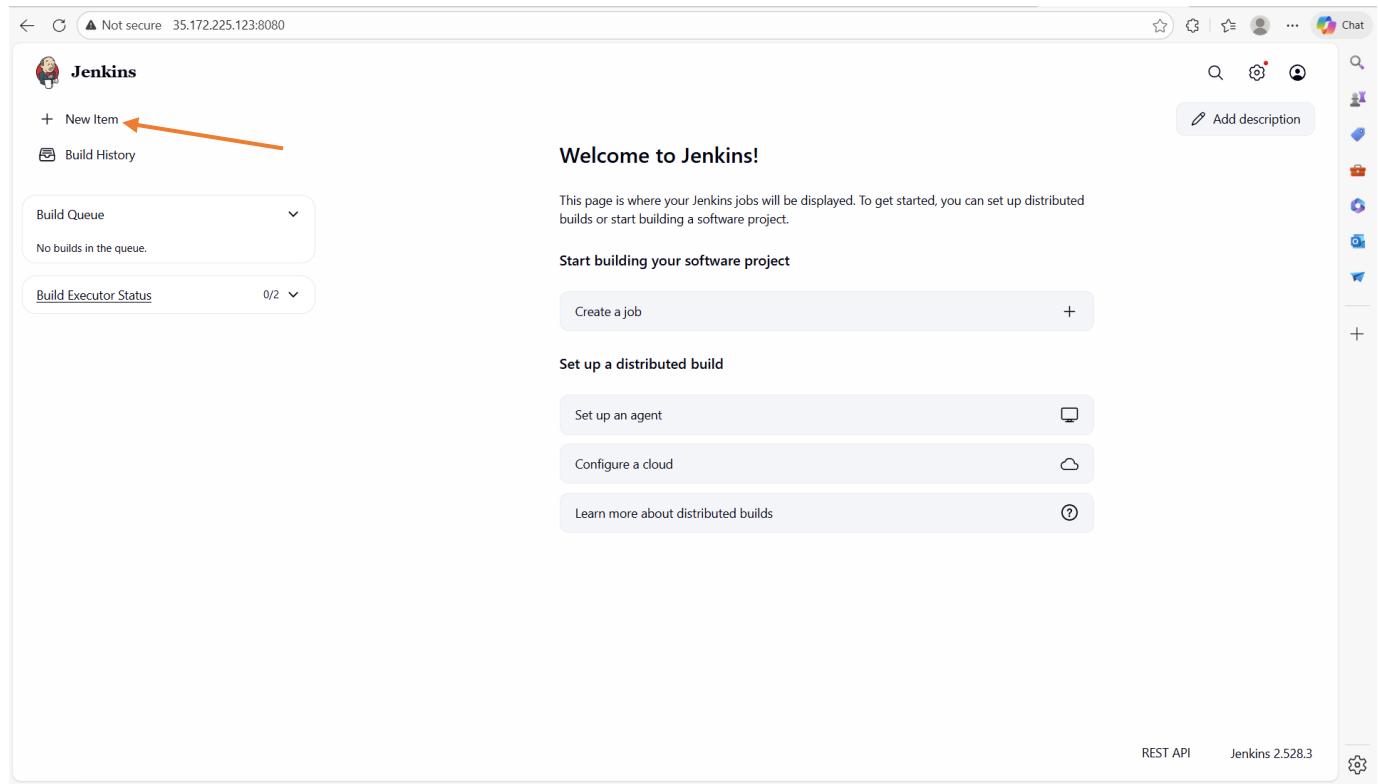
The credential for email notification has been created.

## 5.4 Create and configure the CI/CD Pipeline

In this part, we are going to create the CI/CD pipeline.

### 5.4.1 Create Pipeline

Let us create the pipeline. Go to Jenkins dashboard.



The screenshot shows the Jenkins dashboard. At the top left, there is a 'New Item' button with an orange arrow pointing to it. The main content area features a 'Welcome to Jenkins!' message and a 'Start building your software project' button. Below these are sections for 'Build Queue' (empty) and 'Build Executor Status' (0/2). A 'Create a job' button is located in the center. To the right, there's a sidebar with various Jenkins management links like 'Set up a distributed build', 'REST API', and 'Jenkins 2.528.3'. The bottom right corner has a page number '318'.

Click on “**New Item**”

Not secure 35.172.225.123:8080/view/all/newJob

Jenkins / All / New Item

### New Item

Enter an item name

» This field cannot be empty, please enter a valid 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.
- Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- 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.

OK

Let us give the Pipeline a name, we will call it “**BoardGame**”

Not secure 35.172.225.123:8080/view/all/newJob

Jenkins / All / New Item

### New Item

Enter an item name

BoardGame

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.
- Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- 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**

OK

Then, on “**Item Type**”, select “**Pipeline**”

New Item

Enter an item name

BoardGame

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.
- Maven project  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- 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

OK

Then, click on “OK”

Configure

General

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

Pipeline speed/durability override [?](#)

Preserve stashes from completed builds [?](#)

This project is parameterized [?](#)

Throttle builds [?](#)

Triggers

Save Apply

Check the box “Discard old builds”, we want to keep just two old builds to avoid occupying much space.

The screenshot shows the Jenkins 'General' configuration page for a job named 'BoardGame'. The 'Enabled' switch is turned on. In the 'Strategy' section, there is a dropdown menu set to 'Log Rotation'. Below it, there are two input fields: 'Days to keep builds' (empty) and 'Max # of builds to keep' (empty). An orange arrow points from the text 'Then, on "Max # of builds to keep", enter "2"' to the 'Max # of builds to keep' input field.

Then, on “Max # of builds to keep”, enter “2”

The screenshot shows the same Jenkins 'General' configuration page as before, but now the 'Max # of builds to keep' input field contains the value '2'. An orange arrow points from the text 'Then click on “Save”' to the blue 'Save' button at the bottom of the page.

Then click on “Save”

## 5.4.2 Configure the Pipeline

We will now start to configure the pipeline.

The screenshot shows the Jenkins configuration page for the 'BoardGame' pipeline. The left sidebar has 'Configure' selected, with sub-options: General, Triggers, Pipeline, and Advanced. The main panel is titled 'General' and contains the following settings:

- Description:** A large text area with a 'Plain text' link and a 'Preview' link.
- Discard old builds:** A checked checkbox with a help link. Below it is a 'Strategy' dropdown set to 'Log Rotation'.
  - Days to keep builds:** A dropdown menu showing 'if not empty, build records are only kept up to this number of days'.
  - Max # of builds to keep:** A dropdown menu showing 'if not empty, only up to this number of build records are kept' with the value '2'.
- Advanced:** A button to expand more options.

At the bottom are 'Save' and 'Apply' buttons. The top right shows the pipeline status as 'Enabled' with a blue toggle switch.

Go to the Jenkins dashboard by clicking on “BoardGame”

The screenshot shows the Jenkins dashboard for the 'BoardGame' pipeline. The left sidebar has 'Configure' selected, with other options like Status, Changes, Build Now, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The main panel is titled 'BoardGame' and contains:

- Stage View:** A box stating 'No data available. This Pipeline has not yet run.'
- Permalinks:** A section for sharing links.
- Builds:** A table showing 'No builds'.

The top right includes 'Summarize', 'Add description', and other dashboard controls. The bottom right shows 'REST API' and 'Jenkins 2.528.3'.

Click on “Configure”

Jenkins / BoardGame

Status

BoardGame

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Stage View

No data available. This Pipeline has not yet run.

Permalinks

Builds

No builds

Add description

REST API Jenkins 2.528.3

Click on “Configure”

Jenkins / BoardGame / Configuration

Configure

General

Enabled

Description

Plain text Preview

Discard old builds ?

Strategy

Log Rotation

Days to keep builds  
if not empty, build records are only kept up to this number of days

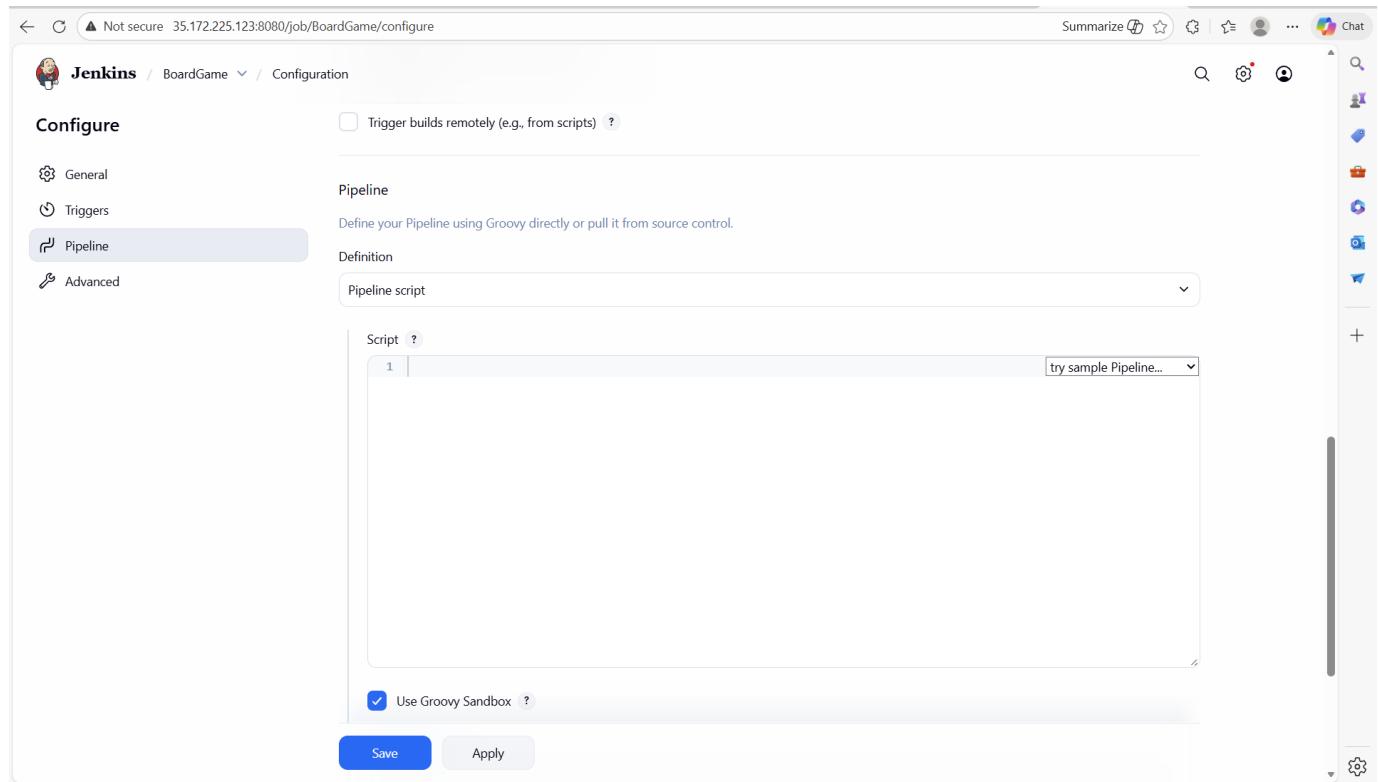
Max # of builds to keep  
if not empty, only up to this number of build records are kept

2

Advanced ▾

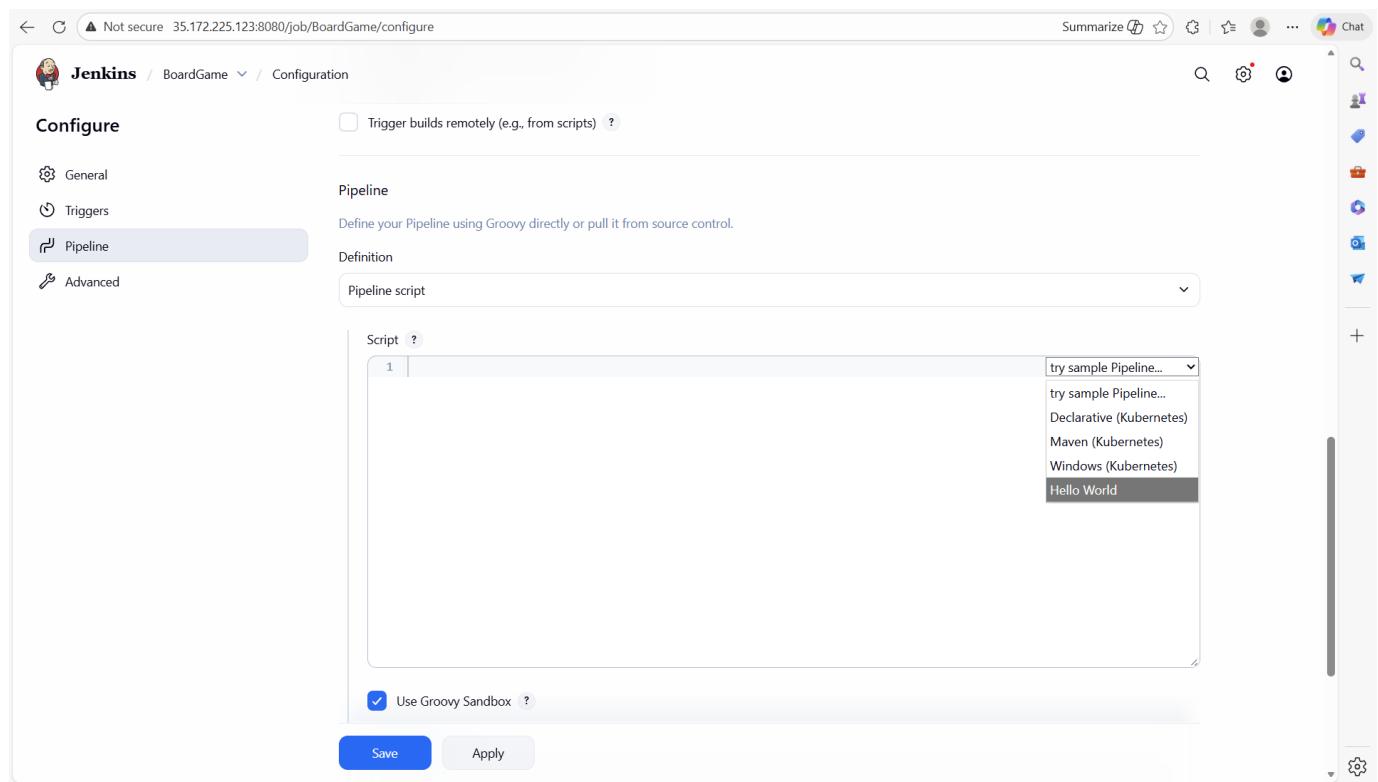
Save Apply

Scroll down to “Pipeline”



The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The left sidebar has "Pipeline" selected. The main area is titled "Pipeline" with the sub-instruction "Define your Pipeline using Groovy directly or pull it from source control." Below this is a "Definition" dropdown set to "Pipeline script". A large text area labeled "Script" contains the number "1". To the right of the script area is a dropdown menu with the option "try sample Pipeline..." highlighted. At the bottom are "Save" and "Apply" buttons.

Let us use a generated template for the Pipeline. Click on the drop down on “Try Sample Pipeline”



The screenshot shows the same Jenkins Pipeline configuration page as before, but with a different selection in the dropdown menu. The "try sample Pipeline..." option is now highlighted. Other options visible in the dropdown include "try sample Pipeline...", "Declarative (Kubernetes)", "Maven (Kubernetes)", "Windows (Kubernetes)", and "Hello World", which is also highlighted.

Select “Hello World”

The screenshot shows the Jenkins Pipeline configuration page for a job named 'BoardGame'. The 'Pipeline' tab is selected in the left sidebar. The main area contains a Groovy script for a pipeline:

```
1~ pipeline {  
2     agent any  
3  
4~     stages {  
5~         stage('Hello') {  
6~             steps {  
7                 echo 'Hello World'  
8             }  
9         }  
10    }  
11 }  
12 |
```

A dropdown menu next to the script editor is set to 'Hello World'. Below the script, there is a checked checkbox for 'Use Groovy Sandbox'. At the bottom are 'Save' and 'Apply' buttons.

Let us copy and create more stages that we will modify later to meet up with the requirements of our project.

The screenshot shows the Jenkins Pipeline configuration page for the same 'BoardGame' job. The 'Pipeline' tab is selected. The pipeline script now includes three stages:

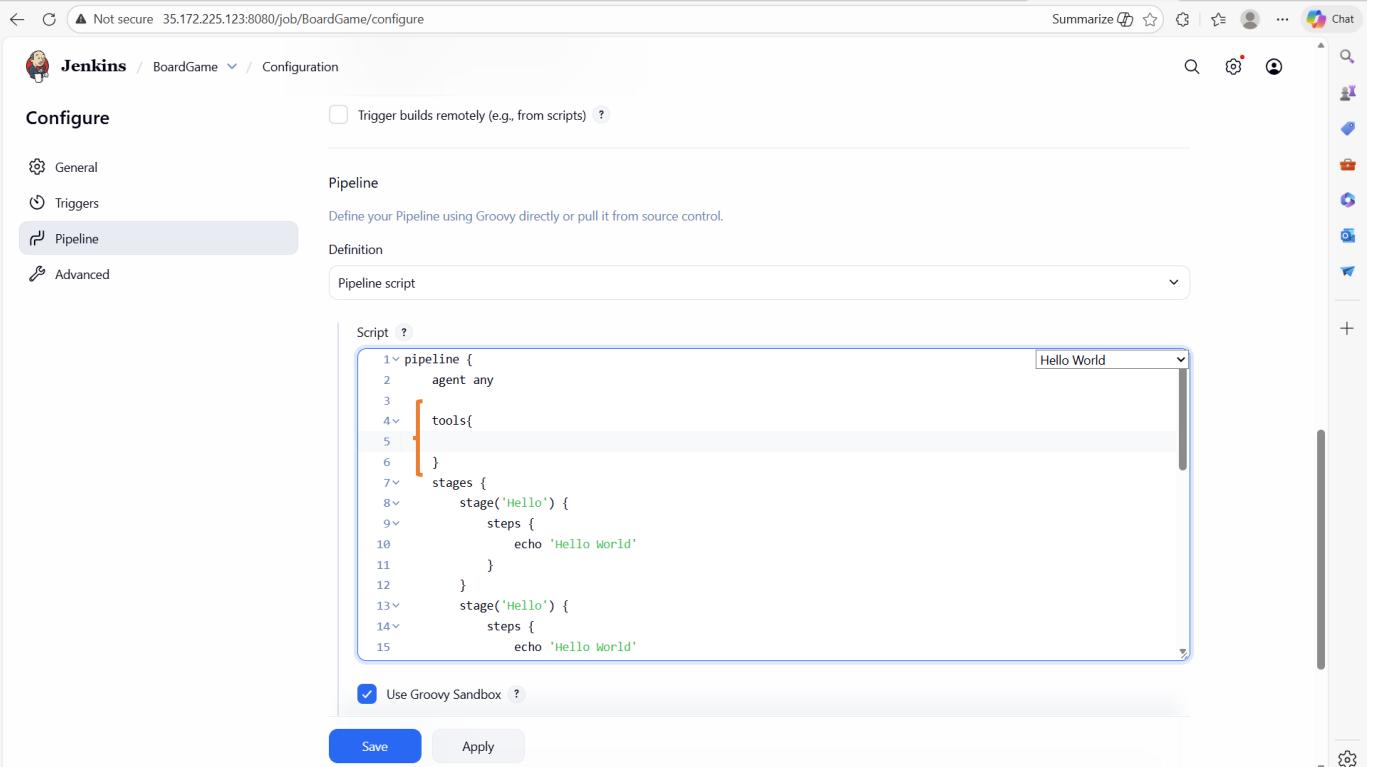
```
1~ pipeline {  
2     agent any  
3  
4~     stages {  
5~         stage('Hello') {  
6~             steps {  
7                 echo 'Hello World'  
8             }  
9         }  
10~        stage('Hello') {  
11~            steps {  
12                 echo 'Hello World'  
13             }  
14         }  
15~        stage('Hello') {  
16~            steps {  
17                 echo 'Hello World'  
18             }  
19         }  
20     }  
21 }
```

The dropdown menu next to the script editor is still set to 'Hello World'. Below the script, there is a checked checkbox for 'Use Groovy Sandbox'. At the bottom are 'Save' and 'Apply' buttons.

Let us start modifying the stages.

#### 5.4.2.1 Configure the JDK

We can now modify the first stage, but we have to first define the plugins/tools we have installed.

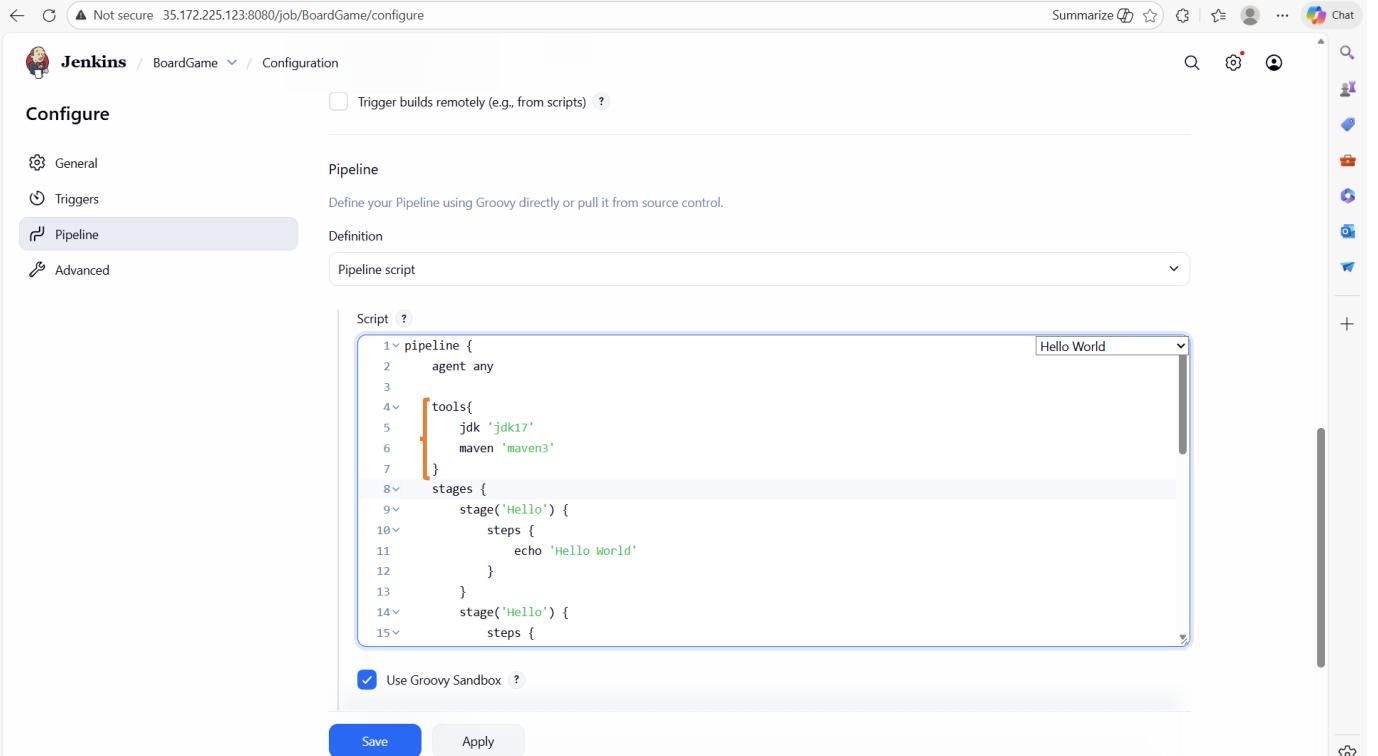


The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The "Pipeline" tab is selected. The pipeline script is defined as follows:

```
1~ pipeline {
2~     agent any
3~ 
4~     tools{
5~         jdk 'jdk17'
6~         maven 'maven3'
7~     }
8~     stages {
9~         stage('Hello') {
10~             steps {
11~                 echo 'Hello World'
12~             }
13~         }
14~         stage('Hello') {
15~             steps {
16~                 echo 'Hello World'
17~             }
18~         }
19~     }
20~ }
```

The "Use Groovy Sandbox" checkbox is checked. At the bottom are "Save" and "Apply" buttons.

Let us define Java JDK and Maven.



The screenshot shows the Jenkins Pipeline configuration page for the same job "BoardGame". The "Pipeline" tab is selected. The pipeline script has been updated to include tool definitions:

```
1~ pipeline {
2~     agent any
3~ 
4~     tools{
5~         jdk 'jdk17'
6~         maven 'maven3'
7~     }
8~     stages {
9~         stage('Hello') {
10~             steps {
11~                 echo 'Hello World'
12~             }
13~         }
14~         stage('Hello') {
15~             steps {
16~                 echo 'Hello World'
17~             }
18~         }
19~     }
20~ }
```

The "Use Groovy Sandbox" checkbox is checked. At the bottom are "Save" and "Apply" buttons.

We have defined the Maven and Java JDK tools. Now, let us modify our first stage.

The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area contains a code editor with a Groovy script:

```
4<v tools{
5     jdk 'jdk17'
6     maven 'maven3'
7 }
8<v stages {
9<v     stage('Git Checkout') {
10<v         steps {
11             echo 'Hello World' ← orange arrow
12         }
13     }
14<v     stage('Hello') {
15<v         steps {
16             echo 'Hello World'
17         }
18 }
```

A red arrow points from the text "Pipeline Syntax" in the sidebar to the "Pipeline Syntax" link in the main editor area. Below the editor are "Save" and "Apply" buttons.

Now, let us modify the “step”. We can get help by clicking on “Pipeline Syntax”

The screenshot shows the Jenkins Snippet Generator page under the "Pipeline Syntax" tab. The left sidebar lists various snippets: Snippet Generator, Declarative Directive Generator, Declarative Online Documentation, Steps Reference, Global Variables Reference, Online Documentation, Examples Reference, and IntelliJ IDEA GDSL. The main area is titled "Overview" and describes the Snippet Generator's purpose. Below is the "Steps" section:

Steps

Sample Step

archiveArtifacts: Archive the artifacts

archiveArtifacts ?

Files to archive ?

Advanced ▾

Generate Pipeline Script

Global Variables

A red arrow points from the text "Sample Step" to a dropdown menu icon next to it.

Click on the drop down on “Sample Step”

This screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, there's a sidebar with links like Snippet Generator, Declarative Directive Generator, and Steps Reference. The main area has a heading 'Overview' with a brief description of the Snippet Generator. Below it is a section titled 'Steps' with a 'Sample Step' dropdown menu. The 'git: Git' option is highlighted with a blue selection bar. A scrollable list of other steps is visible below, including checkout, cleanWs, configFileProvider, container, deleteDir, dir, dockerNode, echo, emailext, emalexrecipients, error, fileExists, findBuildScans, fingerprint, git, input, isUnix, and javadoc.

## Select “git:Git”

This screenshot shows the same Jenkins Pipeline Syntax Snippet Generator interface as the previous one, but with a red arrow pointing to the 'Repository URL' field in the 'git' configuration form. The 'git' step is selected in the dropdown. The configuration form includes fields for Repository URL (with a required error message), Branch (set to 'master'), Credentials (set to 'none'), and two checkboxes for 'Include in polling?' and 'Include in changelog?'.

For the repository URL, go to our GitHub repository on GitHub page

A screenshot of a GitHub repository page for 'Boardgame'. The repository is private and has 1 branch and 0 tags. The main commit was made by 'ebotsmith2000' 8 hours ago. The 'Code' dropdown menu is highlighted with an orange arrow. The dropdown shows options: Local, Codespaces, Clone, HTTPS, SSH, GitHub CLI, and links to Open with GitHub Desktop and Download ZIP. The URL `https://github.com/ebotsidneysmith/Boardgame.git` is visible in the 'Clone' field.

Click on the drop down on “Code”

A screenshot of the same GitHub repository page for 'Boardgame'. The 'Code' dropdown menu is now fully expanded, showing the 'Local' tab selected. The URL `https://github.com/ebotsidneysmith/Boardgame.git` is highlighted with an orange arrow. The other tabs in the dropdown are 'Codespaces', 'Clone', 'HTTPS', 'SSH', and 'GitHub CLI'. The rest of the repository page content is visible on the right.

Copy the repository URL

`https://github.com/ebotsidneysmith/Boardgame.git`

And paste in the Jenkins pipeline

The screenshot shows the Jenkins Pipeline Syntax configuration page. In the 'Sample Step' dropdown, 'git: Git' is selected. The 'Repository URL' field contains 'https://github.com/ebotsidneysmith/Boardgame.git'. A red error message below it states: 'Failed to connect to repository : Command "git ls-remote -h -- https://github.com/ebotsidneysmith/Boardgame.git HEAD" returned status code 128: stdout: stderr: remote: Invalid username or token. Password authentication is not supported for Git operations. fatal: Authentication failed for 'https://github.com/ebotsidneysmith/Boardgame.git/''.

The 'Branch' dropdown currently has 'master' selected. An orange arrow points from the text 'On "Branch", change the "master" to "main"' to this 'master' selection. Other configuration options shown include 'Credentials' set to '- none -', and two checked checkboxes: 'Include in polling?' and 'Include in changelog?'. A blue 'Generate Pipeline Script' button is at the bottom.

On “Branch”, change the “master” to “main”

This screenshot shows the same Jenkins Pipeline Syntax configuration page as the previous one, but with the 'Branch' dropdown now set to 'main'. The rest of the configuration remains the same, including the failed Git connection message and the checked polling and changelog checkboxes.

Then, click on the drop down on “Credentials” and select our GitHub credential

This screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, there's a sidebar with links like 'Declarative Directive Generator', 'Declarative Online Documentation', 'Steps Reference', etc. The main area has a heading 'This Snippet Generator will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click Generate Pipeline Script, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)'. Below this is a 'Steps' section with a 'Sample Step' dropdown set to 'git'. A search bar above the dropdown contains 'git'. The configuration form for 'git' includes fields for 'Repository URL' (set to 'https://github.com/ebotsidneysmith/Boardgame.git'), 'Branch' (set to 'main'), and 'Credentials' (set to 'ebotsidneysmith/\*\*\*\*\*\*\*\* (git-credential)'). There are two checked checkboxes: 'Include in polling?' and 'Include in changelog?'. An orange arrow points from the text 'Then, click on "Generate Pipeline Script"' to the blue 'Generate Pipeline Script' button at the bottom of the form.

Then, click on “Generate Pipeline Script”

This screenshot shows the same Jenkins Pipeline Syntax Snippet Generator interface after the 'Generate Pipeline Script' button was clicked. The configuration remains the same as in the previous screenshot. However, below the configuration form, a large blue box displays the generated Pipeline Script code: 'git branch: 'main', credentialsId: 'git-credential', url: 'https://github.com/ebotsidneysmith/Boardgame.git''. An orange arrow points from the text 'Copy this line of code:' to this generated code.

Copy this line of code:

```
git branch: 'main', credentialsId: 'git-credential', url: 'https://github.com/ebotsidneysmith/Boardgame.git'
```

And paste on the step of the first stage in the Pipeline

The complete stage is as follows:

```
stage('Git Checkout') {  
    steps {  
        git branch: 'main', credentialsId: 'git-credential', url:  
        'https://github.com/ebotsidneysmith/Boardgame.git'  
    }  
}
```

The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area contains a "Script" editor with the following Groovy code:

```
4~     tools{  
5~         jdk 'jdk17'  
6~         maven 'maven3'  
7~     }  
8~     stages {  
9~         stage('Git Checkout') {  
10~             steps {  
11~                 git branch: 'main', credentialsId: 'git-credential', url: 'https://github.com/ebotsidneysmith/Bo  
12~             }  
13~         }  
14~         stage('Hello') {  
15~             steps {  
16~                 echo 'Hello World'  
17~             }  
18~     }
```

A blue "Save" button is highlighted with an orange arrow pointing to it. Below the "Save" button is a "Pipeline Syntax" link.

The first stage is complete, we can then save it

Jenkins / BoardGame

Status

BoardGame

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

No data available. This Pipeline has not yet run.

Permalinks

Builds

No builds

REST API Jenkins 2.528.3

Now, we can proceed with the second stage.

#### 5.4.2.2 Configure Maven to Compile Source Code

In the second stage, we will compile the source code using Maven. So, we have to configure Maven. We will call the stage “**compile**”.

Jenkins / BoardGame

Status

BoardGame

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

No data available. This Pipeline has not yet run.

Permalinks

Builds

No builds

REST API Jenkins 2.528.3

Click on “Configure” and scroll down to “Pipeline”

The screenshot shows the Jenkins configuration interface for a job named "BoardGame". The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area is titled "Definition" with a dropdown set to "Pipeline script". A code editor displays the following Groovy pipeline script:

```
1~ pipeline {
2~     agent any
3~ 
4~     tools{
5~         jdk 'jdk17'
6~         maven "maven3"
7~     }
8~     stages {
9~         stage('Git checkout') {
10~             steps {
11~                 git branch: 'main', credentialsId: 'git-credential', url: 'https://github.com/ebotsidneysmith/BoardGame.git'
12~             }
13~         }
14~         stage('Hello') {
15~             steps {
16~                 echo 'Hello World'
17~             }
18~         }
19~         stage('Hello') {
20~             steps {
21~                 echo 'Hello World'
22~             }
23~         }
24~         stage('Hello') {
25~             steps {
26~                 echo 'Hello World'
27~             }
28~         }
29~     }
30~ }
```

Below the code editor is a checkbox for "Use Groovy Sandbox". At the bottom are "Save" and "Apply" buttons.

Let us modify the second stage now, we will call it “Compile”

The screenshot shows the same Jenkins configuration interface. The "Pipeline" tab is still selected. The pipeline script now includes a new stage named "Compile":

```
13~         }
14~         stage('compile') {
15~             steps {
16~                 echo 'Hello World' | ←
17~             }
18~         }
19~         stage('Hello') {
20~             steps {
21~                 echo 'Hello World'
22~             }
23~         }
24~         stage('Hello') {
25~             steps {
26~                 echo 'Hello World'
27~             }
28~         }
29~     }
30~ }
```

An orange arrow points to the "echo" command in the "compile" stage. Below the code editor is a checkbox for "Use Groovy Sandbox". At the bottom are "Save" and "Apply" buttons.

Let us modify the “step” in this stage. The Maven command is a shell command, so we have to include the keyword “sh”. So, we will enter the command:

```
sh "mvn compile"
```

The complete stage will be as follows:

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

The screenshot shows the Jenkins Pipeline configuration interface for a job named 'BoardGame'. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area is titled 'Configure' and contains a 'Definition' section with a 'Pipeline script' dropdown set to 'Script'. Below it is a code editor window with the following Groovy pipeline script:

```
13  
14 }  
15 stage('compile') {  
16     steps {  
17         sh "mvn compile"  
18     }  
19 }  
20 stage('Hello') {  
21     steps {  
22         echo 'Hello World'  
23     }  
24 }  
25 stage('Hello') {  
26     steps {  
27         echo 'Hello World'  
28     }  
29 }
```

Below the code editor is a checkbox labeled 'Use Groovy Sandbox' which is checked. At the bottom of the configuration page are two buttons: 'Save' (highlighted with a red arrow) and 'Apply'.

We have completed this stage. We can now save it. Click on “Save”

Jenkins / BoardGame

Status

BoardGame

</> Changes

▷ Build Now

⚙ Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Stage View

No data available. This Pipeline has not yet run.

Permalinks

Builds

No builds

REST API Jenkins 2.528.3

We can now proceed with the next stage.

#### 5.4.2.3 Configure Maven to Test Source Code

In this stage, we will execute test cases on the source code using Maven. So, we have to configure Maven. We will call the stage “**Test**”.

Jenkins / BoardGame

Status

BoardGame

</> Changes

▷ Build Now

⚙ Configure

Delete Pipeline

Full Stage View

Stages

Rename

Pipeline Syntax

Stage View

No data available. This Pipeline has not yet run.

Permalinks

Builds

No builds

REST API Jenkins 2.528.3

Click on “Configure” and scroll down to “Pipeline”

The screenshot shows the Jenkins configuration interface for a pipeline job named "BoardGame". The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area is titled "Configure" and contains a "Definition" section with a "Pipeline script" dropdown set to "Pipeline script". Below it is a code editor with the following Groovy script:

```
17      }
18    }
19    stage('Hello') {
20      steps {
21        echo 'Hello World'
22      }
23    }
24    stage('Hello') {
25      steps {
26        echo 'Hello World'
27      }
28    }
29    stage('Hello') {
30      steps {
31
```

Below the code editor is a checkbox for "Use Groovy Sandbox" which is checked. At the bottom are "Save" and "Apply" buttons.

We will now modify the third stage, let us first change the name to ‘Test’

The screenshot shows the same Jenkins configuration interface as before, but with a red arrow pointing to the word "Test" in the Groovy script. The script has been modified to:

```
17      }
18    }
19    stage('Test') {
20      steps {
21        echo 'Hello World'
22      }
23    }
24    stage('Hello') {
25      steps {
26        echo 'Hello World'
27      }
28    }
29    stage('Hello') {
30      steps {
31
```

The rest of the interface remains the same with the "Use Groovy Sandbox" checkbox checked and "Save" and "Apply" buttons at the bottom.

Let us modify the “Step” in the third stage now. We will still use shell and the command will be:

```
sh "mvn test"
```

The complete stage will be as follows:

```
stage('Test') {  
    steps {  
        sh "mvn test"  
    }  
}
```

The screenshot shows the Jenkins Pipeline Configuration screen for a job named 'BoardGame'. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area is titled 'Definition' with a dropdown set to 'Pipeline script'. A large text area contains the Groovy pipeline script. The script defines three stages: 'Test', 'Hello', and 'Hello', each with a single step of 'sh "mvn test"' or 'echo 'Hello World''. Below the script is a checkbox 'Use Groovy Sandbox' which is checked. At the bottom are 'Save' and 'Apply' buttons.

```
18      }  
19  stage('Test') {  
20      steps {  
21          sh "mvn test"  
22      }  
23  }  
24  stage('Hello') {  
25      steps {  
26          echo 'Hello World'  
27      }  
28  }  
29  stage('Hello') {  
30      steps {  
31          echo 'Hello World'  
32      }
```

Save it by clicking on “Save”

#### 5.4.2.3 Configure Trivy to Scan the file System

In this stage, we will scan the file system for vulnerabilities that may exist in the dependencies we are using with Trivy. The dependencies are defined in the pom.xml file. We also want to find if there is any kind of sensitive data that is being stored in the repository. For that, we will use a third-party tool called Trivy. So, we have to configure Trivy.

We will call the stage “File System Scan”.

The screenshot shows the Jenkins Pipeline configuration interface. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area is titled 'Configure' and contains a 'Definition' section set to 'Pipeline script'. The script editor shows the following Groovy code:

```

18    }
19    stage('Test') {
20        steps {
21            sh "mvn test"
22        }
23    }
24    stage('File System Scan') {
25        steps {
26            echo 'Hello World' ← Orange arrow points here
27        }
28    }
29    stage('Hello') {
30        steps {
31            echo 'Hello World'
32        }
33    }

```

Below the script, there is a checkbox for 'Use Groovy Sandbox' which is checked. At the bottom are 'Save' and 'Apply' buttons.

Let us now modify the “Step” in this stage.

Trivy is going to generate a report. If we use the command:

```
sh "trivy fs ."
```

This is going to generate the report inside the console log which is hard to analyze. The best way is to ask Trivy to generate the report in a specific format and export it inside a separate file. So, we will use the command:

```
sh "trivy fs -format table -o trivy-fs-report.html ."
```

The complete stage is as follows:

```
stage('File System Scan') {
    steps {
        sh "trivy fs -format table -o trivy-fs-report.html ."
    }
}
```

The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The "Pipeline" tab is selected in the left sidebar. The main area contains a Groovy script editor with the following code:

```
18    }
19    stage('Test') {
20        steps {
21            sh "mvn test"
22        }
23    }
24    stage('File System Scan') {
25        steps {
26            sh "trivy fs -format table -o trivy-fs-report.html"
27        }
28    }
29    stage('Hello') {
30        steps {
31            echo 'Hello World'
32        }
33    }

```

Below the script, there is a checkbox labeled "Use Groovy Sandbox" which is checked. At the bottom of the editor, there are two buttons: "Save" (highlighted with a red arrow) and "Apply".

Then click on “Save”.

The screenshot shows the Jenkins Pipeline status page for the "BoardGame" pipeline. The left sidebar includes options like "Status", "Changes", "Build Now", "Configure", "Delete Pipeline", "Full Stage View", "Stages", "Rename", and "Pipeline Syntax". The main area is titled "Stage View" and displays the message "No data available. This Pipeline has not yet run". Below this is a "Permalinks" section. At the bottom of the page, there is a "Builds" section with the message "No builds". At the very bottom right, there are links for "REST API" and "Jenkins 2.528.3".

#### 5.4.2.4 Configure SonarQube to Scan

In this stage, we will scan the code for analysis using SonarQube. We will call this stage “**SonarQube Analysis**”

The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The left sidebar has "Pipeline" selected under "Configure". The main area contains a Groovy script:

```
--> sh "trivy fs --format table -o trivy-fs-report.html"
26
27 }
28 }
29 <-- stage('SonarQube Analysis') {
30   steps {
31     echo 'Hello World'
32   }
33 }
34 <-- stage('Hello') {
35   steps {
36     echo 'Hello World'
37   }
38 }
```

A checkbox labeled "Use Groovy Sandbox" is checked. Below the script, a blue link "Pipeline Syntax" is highlighted with an orange arrow.

At the bottom, there are "Save" and "Apply" buttons. The status bar at the bottom right shows "REST API" and "Jenkins 2.528.3".

Then, we have to modify the “step” in this stage. Click on “**Pipeline Syntax**”

The screenshot shows the "Pipeline Syntax" Snippet Generator page. The left sidebar has "Snippet Generator" selected. The main area has "Steps" selected under "Sample Step". A sample step "archiveArtifacts: Archive the artifacts" is shown with a dropdown menu icon highlighted with an orange arrow.

Below the sample step, there is a "Generate Pipeline Script" button and a large text area for the generated script.

At the bottom, there is a "Global Variables" section.

Click on the drop down on “Sample Step” and select “withSonarQubeEnv”

The screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, there's a sidebar with links like Snippet Generator, Declarative Directive Generator, etc. The main area has a title 'Overview' and a section 'Steps'. A dropdown menu under 'Sample Step' is open, showing 'withSonarQubeEnv: Prepare SonarQube Scanner environment'. Below it, there's a field for 'Server authentication token' with a dropdown containing '- none -'. To the right of this field is a button with a minus sign and a plus sign labeled '+ Add'. An orange arrow points from the text 'Select "withSonarQubeEnv"' to the '+ Add' button. At the bottom, there's a 'Generate Pipeline Script' button and a large empty text area for the generated script.

Click on the drop down on “Server Authentication Token” and select our SonarQube token.

This screenshot is similar to the previous one but shows the 'Server authentication token' dropdown now containing 'sonar-token'. Below the dropdown, a red warning message reads '⚠ Cannot find any credentials with id sonar-token'. The rest of the interface is identical to the first screenshot, including the sidebar, 'Overview' title, and 'Generate Pipeline Script' button.

Click on “Generate Pipeline Script” to generate the block

This Snippet Generator will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click Generate Pipeline Script, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

**Steps**

Sample Step

withSonarQubeEnv: Prepare SonarQube Scanner environment

withSonarQubeEnv ?

Server authentication token  
SonarQube authentication token. Mandatory when anonymous access is disabled. Will default to the one defined in the SonarQube installation.

sonar-token

⚠ Cannot find any credentials with id sonar-token

**Generate Pipeline Script**

```
withSonarQubeEnv(credentialsId: 'sonar-token') {  
    // some block  
}
```

Copy the block and paste on the step on the “SonarQube Analysis” stage in our Pipeline

Configure

General

Triggers

Pipeline

Advanced

```
--  
26      sh "trivy fs --format table -o trivy-fs-report.html."  
27  }  
28 }  
29 stage('SonarQube Analysis') {  
30   steps {  
31     withSonarQubeEnv(credentialsId: 'sonar-token') {  
32       // some block  
33     }  
34   }  
35 }  
36 stage('Hello') {  
37   steps {  
38     echo 'Hello World'  
39 }
```

Use Groovy Sandbox ?

**Pipeline Syntax**

**Advanced**

Advanced ▾

**Save** **Apply**

But we still have to modify the “step”. We are not going to use the credential. We are just going to use “sonar”, since we have already configured the URL and password to “sonar”

The screenshot shows the Jenkins Pipeline Configuration page for a job named 'BoardGame'. The left sidebar has 'Pipeline' selected. The main area contains a Groovy script:

```

--> sh "trivy fs -format table -o trivy-fs-report.html"
27 }
28 }
29< stage('SonarQube Analysis') {
30<   steps {
31<     withSonarQubeEnv('sonar') {
32       // some block
33     }
34   }
35 }
36< stage('Hello') {
37<   steps {
38     echo 'Hello World'
39

```

A checkbox 'Use Groovy Sandbox' is checked. Below the script, there's an 'Advanced' section with a dropdown set to 'Advanced'. At the bottom are 'Save' and 'Apply' buttons.

Lastly, we have to define the environment as follows:

`SCANNER_HOME=tool 'sonar-scanner'`

The screenshot shows the Jenkins Pipeline Configuration page for the same 'BoardGame' job. The 'Pipeline' section is selected in the sidebar. The Groovy script now includes the environment variable definition:

```

1< pipeline {
2   agent any
3
4   tools{
5     jdk 'jdk17'
6     maven 'maven3'
7   }
8   environment{
9     SCANNER_HOME=tool 'sonar-scanner' ← orange arrow points here
10  }
11  stages {
12    stage('Git Checkout') {
13      steps {
14        git branch: 'main', credentialsId: 'git-credential', url: 'https://github.com/ebotsidneysmith/Bo
15

```

An orange arrow points to the line 'SCANNER\_HOME=tool 'sonar-scanner''. Below the script, the 'Use Groovy Sandbox' checkbox is checked. The 'Advanced' section is expanded, and at the bottom are 'Save' and 'Apply' buttons.

Let us head back to the “step” in the “SonarQube Analysis” stage and complete the modification. We have to write the command to execute the analysis.

```
sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=BoardGame -  
Dsonar.projectKey=BoardGame \  
          -Dsonar.java.binaries=. '''
```

The complete stage will be as follows:

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

The screenshot shows the Jenkins Pipeline configuration interface. The pipeline script is defined as follows:

```
27< stage('File System Scan') {  
28<     steps {  
29<         sh "trivy fs --format table -o trivy-fs-report.html."  
30<     }  
31< }  
32< stage('SonarQube Analysis') {  
33<     steps {  
34<         withSonarQubeEnv('sonar') {  
35<             sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=BoardGame -Dsonar.projectKey=BoardGame \  
-Dsonar.java.binaries=. '''  
36<         }  
37<     }  
38< }  
39< stage('Hello') {  
40<     steps {  
41<         echo "Hello World!"  
42<     }  
43< }
```

The 'Pipeline' tab is highlighted in the sidebar. A 'Use Groovy Sandbox' checkbox is checked. At the bottom, there are 'Save' and 'Apply' buttons.

Click on “**Apply**” followed by “**Save**”

The screenshot shows the Jenkins Pipeline configuration for a job named 'BoardGame'. On the left, there's a sidebar with links like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, and Pipeline Syntax. The main area has tabs for 'Stage View' and 'Permalinks'. Under 'Stage View', it says 'No data available. This Pipeline has not yet run.' Under 'Permalinks', there's a section for 'Builds' which says 'No builds'. At the bottom right, it shows 'REST API' and 'Jenkins 2.528.3'.

#### 5.4.2.4 Configure SonarQube to perform Quality Gate

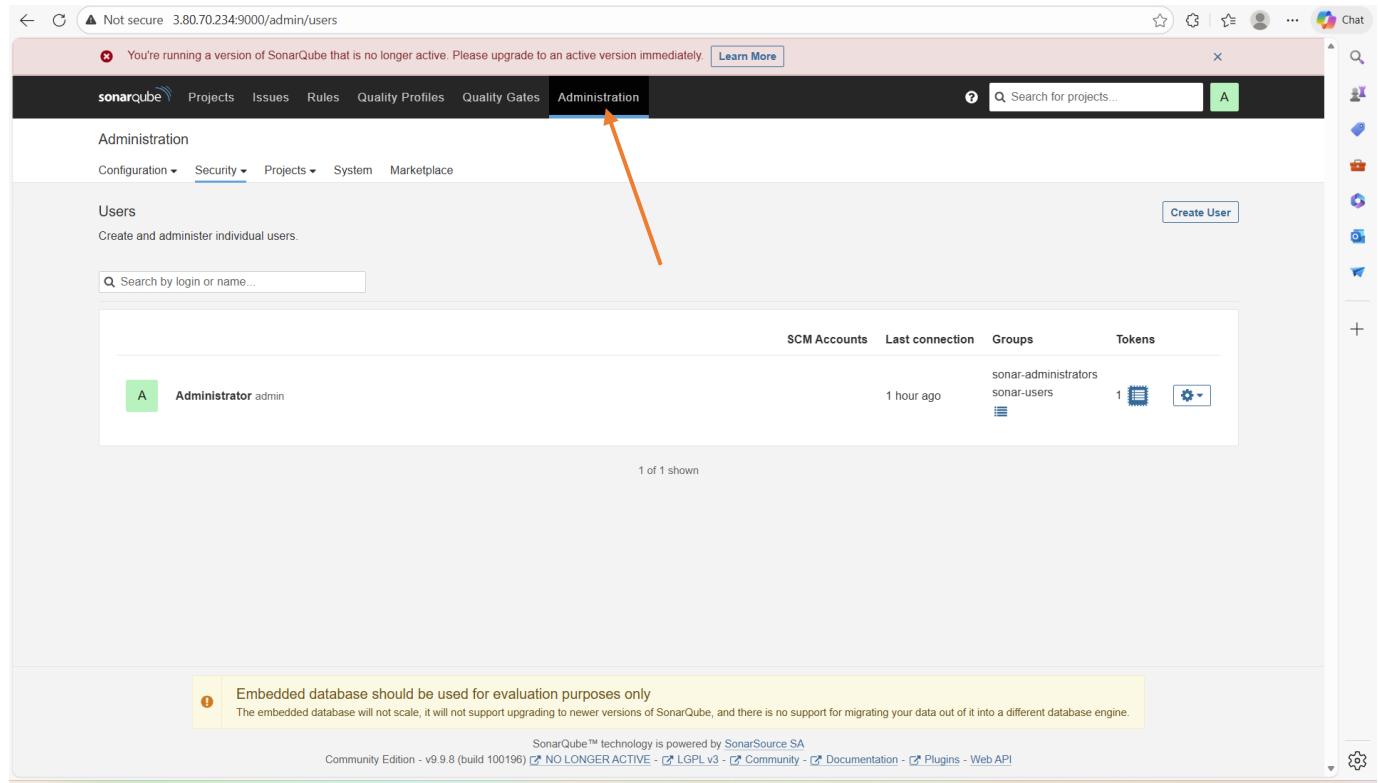
In this stage, we will use SonarQube to perform Quality Gates. Quality Gates are conditions. We will call this stage “**Quality Gate**”. If the conditions are passed, then the code is fine, we can say it has good quality.

The screenshot shows the Jenkins Pipeline configuration for 'BoardGame'. The 'Pipeline' tab is selected in the sidebar. The main area contains a Groovy script editor with the following code:

```
34      withSonarQubeEnv('sonar') {  
35          sh '' '$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=BoardGame -Dsonar.projectKey=Boar  
36              -Dsonar.java.binaries=...'   
37      }  
38  }  
39 }  
40 stage('Quality Gate') {  
41     steps {  
42         echo 'Hello World'  
43     }  
44 }  
45 }  
46 }  
47 }
```

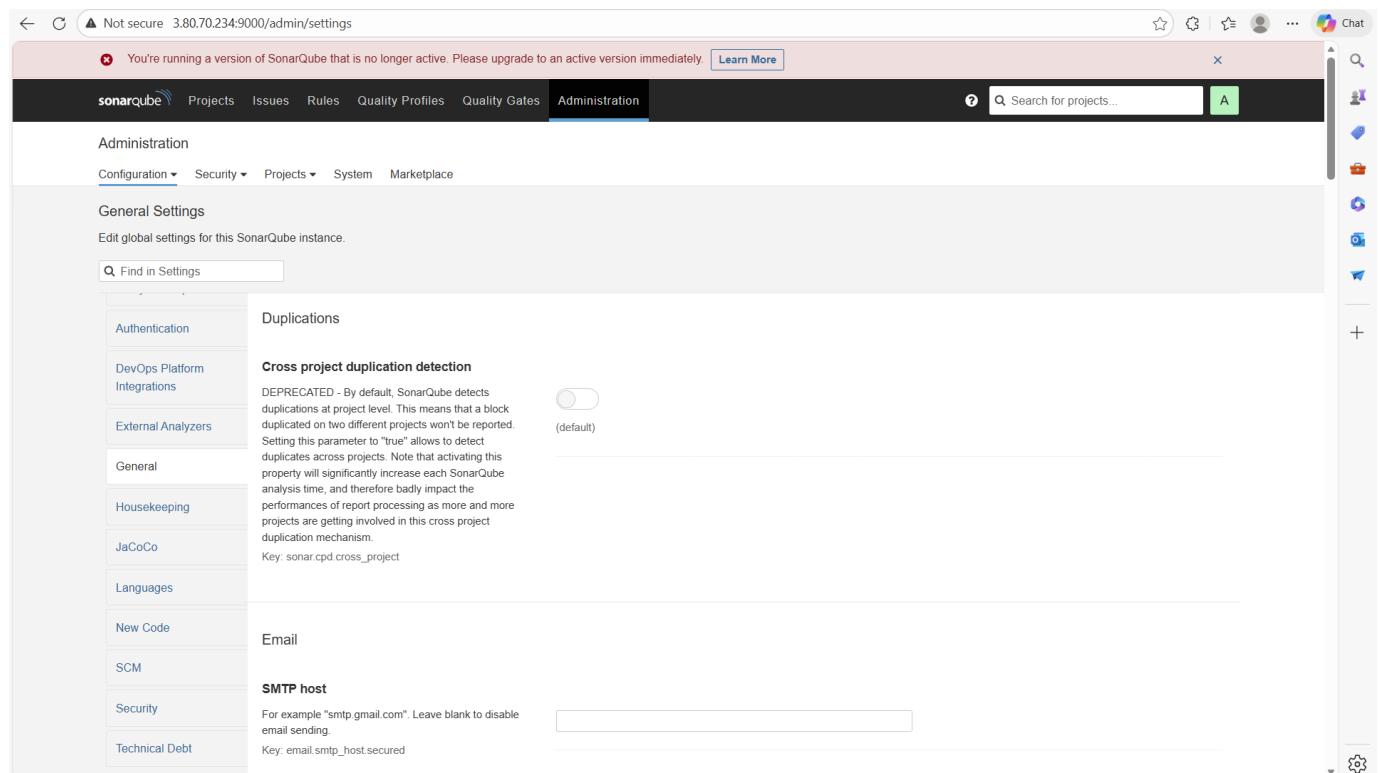
Below the script, there's a checkbox for 'Use Groovy Sandbox' which is checked. At the bottom, there are 'Save' and 'Apply' buttons, and a link to 'Pipeline Syntax'.

Let us modify the “**step**” in this stage now. To write the quality gate, let us go back to the SonarQube browser.



The screenshot shows the SonarQube Administration interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A red arrow points to the 'Administration' link. Below the navigation, there's a sub-navigation with Configuration, Security, Projects, System, and Marketplace. The main content area is titled 'Administration' and 'Users'. It displays a table with one user entry: 'Administrator admin'. The table includes columns for SCM Accounts, Last connection, Groups, and Tokens. A note at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.' The footer provides version information: 'Community Edition - v9.9.8 (build 100196) NO LONGER ACTIVE - LGPL v3 - Community - Documentation - Plugins - Web API'.

Click on “Administration”



The screenshot shows the SonarQube Administration interface with 'Configuration' selected in the sub-navigation. The left sidebar lists various configuration categories: Authentication, DevOps Platform Integrations, External Analyzers, General, Housekeeping, JaCoCo, Languages, New Code, SCM, Security, and Technical Debt. The main content area is titled 'General Settings' and describes how to edit global settings for the SonarQube instance. It includes sections for 'Duplications' (with 'Cross project duplication detection' turned off), 'Email' (with an input field for 'SMTP host'), and 'New Code' (with a note about the 'sonar.cpdl.cross\_project' key). The footer is identical to the previous screenshot, showing the same version and legal information.

Then, click on the drop down on “Configuration”

The screenshot shows the SonarQube Administration interface. The left sidebar has a 'Webhooks' item highlighted with an orange arrow. The main content area displays the 'Webhooks' configuration page, which includes sections for 'Cross project duplication detection' (disabled by default) and 'SMTP host' (set to blank). A search bar at the top right says 'Search for projects...'. The URL in the address bar is 3.80.70.234:9000/admin/webhooks.

## Select “Webhooks”

The screenshot shows the SonarQube Webhooks configuration page. It features a 'Create' button highlighted with an orange arrow. Below it, a message states 'No webhook defined.' A warning box at the bottom left says 'Embedded database should be used for evaluation purposes only'. The URL in the address bar is 3.80.70.234:9000/admin/webhooks.

Let us create a webhooks, click on create

**Create Webhook**

All fields marked with \* are required

**Name \***

**URL \***

Server endpoint that will receive the webhook payload, for example:  
"http://my\_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example:  
"https://myLogin:myPassword@my\_server/foo"

**Secret**

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header.

[Create](#) [Cancel](#)

Give the webhooks a name, we will call it “**Jenkins**”

**Create Webhook**

All fields marked with \* are required

**Name \***  
 ✓

**URL \***

Server endpoint that will receive the webhook payload, for example:  
"http://my\_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example:  
"https://myLogin:myPassword@my\_server/foo"

**Secret**

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header.

[Create](#) [Cancel](#)

For the URL, it will be <http://<Jenkins Server's Public IP>:8080/sonarqube-webhook/>

That is <http://54.226.220.79:8080/sonarqube-webhook/>

**Create Webhook**

All fields marked with \* are required

**Name \***  
jenkins

**URL \***  
http://35.172.225.123:8080/sonarqube-webhook/

Server endpoint that will receive the webhook payload, for example: "http://my\_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example: "https://myLogin.myPassword@my\_server/foo"

**Secret**

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header.

**Create** **Cancel**

Click on “Create”

We will use script.

```
script {
    waitForQualityGate abortPipeline: false, credentialsId: 'sonar-token'
}
```

And the complete stage will be as follows:

```
stage('Quality Gate') {
    steps {
        script {
            waitForQualityGate abortPipeline: false, credentialsId: 'sonar-token'
        }
    }
}
```

The screenshot shows the Jenkins Pipeline Configuration page for a job named "BoardGame". The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area contains Groovy pipeline code:

```

34      withSonarQubeEnv('sonar') {
35          sh '$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=BoardGame -Dsonar.projectKey=BoardGame -Dsonar.java.binaries='
36      }
37  }
38 }
39 }
40 stage('Quality Gate') {
41     steps {
42         script {
43             waitForQualityGate abortPipeline: false, credentialsId: 'sonar-token'
44         }
45     }
46 }
47 }
48

```

Below the code is a checkbox for "Use Groovy Sandbox". At the bottom are "Save" and "Apply" buttons. An orange arrow points from the "Apply" button to the "Save" button.

Click on “**Apply**” followed by “**Save**”. Then click on “**Build Now**” let us try to build the stages we have completed so far.

The screenshot shows the Jenkins Job page for "BoardGame". The left sidebar includes links for Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, SonarQube, Stages, Rename, and Pipeline Syntax. The main area displays the "Stage View" with a table showing average stage times:

	Declarative: Tool Install	Git Checkout	compile	Test	File System Scan	SonarQube Analysis	Quality Gate
Average stage times: (full run time: ~37s)	186ms	577ms	3s	15s	1s	13s	425ms
#11 Jan 05 06:32 No Changes	186ms	577ms	3s	15s	1s	13s	425ms (paused for 2s)

Below the Stage View is the "SonarQube Quality Gate" section, which shows "BoardGame" as Passed. The "Permalinks" section lists the last four builds as successful. At the bottom right are "REST API" and "Jenkins 2.528.3" links.

The build is successful.

#### 5.4.2.5 Configure Maven to Build Source Code

In this stage, we will use Maven to Build or package the source code and its dependencies. We will call this stage “Build”.

The screenshot shows the Jenkins Pipeline interface for the 'BoardGame' pipeline. On the left, there's a sidebar with options like 'Status', 'Changes', 'Build Now', 'Configure' (which has an orange arrow pointing to it), 'Delete Pipeline', 'Full Stage View', 'SonarQube', 'Stages', 'Rename', and 'Pipeline Syntax'. The main area is titled 'Stage View' and shows a summary of stage times: Declarative: Tool Install (186ms), Git Checkout (577ms), compile (3s), Test (15s), File System Scan (1s), SonarQube Analysis (13s), and Quality Gate (425ms). Below this is a detailed timeline for build #11, showing each stage's duration. To the right, there's a 'SonarQube Quality Gate' section indicating a 'Passed' status with a green button labeled 'Success'. At the bottom, there are links for 'Builds' (Today: #11 11:32 AM) and 'Permalinks' (listing the last four builds). The footer includes links for 'REST API' and 'Jenkins 2.528.3'.

Click on “Configure” and scroll down to “Pipeline”

The screenshot shows the 'Configuration' screen for the 'BoardGame' pipeline. The left sidebar has tabs for 'General', 'Triggers', 'Pipeline' (which is selected and highlighted in blue), and 'Advanced'. The main area starts with a 'Trigger builds remotely' checkbox. Below that is the 'Pipeline' section, which says 'Define your Pipeline using Groovy directly or pull it from source control.' A 'Definition' dropdown is set to 'Pipeline script'. The script editor contains the following Groovy code:

```
1~ pipeline {
2    agent any
3
4~   tools{
5      jdk 'jdk17'
6      maven 'maven3'
7    }
8~   environment{
9      SCANNER_HOME=tool 'sonar-scanner'
10 }
11~   stages {
12~     stage('Git Checkout') {
13~       steps {
14         git branch: 'main', credentialsId: 'git-credential', url: 'https://github.com/ebotsidneysmith/BoardGame.git'
15~     }
16~   }
17~ }
```

At the bottom of the script editor, there's a 'Use Groovy Sandbox' checkbox. At the very bottom are 'Save' and 'Apply' buttons.

Then add the new stage called “Build” and in the step, we will just run the shell command:

```
sh "mvn package"
```

The complete stage is as follows:

```
stage('Build') {  
    steps {  
        sh "mvn package"  
    }  
}
```

The screenshot shows the Jenkins Pipeline configuration interface. On the left, there's a sidebar with tabs: General, Triggers, Pipeline (which is selected and highlighted in grey), and Advanced. The main area is titled 'Configure' and has a sub-section 'Definition' set to 'Pipeline script'. A code editor displays Groovy pipeline script. An orange arrow points from the text 'We have added the “Build” stage. We can now save the file. Click on “Apply” followed by “Save”' to the 'Apply' button at the bottom of the editor. The script content is as follows:

```
40 ~   stage('Quality Gate') {  
41 ~     steps {  
42 ~       script {  
43 ~         waitForQualityGate abortPipeline: false, credentialsId: 'sonar-token'  
44 ~       }  
45 ~     }  
46 ~   }  
47 ~   stage('Build') {  
48 ~     steps {  
49 ~       sh "mvn package"  
50 ~     }  
51 ~   }  
52 ~ }  
53 }  
54 }
```

At the bottom of the editor, there are two buttons: 'Save' and 'Apply'. An orange arrow points to the 'Apply' button.

We have added the “Build” stage. We can now save the file. Click on “Apply” followed by “Save”

**Stage View**

Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build
Average stage times: (full run time: ~49s)	203ms	572ms	3s	15s	1s	13s	443ms
#6 Jan 11 16:36 No Changes	212ms	562ms	3s	15s	1s	13s	432ms (paused for 1s)
#5 Jan 11 16:33 No Changes	195ms	583ms	3s	15s	1s	14s	454ms (paused for 2s)

**SonarQube Quality Gate**

BoardGame Passed  
server-side processing: Success

**Permalinks**

- Last build (#5), 2 min 47 sec ago
- Last stable build (#5), 2 min 47 sec ago
- Last successful build (#5), 2 min 47 sec ago
- Last completed build (#5), 2 min 47 sec ago

## 5.4.2.5 Configure Nexus to Publish Artifacts

In this stage, we will use Nexus to publish the artifacts. We will call this stage “**Publish to Nexus**”.

**Configure**

- General
- Triggers
- Pipeline**
- Advanced

```

47~    stage('Build') {
48~      steps {
49~        sh "mvn package"
50~      }
51~    }
52~    stage('Publish to Nexus') {
53~      steps {
54~        sh "mvn package" ← Orange arrow points here
55~      }
56~    }
57~  }
58~}
59~
```

Use Groovy Sandbox ?

**Advanced**

Advanced ▾

**Save** **Apply**

REST API Jenkins 2.528.3

We have to now modify the step. In order to publish the artifacts to Nexus, we need to add the repository URL in our pom.xml file. But we have to first add Maven releases to our pom.xml file. To get the Maven releases, go to Nexus browser

The screenshot shows the Sonatype Nexus Repository Manager interface. On the left, there's a sidebar with 'Dashboard', 'Search', and 'Browse' buttons. The 'Browse' button is highlighted with a green background. The main area is titled 'Browse' and contains a table with columns: Name, Type, Format, Status, URL, and Health check. The rows listed are: maven-central (proxy, maven2, Online - Ready to Connect), maven-public (group, maven2, Online), maven-releases (hosted, maven2, Online), maven-snapshots (hosted, maven2, Online), nuget-group (group, nuget, Online), nuget-hosted (hosted, nuget, Online), and nuget.org-proxy (proxy, nuget, Online - Ready to Connect). Each row has a 'copy' button in the URL column. A red arrow points to the 'copy' button for the 'maven-releases' row.

Click on “copy” on “Maven-releases”

The screenshot shows the Sonatype Nexus Repository Manager interface with a modal dialog box overlaid on the 'Browse' table. The dialog is titled 'Copy to clipboard: Ctrl+C, Enter' and contains the text: 'Use your repository's direct URL (shown below) to connect other tools to your repository. For more information, see our [Maven-specific help documentation](#)'. Below this is a text input field containing the URL 'http://52.55.215.200:8081/repository/maven-releases/'. At the bottom of the dialog are 'Close' and 'OK' buttons. The background table remains visible with its original data.

`http://52.55.215.200:8081/repository/maven-releases/`

Then go to the pom.xml file in our GitHub repository

ebotsidneysmith / Boardgame

**Code** Issues Pull requests Actions Projects Security Insights Settings

**Boardgame** Private

main 1 Branch 0 Tags Go to file Add file Code About Jenkins Project

ebotsmith2000 Initial commit ✓ 7da1992 · yesterday 1 Commit

- .github/workflows Initial commit yesterday
- .mvn/wrapper Initial commit yesterday
- src Initial commit yesterday
- .gitignore Initial commit yesterday
- Dockerfile Initial commit yesterday
- Jenkinsfile Initial commit yesterday
- README.md Initial commit yesterday
- deployment-service.yaml Initial commit yesterday
- mvnw Initial commit yesterday
- mvnw.cmd Initial commit yesterday
- pom.xml** Initial commit yesterday
- sonar-project.properties Initial commit yesterday

README

**BoardgameListingWebApp**

Double-click on the file to open it

ebotsidneysmith / Boardgame

**Code** Issues Pull requests Actions Projects Security Insights Settings

**Files** main + Go to file

**Boardgame / pom.xml**

ebotsmith2000 Initial commit ✓ 7da1992 · yesterday History

Code	Blame	134 lines (122 loc) - 3.83 KB
<pre> 1 &lt;?xml version="1.0" encoding="UTF-8"?&gt; 2 &lt;project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" 3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"&gt; 4     &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt; 5     &lt;parent&gt; 6       &lt;groupId&gt;org.springframework.boot&lt;/groupId&gt; 7       &lt;artifactId&gt;spring-boot-starter-parent&lt;/artifactId&gt; 8       &lt;version&gt;2.5.6&lt;/version&gt; 9       &lt;relativePath/&gt; &lt;!-- lookup parent from repository --&gt; 10    &lt;/parent&gt; 11    &lt;groupId&gt;com.java_project&lt;/groupId&gt; 12    &lt;artifactId&gt;database_service_project&lt;/artifactId&gt; 13    &lt;version&gt;0.5-SNAPSHOT&lt;/version&gt; 14    &lt;name&gt;database_service_project&lt;/name&gt; 15    &lt;description&gt;Project for Spring Boot&lt;/description&gt; 16    &lt;packaging&gt;jar&lt;/packaging&gt; 17    &lt;properties&gt; 18      &lt;java.version&gt;11&lt;/java.version&gt; 19      &lt;jacoco.version&gt;0.8.7&lt;/jacoco.version&gt; 20      &lt;sonar.java.coveragePlugin&gt;jacoco/sonar.java.coveragePlugin&lt;/sonar.java.coveragePlugin&gt; 21      &lt;sonar.dynamicAnalysis&gt;reuseReports&lt;/sonar.dynamicAnalysis&gt; 22      &lt;sonar.jacoco.reportPath&gt;\${project.basedir}/..target/jacoco.exec&lt;/sonar.jacoco.reportPath&gt; 23      &lt;sonar.language&gt;javac/sonar.language&lt;/sonar.language&gt; 24    &lt;/properties&gt; 25 26    &lt;dependencies&gt; 27      &lt;dependency&gt; 28        &lt;groupId&gt;org.thymeleaf.extras&lt;/groupId&gt; 29        &lt;artifactId&gt;thymeleaf-extras-springsecurity5&lt;/artifactId&gt; 30      &lt;/dependency&gt; 31      &lt;dependency&gt;</pre>		

Scroll down to the end

https://github.com/ebotsidneysmith/Boardgame/blob/main/pom.xml

Code Blame 134 lines (122 loc) · 3.83 KB

```
96     <plugin>
97       <groupId>org.jacoco</groupId>
98       <artifactId>jacoco-maven-plugin</artifactId>
99       <version>${jacoco.version}</version>
100      <executions>
101        <execution>
102          <id>jacoco-initialize</id>
103          <goals>
104            <goal>prepare-agent</goal>
105          </goals>
106        </execution>
107        <execution>
108          <id>jacoco-site</id>
109          <phase>package</phase>
110          <goals>
111            <goal>report</goal>
112          </goals>
113        </execution>
114      </executions>
115    </plugin>
116  </plugins>
117 </build>
118
119
120 <distributionManagement>
121   <repository>
122     <id>maven-releases</id>
123     <url>http://13.127.177.61:8081/repository/maven-releases</url>
124   </repository>
125   <snapshotRepository>
126     <id>maven-snapshots</id>
127     <url>http://13.127.177.61:8081/repository/maven-snapshots</url>
128   </snapshotRepository>
129 </distributionManagement>
130
131
132
133
134 </project>
```

Then replace this part with the copied URL from Nexus. That is  
http://52.55.215.200:8081/repository/maven-releases/

https://github.com/ebotsidneysmith/Boardgame/blob/main/pom.xml

Code Blame 134 lines (122 loc) · 3.83 KB

```
96     <plugin>
97       <groupId>org.jacoco</groupId>
98       <artifactId>jacoco-maven-plugin</artifactId>
99       <version>${jacoco.version}</version>
100      <executions>
101        <execution>
102          <id>jacoco-initialize</id>
103          <goals>
104            <goal>prepare-agent</goal>
105          </goals>
106        </execution>
107        <execution>
108          <id>jacoco-site</id>
109          <phase>package</phase>
110          <goals>
111            <goal>report</goal>
112          </goals>
113        </execution>
114      </executions>
115    </plugin>
116  </plugins>
117 </build>
118
119
120 <distributionManagement>
121   <repository>
122     <id>maven-releases</id>
123     <url>http://13.127.177.61:8081/repository/maven-releases</url>
124   </repository>
125   <snapshotRepository>
126     <id>maven-snapshots</id>
127     <url>http://13.127.177.61:8081/repository/maven-snapshots</url>
128   </snapshotRepository>
129 </distributionManagement>
130
131
132
133
134 </project>
```

To do this, click on “Edit”

The screenshot shows the GitHub interface for editing the `pom.xml` file of a project named "Boardgame". The left sidebar lists project files like `.github`, `.mvn`, `src`, and `sonar-project.properties`. The main area displays the XML code for the `pom.xml` file. At the top right, there are two "Commit changes" buttons: a smaller one labeled "Commit changes..." and a larger green one labeled "Commit changes". A red arrow highlights the green "Commit changes" button.

Then, commit the changes by clicking on “Commit Changes”

The screenshot shows the GitHub interface for editing the `pom.xml` file of a project named "Boardgame". A modal window titled "Commit changes" is open over the code editor. It contains a "Commit message" field with the text "Update pom.xml", an "Extended description" field with placeholder text "Add an optional extended description...", and two radio button options: "Commit directly to the main branch" (selected) and "Create a new branch for this commit and start a pull request". At the bottom of the modal are "Cancel" and "Commit changes" buttons, with a red arrow pointing to the "Commit changes" button.

Confirm the commit by clicking on “Commit Changes” again

```

<?xml version="1.0" encoding="UTF-8">
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.5.6</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.javaproject</groupId>
  <artifactId>database_service_project</artifactId>
  <version>0.0.5-SNAPSHOT</version>
  <name>database_service_project</name>
  <description>Project for Spring Boot</description>
  <packaging>jar</packaging>
  <properties>
    <java.version>11</java.version>
    <jacoco.version>0.8.7</jacoco.version>
    <sonar.java.coveragePlugin>jacoco</sonar.java.coveragePlugin>
    <sonar.dynamicAnalysis>reuseReports</sonar.dynamicAnalysis>
    <sonar.jacoco.reportPath>${project.basedir}/../target/jacoco.exec</sonar.jacoco.reportPath>
    <sonar.language>javac</sonar.language>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.thymeleaf.extras</groupId>
      <artifactId>thymeleaf-extras-springsecurity5</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-security</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
    </dependency>
  </dependencies>

```

We will do same thing for snapshot. Go back to Nexus browser

Name	Type	Format	Status	URL	Health check
maven-central	proxy	maven2	Online - Ready to Connect		0 0 0 >
maven-public	group	maven2	Online		0 0 0 >
maven-releases	hosted	maven2	Online		0 0 0 >
maven-snapshots	hosted	maven2	Online		0 0 0 >
nuget-group	group	nuget	Online		0 0 0 >
nuget-hosted	hosted	nuget	Online		0 0 0 >
nuget.org-proxy	proxy	nuget	Online - Ready to Connect		0 0 0 >

Click on "copy" on "maven-snapshots"

The screenshot shows the Sonatype Nexus Repository Manager interface. On the left, there's a sidebar with 'Dashboard', 'Search', and 'Browse' buttons. The 'Browse' button is highlighted in green. The main area is titled 'Browse' and lists various repositories: maven-central, maven-public, maven-releases, maven-snapshots, nuget-group, nuget-hosted, and nuget.org-proxy. A modal dialog is centered over the list, with the title 'Copy to clipboard: Ctrl+C, Enter'. It contains instructions: 'Use your repository's direct URL (shown below) to connect other tools to your repository. For more information, see our Maven-specific help documentation.' Below the instructions is the URL: 'http://52.55.215.200:8081/repository/maven-snapshots/'. At the bottom of the modal is a 'Close' button.

## Copy the URL

<http://52.55.215.200:8081/repository/maven-snapshots/>

Then, go to the pom.xml file on your GitHub repository

The screenshot shows a GitHub repository page for 'ebotsidneysmith/Boardgame'. The 'Code' tab is selected. In the sidebar, the 'pom.xml' file is highlighted. The main area displays the content of the 'pom.xml' file. The code is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.5.6</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.javaproject</groupId>
  <artifactId>database_service_project</artifactId>
  <version>0.0.5-SNAPSHOT</version>
  <name>database_service_project</name>
  <description>Project for Spring Boot</description>
  <packaging>jar</packaging>
  <properties>
    <java.version>11</java.version>
    <jacoco.version>0.8.7</jacoco.version>
    <sonar.java.coveragePlugin>jacoco</sonar.java.coveragePlugin>
    <sonar.dynamicAnalysis>reuseReports</sonar.dynamicAnalysis>
    <sonar.jacoco.reportPath>${project.basedir}/../target/jacoco.exec</sonar.jacoco.reportPath>
    <sonar.language>java</sonar.language>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.thymeleaf.extras</groupId>
      <artifactId>thymeleaf-extras-springsecurity5</artifactId>
    </dependency>
    <dependency>
```

Scroll down to the end

https://github.com/ebotsidneysmith/Boardgame/blob/main/pom.xml

Code Blame 134 lines (122 loc) · 3.83 KB

```
96     <plugin>
97         <groupId>org.jacoco</groupId>
98         <artifactId>jacoco-maven-plugin</artifactId>
99         <version>${jacoco.version}</version>
100        <executions>
101            <execution>
102                <id>jacoco-initialize</id>
103                <goals>
104                    <goal>prepare-agent</goal>
105                </goals>
106            </execution>
107            <execution>
108                <id>jacoco-site</id>
109                <phase>package</phase>
110                <goals>
111                    <goal>report</goal>
112                </goals>
113            </execution>
114        </executions>
115    </plugin>
116    </plugins>
117  </build>
118
119
120  <distributionManagement>
121    <repository>
122        <id>maven-releases</id>
123        <url>http://52.55.215.200:8081/repository/maven-releases/</url>
124    </repository>
125    <snapshotRepository>
126        <id>maven-snapshots</id>
127        <url>http://13.127.177.61:8881/repository/maven-snapshots/</url>
128    </snapshotRepository>
129  </distributionManagement>
130
131
132
133
134 </project>
```

Paste the copies URL on the highlighted part. To do this, first click on “Edit”

https://github.com/ebotsidneysmith/Boardgame/edit/main/pom.xml

Edit Preview

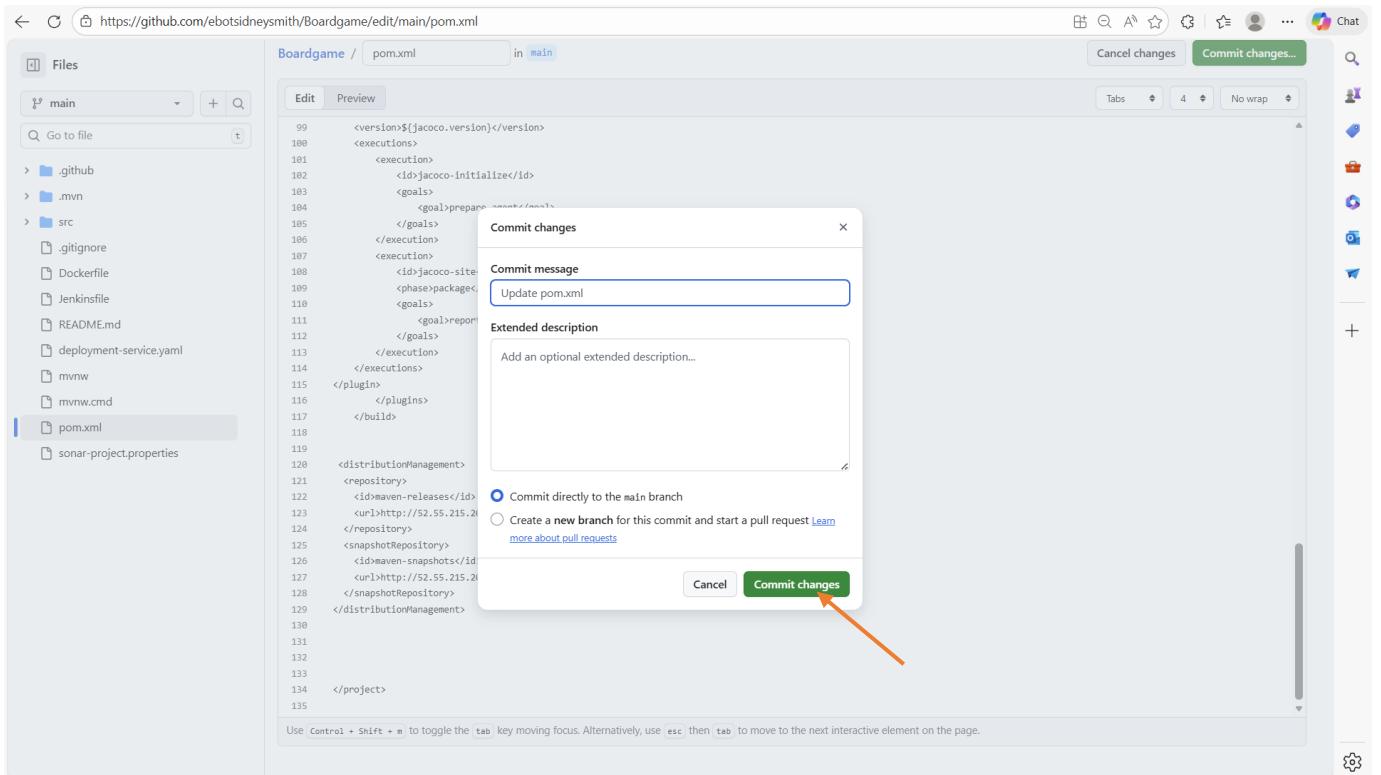
```
99     <version>${jacoco.version}</version>
100    <executions>
101        <execution>
102            <id>jacoco-initialize</id>
103            <goals>
104                <goal>prepare-agent</goal>
105            </goals>
106        </execution>
107        <execution>
108            <id>jacoco-site</id>
109            <phase>package</phase>
110            <goals>
111                <goal>report</goal>
112            </goals>
113        </execution>
114    </executions>
115    </plugin>
116    </plugins>
117  </build>
118
119
120  <distributionManagement>
121    <repository>
122        <id>maven-releases</id>
123        <url>http://52.55.215.200:8081/repository/maven-releases/</url>
124    </repository>
125    <snapshotRepository>
126        <id>maven-snapshots</id>
127        <url>http://13.127.177.61:8881/repository/maven-snapshots/</url>
128    </snapshotRepository>
129  </distributionManagement>
130
131
132
133
134 </project>
```

Paste the copied URL here

The screenshot shows the GitHub interface for editing the `pom.xml` file of a project named "Boardgame". The left sidebar lists project files like `.github`, `.mvn`, `src`, and `sonar-project.properties`. The main area displays the XML content of `pom.xml`. At the top right, there are "Cancel changes" and "Commit changes..." buttons. A red arrow highlights the "Commit changes..." button.

```
<version>${jacoco.version}</version>
<executions>
    <execution>
        <id>jacoco-initialize</id>
        <goals>
            <goal>prepare-agent</goal>
        </goals>
    </execution>
    <execution>
        <id>jacoco-site</id>
        <phase>package</phase>
        <goals>
            <goal>report</goal>
        </goals>
    </execution>
</executions>
</plugin>
</plugins>
</build>
</distributionManagement>
<repository>
    <id>maven-releases</id>
    <url>http://52.55.215.200:8081/repository/maven-releases/</url>
</repository>
<snapshotRepository>
    <id>maven-snapshots</id>
    <url>http://52.55.215.200:8081/repository/maven-snapshots/</url>
</snapshotRepository>
</distributionManagement>
```

Commit the changes by clicking on “Commit Changes”



Click on “Commit Changes” to confirm

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.5.6</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.database_service_project</groupId>
<artifactId>database_service_project</artifactId>
<version>0.0.5-SNAPSHOT</version>
<name>database_service_project</name>
<description>Project For Spring Boot</description>
<packaging>jar</packaging>
<properties>
<java.version>11</java.version>
<jacoco.version>0.8.7</jacoco.version>
<sonar.java.coveragePlugin>jacoco</sonar.java.coveragePlugin>
<sonar.dynamicAnalysis>reuseReports</sonar.dynamicAnalysis>
<sonar.jacoco.reportPath>${project.basedir}/..target/jacoco.exec</sonar.jacoco.reportPath>
<sonar.language>javac</sonar.language>
</properties>
<dependencies>
<dependency>
<groupId>org.thymeleaf.extras</groupId>
<artifactId>thymeleaf-extras-springsecurity5</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>

```

We have made the changes on the pom.xml file. Let us now continue to modify the “step” in this stage.

```

47~     stage('Build') {
48~         steps {
49~             sh "mvn package"
50~         }
51~     }
52~     stage('Publish to Nexus') {
53~         steps {
54~             sh "mvn package"
55~         }
56~     }
57~ }
58~ }
59~ 
```

Use Groovy Sandbox ?

**Pipeline Syntax**

Advanced

Save Apply

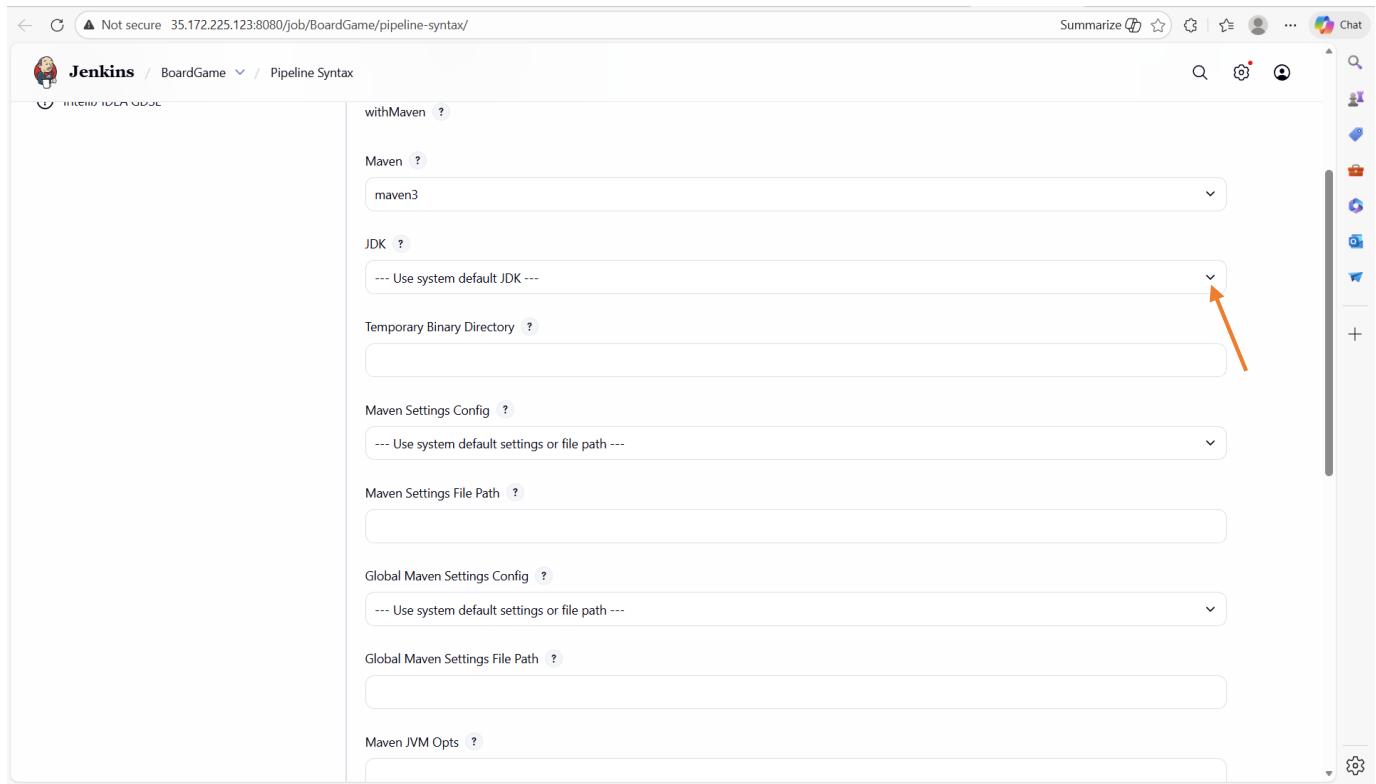
Click on “Pipeline Syntax”

This screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, there's a sidebar with links like 'Declarative Directive Generator', 'Declarative Online Documentation', 'Steps Reference', etc. The main area is titled 'Overview' and contains a section for 'Steps'. Under 'Sample Step', a dropdown menu is open, showing 'archiveArtifacts: Archive the artifacts'. Below this, there are fields for 'Files to archive' and an 'Advanced' button. At the bottom is a blue 'Generate Pipeline Script' button. An orange arrow points from the text above to the dropdown menu.

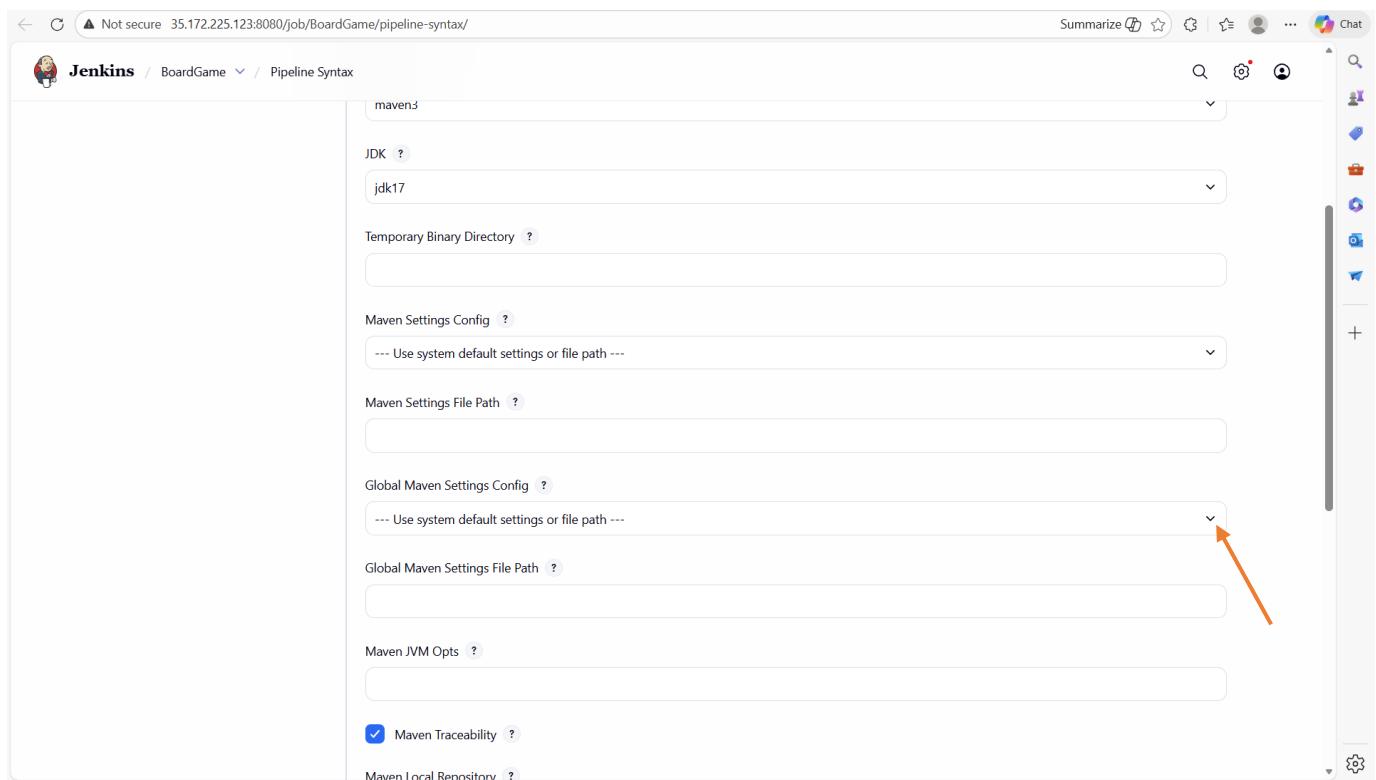
Click on the drop down on “Sample Step” and select “**withMaven**”

This screenshot shows the same Jenkins Pipeline Syntax Snippet Generator interface, but the 'Sample Step' dropdown now contains 'withMaven: Provide Maven environment'. The configuration page for 'withMaven' includes fields for 'Maven', 'JDK', 'Temporary Binary Directory', 'Maven Settings Config', and 'Maven Settings File Path'. An orange arrow points from the text above to the dropdown menu.

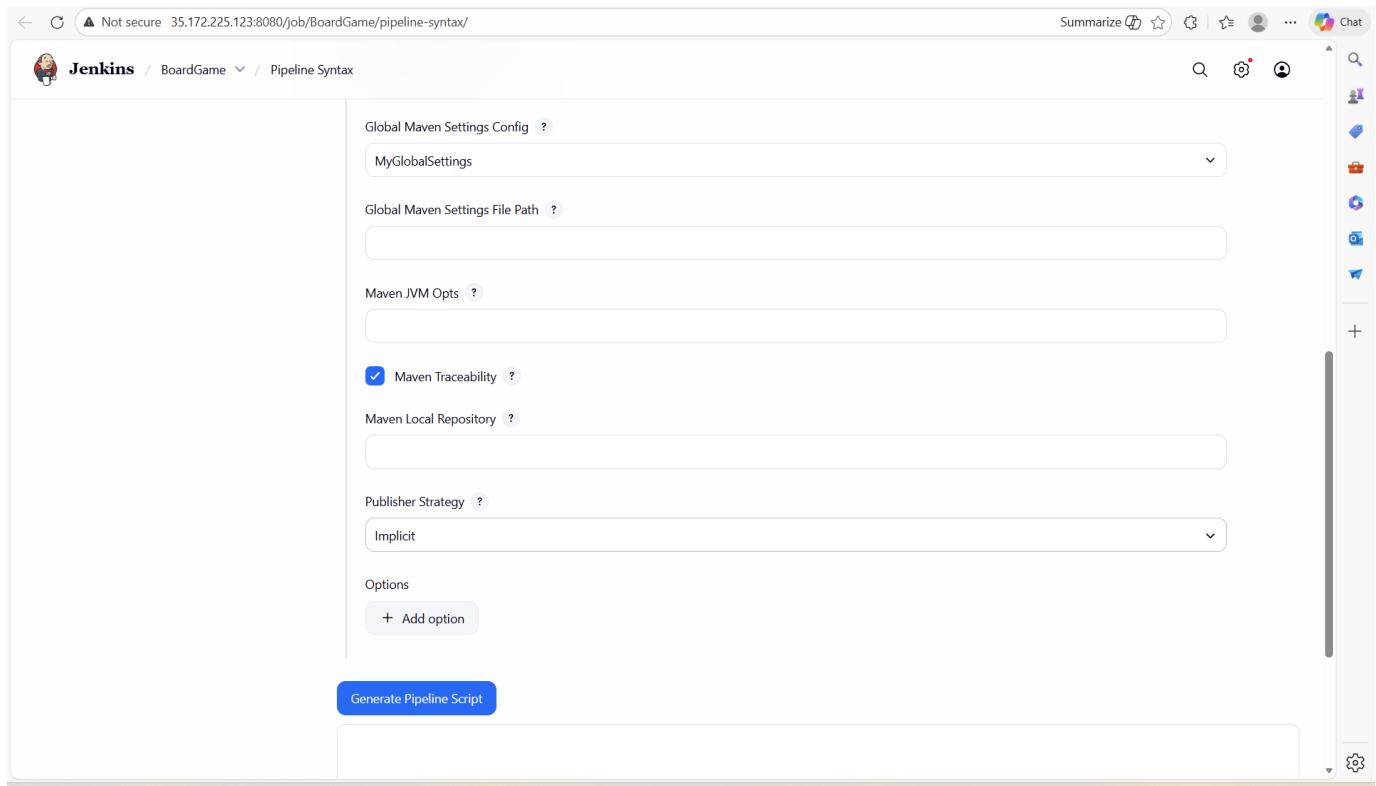
Click on the drop down on “**Maven**” and select “**maven3**”



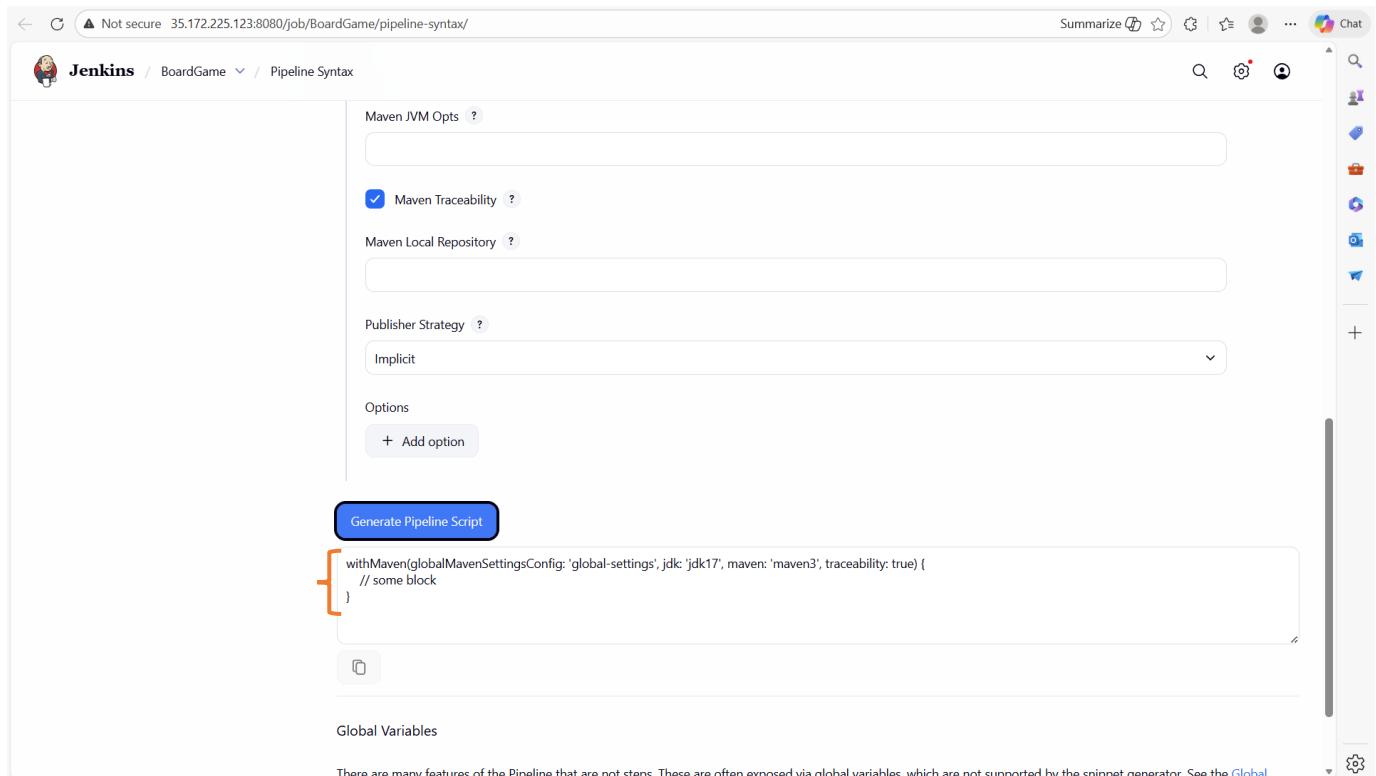
Click on the drop down on “JDK” and select “jdk17”



On “Global Maven Settings Config”, click on the drop down and select the file we have configured



Then, click on “Generate Pipeline Script”



Copy this block and paste in the “step” in the “Public to Nexus” stage.

```
withMaven(globalMavenSettingsConfig: 'global-settings', jdk: 'jdk17', maven: 'maven3', traceability: true){
```

```
// some block
}

<div>
  <div> Jenkins / BoardGame / Configuration </div>
  <div>
    <div> Configure </div>
    <div> General </div>
    <div> Triggers </div>
    <div> Pipeline </div> <span> Pipeline </span>
    <div> Advanced </div>
  </div>
  <div>
    <pre>
      49      sh "mvn package"
      50    }
      51  }
      52< stage('Publish to Nexus') {
      53<   steps {
      54<     withMaven(globalMavenSettingsConfig: 'global-settings', jdk: 'jdk17', maven: 'maven3', traceability: true)
      55       // some block
      56     }
      57   }
      58 }
      59 }
      60 }
      61
    </pre>
    <input checked="" type="checkbox"/> Use Groovy Sandbox ? <a href="#">Pipeline Syntax</a>
  </div>
  <div> Advanced </div>
  <div> Advanced </div>
  <div> Save </div> <div> Apply </div>
</div>

  REST API Jenkins 2.528.3 <img alt="Jenkins logo" data-bbox="928 461 941 474"/>

```

Then, we will add the shell command: `sh "mvn deploy"`

This is going to deploy the artifacts of the application to Nexus.

The complete stage is as follows:

```
stage('Publish To Nexus') {
  steps {
    withMaven(globalMavenSettingsConfig: 'global-settings', jdk: 'jdk17',
maven: 'maven3', mavenSettingsConfig: '', traceability: true) {
      sh "mvn deploy"
    }
  }
}
```

The screenshot shows the Jenkins Pipeline Configuration page for the 'BoardGame' pipeline. The left sidebar has 'Configure' selected under 'Pipeline'. The main area contains the pipeline script:

```

49      ' sh "mvn package"
50    }
51  }
52  stage('Publish to Nexus') {
53    steps {
54      withMaven(globalMavenSettingsConfig: 'global-settings', jdk: 'jdk17', maven: 'maven3', traceability: true) {
55        sh "mvn deploy"
56      }
57    }
58  }
59 }
60 }
61

```

Below the script is a checkbox labeled 'Use Groovy Sandbox'. At the bottom are 'Pipeline Syntax' and 'Advanced' tabs, and 'Save' and 'Apply' buttons.

We have completed this stage. Now, let us save it. Click on “**Save**”

The screenshot shows the Jenkins Pipeline Stage View page for the 'BoardGame' pipeline. The left sidebar has 'Status' selected. The main area displays the 'Stage View' with a table of average stage times:

	Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build
Average stage times: (full run time: ~49s)	203ms	572ms	3s	15s	1s	13s	443ms	22s
#6 Jan 11 16:36	212ms	562ms	3s	15s	1s	13s	432ms (paused for 1s)	22s
#5 Jan 11 16:33	195ms	583ms	3s	15s	1s	14s	454ms (paused for 2s)	

Below the table is a 'SonarQube Quality Gate' section showing 'BoardGame' as 'Passed' with 'server-side processing: Success'. The 'Permalinks' section lists recent builds:

- Last build (#5), 2 min 47 sec ago
- Last stable build (#5), 2 min 47 sec ago
- Last successful build (#5), 2 min 47 sec ago
- Last completed build (#5), 2 min 47 sec ago

Let us try to build the pipeline again. Click on “**Build Now**”

Not secure 54.226.220.79:8080/job/BoardGame/

**Jenkins / BoardGame**

**Status** **BoardGame** **Add description**

</> Changes **Build Now** **Configure** **Delete Pipeline** **Full Stage View** **SonarQube** **Stages** **Rename** **Pipeline Syntax**

**Builds** **Filter**

Today: #7 9:43 PM, #6 9:36 PM

**Stage View**

	Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus
Average stage times: (full run time: ~58s)	206ms	1s	3s	15s	1s	14s	443ms	19s	20s
#7 Jan 11 16:43 1 commit	213ms	3s	3s	15s	1s	14s	443ms (paused for 1s)	16s	20s
#6 Jan 11 16:36 No Changes	212ms	562ms	3s	15s	1s	13s	432ms (paused for 1s)	22s	
#5 Jan 11 16:33 No Changes	195ms	583ms	3s	15s	1s	14s	454ms (paused for 2s)		

**SonarQube Quality Gate**

BoardGame **Passed**  
server-side processing: **Success**

**Permalinks**

- Last build (#6), 7 min 5 sec ago
- Last stable build (#6), 7 min 5 sec ago

The build is successful.

#### 5.4.2.6 Configure Docker to Build the Docker Image

In this stage, we will use Docker to build a Docker image of the application. We will call this stage “**Build and Tag Docker Image**”.

Not secure 35.172.225.123:8080/job/BoardGame/configure

**Jenkins / BoardGame / Configuration**

**Configure** **General** **Triggers** **Pipeline** **Advanced**

```

54    withMaven(globalMavenSettingsConfig: 'global-settings', jdk: 'jdk17', maven: 'maven3', mavenSetting
55        sh "mvn deploy"
56    }
57 }
58 }
59 stage("Build and Tag Docker Image") {
60     steps {
61         sh "mvn package" ←
62     }
63 }
64 }
65 }
66

```

Use Groovy Sandbox ?

**Pipeline Syntax**

**Advanced**

**Save** **Apply**

REST API Jenkins 2.528.3

Then, let us modify the “**steps**” in this stage. We can get help from “**Pipeline Syntax**”. Click on “**Pipeline Syntax**”.

The screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, there's a sidebar with links like Snippet Generator, Declarative Directive Generator, Declarative Online Documentation, Steps Reference, Global Variables Reference, Online Documentation, Examples Reference, and IntelliJ IDEA GDSL. The main area has a title 'Overview' and a section 'Steps'. Under 'Sample Step', it shows 'archiveArtifacts: Archive the artifacts'. Below this, there's a dropdown menu labeled 'archiveArtifacts ?' with an orange arrow pointing to it. A sub-section 'Files to archive ?' contains a text input field. At the bottom of the 'Steps' section is a blue button 'Generate Pipeline Script'.

Click on the drop down on “Sample Step” and select “**withDockerRegistry**”

This screenshot is similar to the previous one but shows the 'withDockerRegistry' step selected. The 'Sample Step' dropdown now contains 'withDockerRegistry: Sets up Docker registry endpoint'. The 'withDockerRegistry ?' dropdown has an orange arrow pointing to it. Below it, there are fields for 'Docker registry URL ?' (with an orange arrow pointing to it) and 'Registry credentials' (which has a dropdown menu showing '- none -' and a '+ Add' button). There's also a 'Docker installation' dropdown set to '(Default)'. The 'Generate Pipeline Script' button is at the bottom.

Then, on “**Docker Registry URL**”, since we are using a public Docker registry, we are not going to provide a URL. But if you are using a private Docker registry, will have to provide a URL.

The screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, there's a sidebar with links like Snippet Generator, Declarative Directive Generator, Declarative Online Documentation, Steps Reference, Global Variables Reference, Online Documentation, Examples Reference, and IntelliJ IDEA GDSDL. The main area has a title 'Overview' with a description of the snippet generator. Below it, under 'Steps', there's a 'Sample Step' section containing a 'withDockerRegistry' step. This step has fields for 'Docker registry URL' (empty), 'Registry credentials' (set to '- none -'), and 'Docker installation' (set to '(Default)'). A blue 'Generate Pipeline Script' button is at the bottom. An orange arrow points to the dropdown menu for 'Registry credentials'.

The next thing is to select the Docker credentials on “**Registry Credentials**”. Click on the drop down.

This screenshot is similar to the previous one but shows the 'Registry credentials' dropdown expanded. The dropdown menu lists three items: '- none -', 'ebotsidneysmith/\*\*\*\*\*\*\*\* (git-credential)', and 'ebotsidneysmith/\*\*\*\*\*\*\*\* (docker-cred)'. The last item, 'ebotsidneysmith/\*\*\*\*\*\*\*\* (docker-cred)', is highlighted with a dark grey background and an orange arrow points to it. The rest of the interface is identical to the first screenshot.

## Select “docker-cred”

This screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, there's a sidebar with links like Snippet Generator, Declarative Directive Generator, and Steps Reference. The main area has a title 'Overview' and a description of the Snippet Generator. Below that is a 'Steps' section with a 'Sample Step' dropdown set to 'withDockerRegistry: Sets up Docker registry endpoint'. Underneath it, there are fields for 'Docker registry URL' (empty), 'Registry credentials' (set to 'ebotsidneysmith/\*\*\*\*\*\*\*\* (docker-cred)'), and 'Docker installation' (a dropdown menu). An orange arrow points from the text 'On “Docker Installation”, click on the drop down and select “docker”' to this 'Docker installation' dropdown. At the bottom is a blue 'Generate Pipeline Script' button.

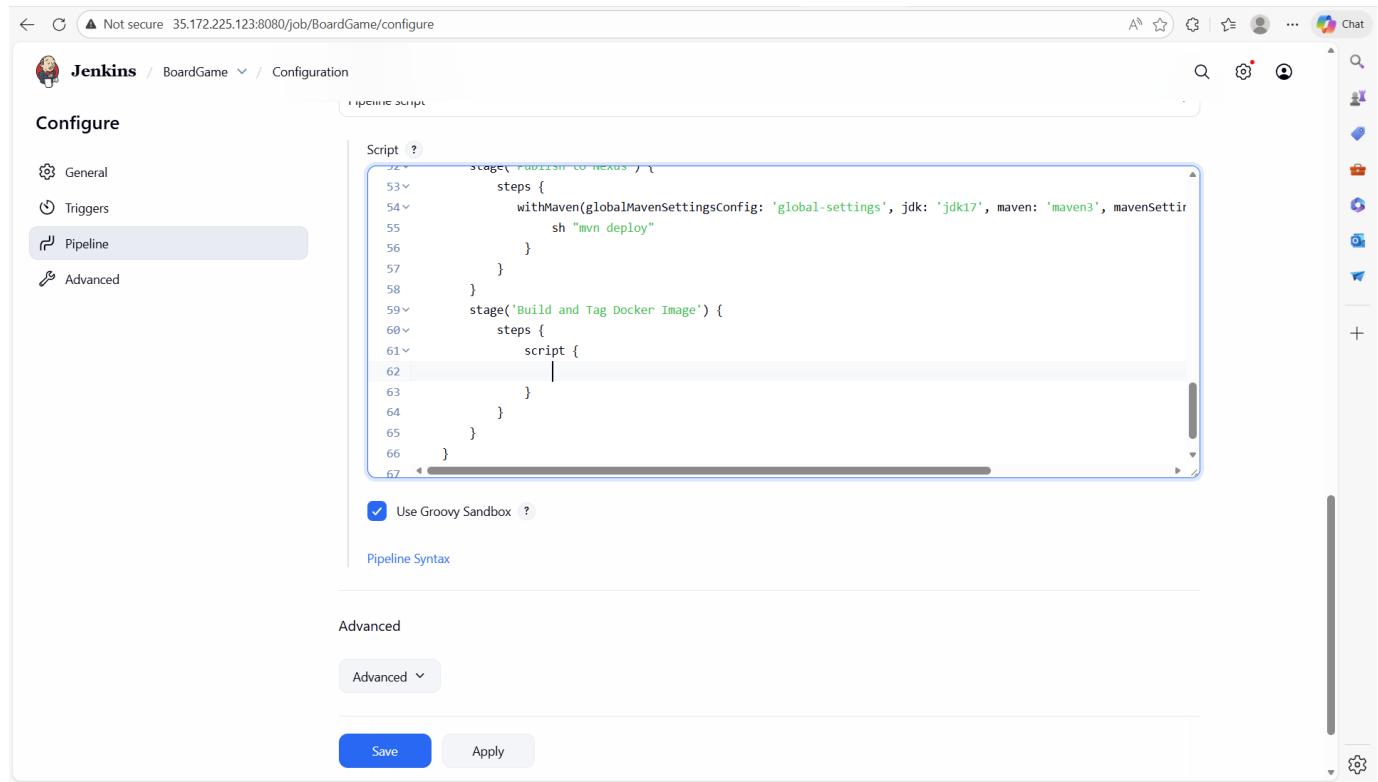
On “Docker Installation”, click on the drop down and select “docker”

This screenshot shows the same Jenkins Pipeline Syntax Snippet Generator interface as the previous one, but with a different selection in the 'Docker installation' dropdown. The dropdown now contains the word 'docker', indicated by an orange arrow. The rest of the interface is identical to the first screenshot, including the sidebar, overview text, sample step, and generate script button.

Click on “Generate Pipeline Script”

Copy this block of code, we will use it in the “step” in the stage. Head back to the Pipeline.

We will paste the block of code in a script. So let us add a script.

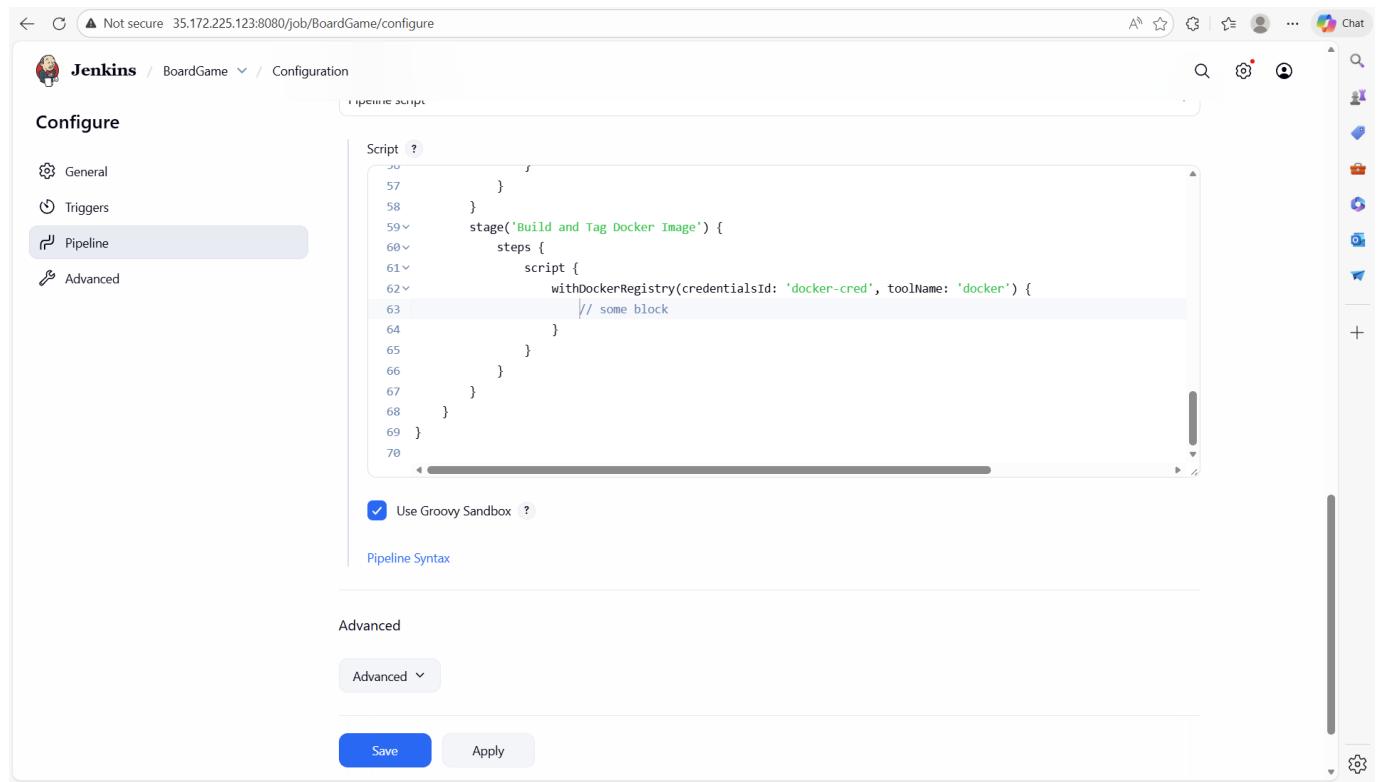


The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The left sidebar has "Pipeline" selected. The main area contains a Groovy script editor with the following code:

```
stage('Publish to Nexus') {
    steps {
        withMaven(globalMavenSettingsConfig: 'global-settings', jdk: 'jdk17', maven: 'maven3', mavenSettingName: 'maven3')
        sh "mvn deploy"
    }
}
stage('Build and Tag Docker Image') {
    steps {
        script {
            // some block
        }
    }
}
```

A checkbox for "Use Groovy Sandbox" is checked. Below the script editor are "Save" and "Apply" buttons.

Let us paste the block of code in the script



The screenshot shows the Jenkins Pipeline configuration page for the same job "BoardGame". The "Pipeline" option is selected in the sidebar. The script editor now includes the previously pasted code block:

```
stage('Publish to Nexus') {
    steps {
        withMaven(globalMavenSettingsConfig: 'global-settings', jdk: 'jdk17', maven: 'maven3', mavenSettingName: 'maven3')
        sh "mvn deploy"
    }
}
stage('Build and Tag Docker Image') {
    steps {
        script {
            withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {
                // some block
            }
        }
    }
}
```

The "Use Groovy Sandbox" checkbox is still checked. The "Save" and "Apply" buttons are present at the bottom.

Now, we can write the docker command to build the docker image:

```
sh "docker build -t ebotsidneysmith/demo:latest ."
```

The completed stage will be as follows:

```
stage('Build and Tag Docker Image') {  
    steps {  
        script {  
            withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {  
                sh "docker build -t ebotsidneysmith/demo:latest ."  
            }  
        }  
    }  
}
```

The screenshot shows the Jenkins Pipeline configuration page for a job named 'BoardGame'. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area contains a Groovy script editor with the provided code. A checkbox for 'Use Groovy Sandbox' is checked. Below the editor is a 'Pipeline Syntax' link. At the bottom, there is an 'Advanced' section with a dropdown menu set to 'Advanced'. A large orange arrow points from the bottom-left towards the 'Save' button, which is highlighted with a blue background. There are also 'Apply' and 'Cancel' buttons.

We have added the stage to build the docker image and tag it. Save by clicking on “Save”

**BoardGame**

**Last Successful Artifacts**

- database\_service\_project-0.0.5-SNAPSHOT.jar (45.89 MiB) [view](#)
- database\_service\_project-0.0.5-SNAPSHOT.pom (3.83 KiB) [view](#)

**Test Result Trend**

Passed (green dot), Skipped (grey dot), Failed (red dot)

**Stage View**

Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus
212ms	2s	3s	15s	1s	13s	437ms	19s	20s
#7 Jan 11 16:43 1 commit	213ms	3s	3s	15s	1s	14s	443ms (paused for 1s)	16s
#6 Jan 11 16:36 No Changes	212ms	562ms	3s	15s	1s	13s	432ms (paused for 1s)	22s

**SonarQube Quality Gate**

Let us build the Pipeline now. Click on “Build Now”

**BoardGame**

**Last Successful Artifacts**

- database\_service\_project-0.0.5-SNAPSHOT.jar (45.89 MiB) [view](#)
- database\_service\_project-0.0.5-SNAPSHOT.pom (3.83 KiB) [view](#)

**Test Result Trend**

Passed (green dot), Skipped (grey dot), Failed (red dot)

**Stage View**

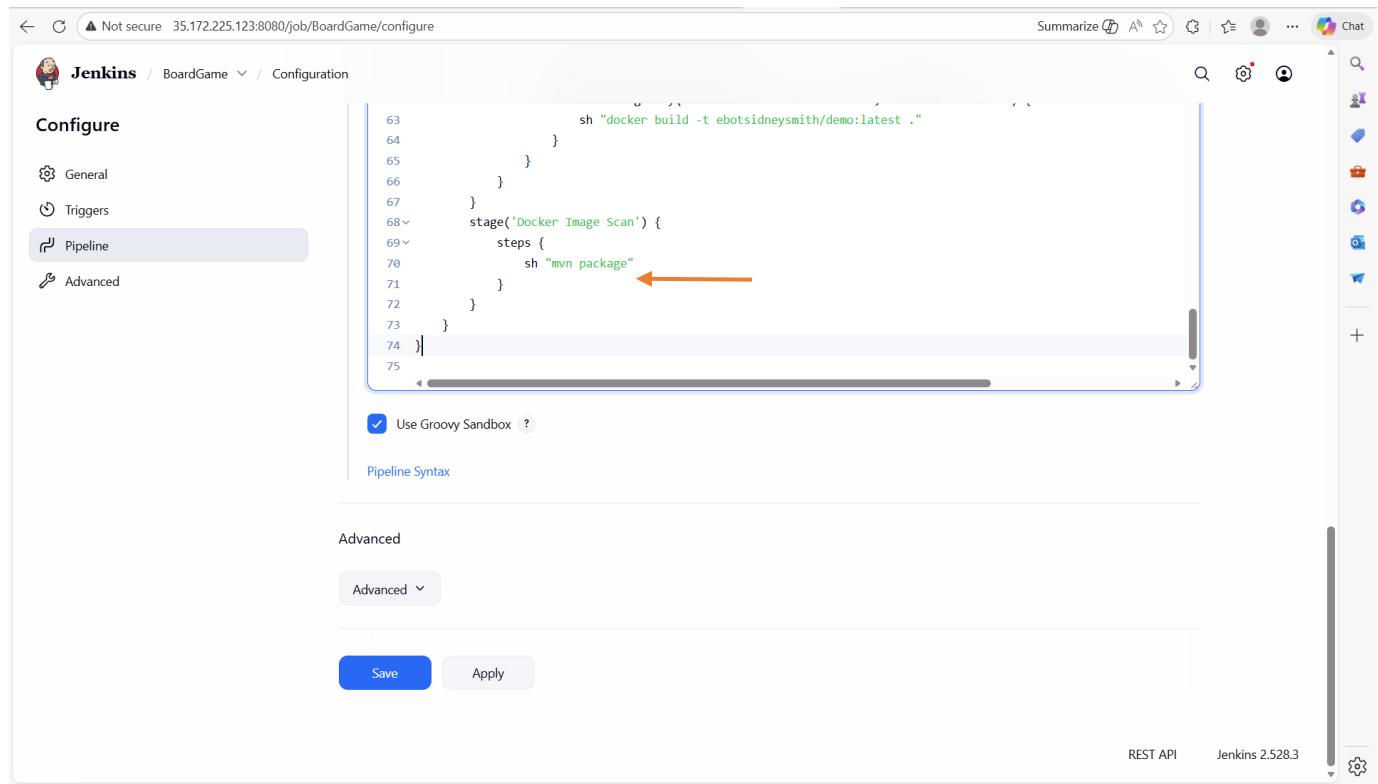
Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image
200ms	2s	3s	15s	1s	13s	438ms	16s	19s	54s
#15 Jan 11 18:26 No Changes	188ms	580ms	3s	15s	1s	13s	434ms (paused for 2s)	16s	19s
#7 Jan 11 16:43 1 commit	213ms	3s	3s	15s	1s	14s	443ms (paused for 1s)	16s	20s

**SonarQube Quality Gate**

The build is successful.

#### 5.4.2.7 Configure Trivy to Scan the Docker Image

In this stage, we will use Trivy to scan the Docker image of the application before we push it to Docker hub. We will call this stage “**Build and Tag Docker Image**”. We will call the stage “**Docker Image Scan**”



The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The left sidebar has "Pipeline" selected. The main area contains a Groovy script editor with the following code:

```
63      sh "docker build -t ebotsidneysmith/demo:latest ."
64    }
65  }
66 }
67 }
68 stage('Docker Image Scan') {
69   steps {
70     sh "mvn package"
71   }
72 }
73 }
74 }
```

An orange arrow points to the line "sh "mvn package"" in the script. Below the script, there is a checkbox labeled "Use Groovy Sandbox" which is checked. At the bottom of the editor, there are "Save" and "Apply" buttons. The status bar at the bottom right shows "REST API" and "Jenkins 2.528.3".

Let us modify the “step” in this stage. We will add the shell command:

```
sh "trivy image --format table -o trivy-image-report.html ebotsidneysmith/demo:latest"
```

The completed stage will be as follows:

```
stage('Docker Image Scan') {
  steps {
    sh "trivy image --format table -o trivy-image-report.html ebotsidneysmith/demo:latest"
  }
}
```

Not secure 35.172.225.123:8080/job/BoardGame/configure

**Jenkins / BoardGame / Configuration**

**Configure**

- General
- Triggers
- Pipeline**
- Advanced

```

63         sh "docker build -t ebotsidneysmith/demo:latest ."
64     }
65   }
66 }
67 }
68 stage('Docker Image Scan') {
69   steps {
70     sh "trivy image --format table -o trivy-image-report.html ebotsidneysmith/demo:latest"
71   }
72 }
73 }
74 }
75

```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

**Advanced**

Advanced

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

REST API Jenkins 2.528.3

Then save by clicking on “Save”

Not secure 54.226.220.79:8080/job/BoardGame/

**Jenkins / BoardGame**

[Status](#) **BoardGame** [Add description](#)

[Changes](#) [Build Now](#) [Configure](#) [Delete Pipeline](#) [Full Stage View](#) [SonarQube](#) [Stages](#) [Rename](#)

**Maven** [Pipeline Syntax](#)

**Builds**

#	Date	Commit	Status
#15	Jan 11 18:26	No Changes	Success
#7	Jan 11 16:43	1 commit	Success

[Filter](#)

**Last Successful Artifacts**

- [database\\_service\\_project-0.0.5-SNAPSHOT.jar](#) 45.89 MiB [view](#)
- [database\\_service\\_project-0.0.5-SNAPSHOT.pom](#) 3.83 KiB [view](#)

**Test Result Trend**

**Stage View**

	Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image
Average stage times: (full run time: ~1min 42s)	200ms	2s	3s	15s	1s	13s	438ms	16s	19s	54s
#15	188ms	580ms	3s	15s	1s	13s	434ms (paused for 2s)	16s	19s	54s
#7	213ms	3s	3s	15s	1s	14s	443ms (paused for 1s)	16s	20s	

**SonarQube Quality Gate**

Then build the pipeline by clicking on “Build Now”

#### 5.4.2.8 Configure Docker to Push the Docker Image

In this stage, we will use Docker to push the Docker image of the application to the Docker hub. We will call this stage “**Push Docker Image**”.

Let us start by naming the step.

```

68<v
69<v
70<v
71<v
72<v
73<v
74<v
75<v
76<v
77<v
78<v
79<v
80<v
    stage('Docker Image Scan') {
        steps {
            sh "trivy image --format table -o trivy-image-report.html ebotsidneysmith/demo:latest"
        }
    }
    stage('Push Docker Image') {
        steps {
            sh "trivy image --format table -o trivy-image-report.html ebotsidneysmith/demo:latest"
        }
    }
}

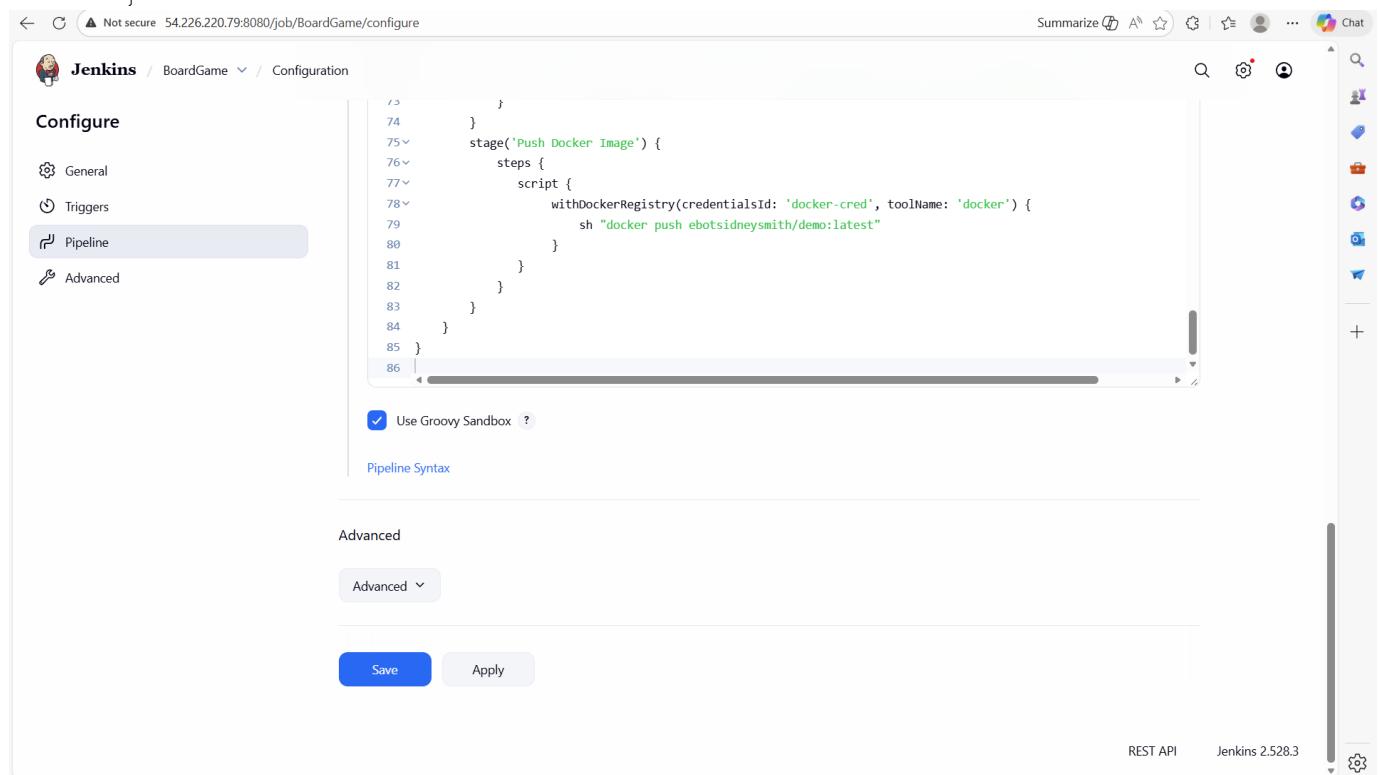
```

Let us now modify the “**step**” in this stage. The “**step**” will be similar to the “**step**” in the stage “**Build and Tag Docker Image**”, we just have to change the shell command. The shell command will be as follows:

```
sh "docker push ebotsidneysmith/demo:latest"
```

And the complete stage will be as follows:

```
stage('Push Docker Image') {  
    steps {  
        script {  
            withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {  
                sh "docker push ebotsidneysmith/demo:latest"  
            }  
        }  
    }  
}
```



The screenshot shows the Jenkins Pipeline configuration page for a job named 'BoardGame'. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area contains Groovy code for the pipeline stages. A checkbox for 'Use Groovy Sandbox' is checked. Below the code is a 'Pipeline Syntax' link. At the bottom, there's an 'Advanced' section with a dropdown menu set to 'Advanced'. Two buttons, 'Save' and 'Apply', are at the bottom right. The top right corner shows the Jenkins version 'Jenkins 2.529.3'.

```
73  
74    }  
75    stage('Push Docker Image') {  
76        steps {  
77            script {  
78                withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {  
79                    sh "docker push ebotsidneysmith/demo:latest"  
80                }  
81            }  
82        }  
83    }  
84}  
85}  
86
```

Use Groovy Sandbox ?

Pipeline Syntax

Advanced

Advanced ▾

Save Apply

REST API Jenkins 2.529.3

Then, click on “Save”

Not secure 54.226.220.79:8080/job/BoardGame/

**Jenkins / BoardGame**

Full Stage View | SonarQube | Stages | Rename | Maven | Pipeline Syntax

Builds: Average stage times: full run time: ~1min 54s

	Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan
#15	179ms	561ms	3s	15s	1s	13s	419ms	16s	19s	27s	28s
#16	171ms	543ms	3s	15s	1s	13s	404ms (paused for 795ms)	15s	19s	1s	28s
	188ms	580ms	3s	15s	1s	13s	434ms (paused for 2s)	16s	19s	54s	

**SonarQube Quality Gate**

BoardGame Passed  
server-side processing: Success

Latest Test Result (no failures)

Permalinks

Then build the pipeline by clicking on “Build Now”

Not secure 54.226.220.79:8080/job/BoardGame/

**Jenkins / BoardGame**

Build Now | Configure | Delete Pipeline | Full Stage View | SonarQube | Stages | Rename | Maven | Pipeline Syntax

Last Successful Artifacts: database\_service\_project-0.0.5-SNAPSHOT.jar (45.89 MB), database\_service\_project-0.0.5-SNAPSHOT.pom (3.83 KB)

Test Result Trend: Passed (green), Skipped (grey), Failed (red)

	Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan	Push Docker Image
stage times: ~1min 32s	175ms	568ms	3s	15s	1s	13s	423ms	16s	19s	1s	14s	7s
#15	180ms	593ms	3s	15s	1s	13s	442ms (paused for 1s)	16s	18s	1s	594ms	7s
#16	171ms	543ms	3s	15s	1s	13s	404ms (paused for 795ms)	15s	19s	1s	28s	

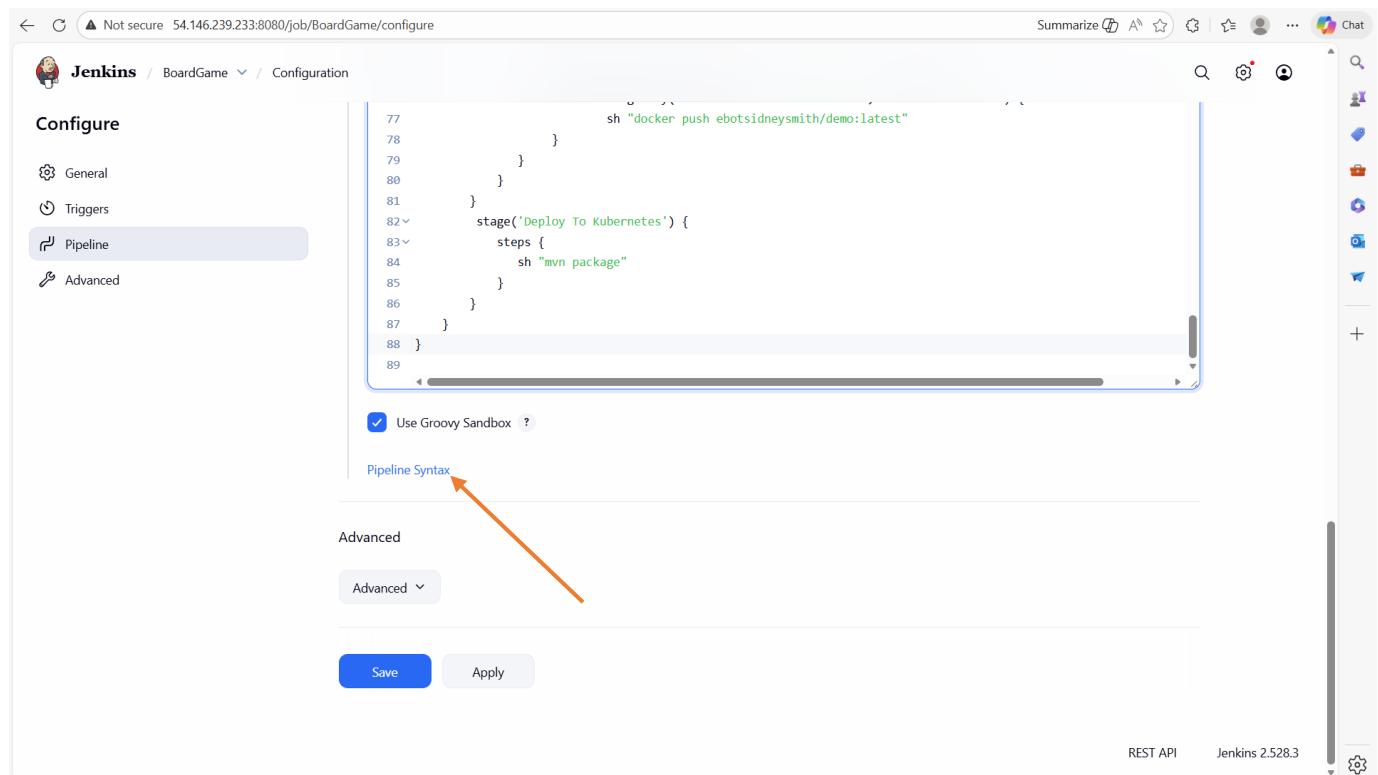
**SonarQube Quality Gate**

BoardGame Passed  
server-side processing: Success

The build is successful.

#### 5.4.2.9 Configure Kubernetes to Deploy the Docker Image to Kubernetes

In this stage, we will use Kubernetes to deploy the Docker image of the application to the Kubernetes. We will call this stage “**Deploy To Kubernetes**”.

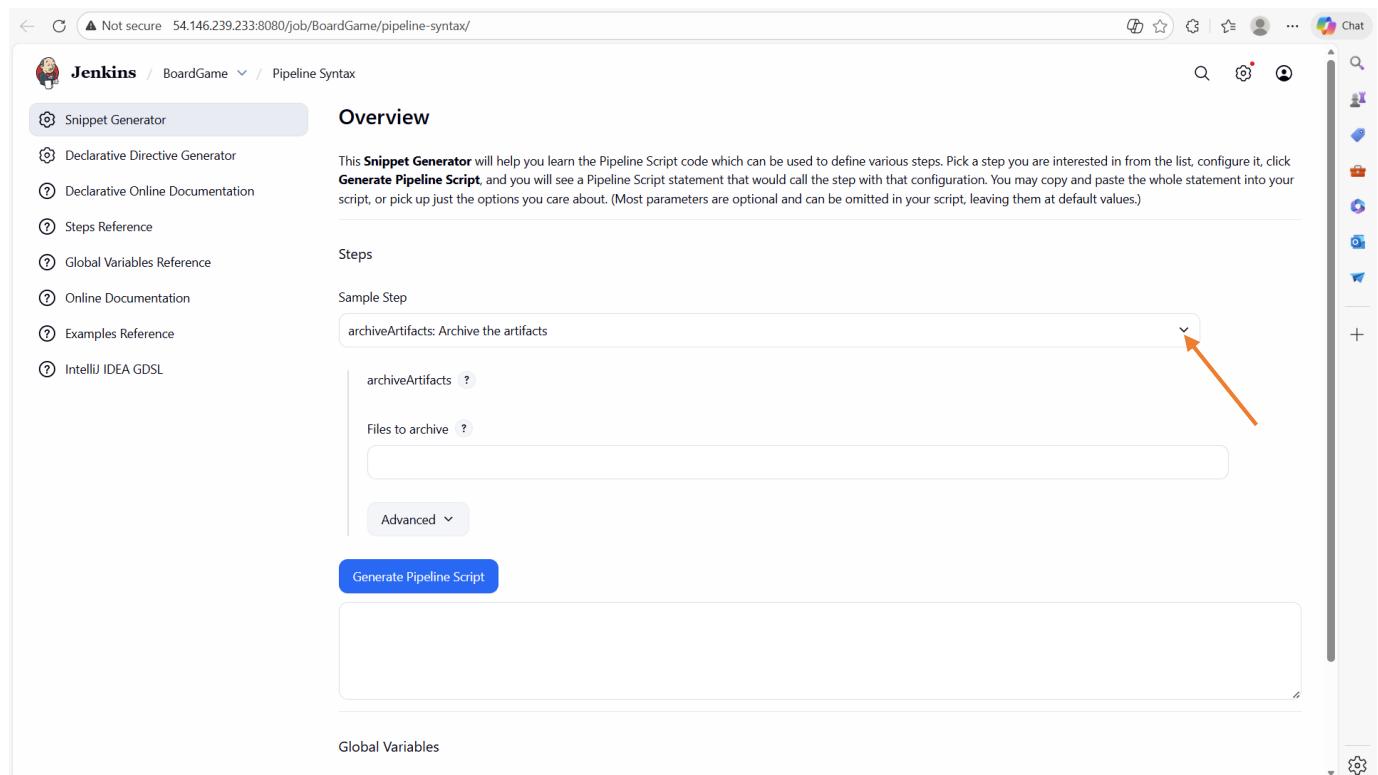


The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The left sidebar has "Pipeline" selected under "Configure". The main area displays Groovy pipeline script code:

```
77      }
78    }
79  }
80  }
81  }
82  stage('Deploy To Kubernetes') {
83    steps {
84      sh "mvn package"
85    }
86  }
87 }
88 }
89 }
```

A checkbox labeled "Use Groovy Sandbox" is checked. Below the script, there is a section titled "Pipeline Syntax" with an orange arrow pointing to it from the text above. At the bottom are "Save" and "Apply" buttons.

Now, let us modify the “step” in this stage. Click on “Pipeline Syntax”



The screenshot shows the Jenkins Snippet Generator Pipeline Syntax page. The left sidebar has "Pipeline Syntax" selected under "Snippet Generator". The main area has a heading "Overview" and a description of the Snippet Generator. Below is a "Steps" section with a "Sample Step" example:

```
archiveArtifacts: Archive the artifacts
```

An orange arrow points from the text above to the "archiveArtifacts" step in the sample. The "Generate Pipeline Script" button is visible at the bottom.

Click on the drop down on “**Sample Step**” and select “**withKubeConfig**”

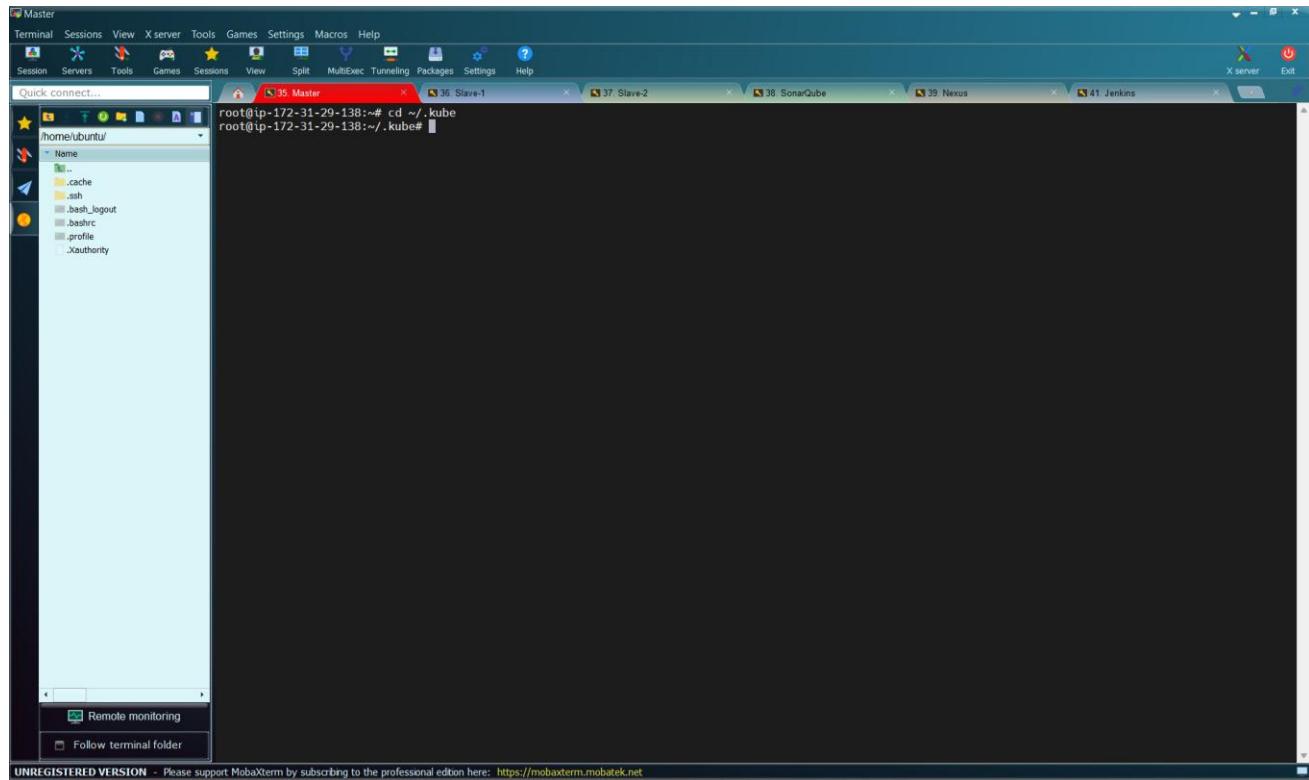
The screenshot shows the Jenkins Pipeline Syntax Snippet Generator interface. On the left, there's a sidebar with links like 'Declarative Directive Generator', 'Steps Reference', and 'Examples Reference'. The main area has a title 'Overview' and a section 'Steps'. Under 'Steps', there's a dropdown menu labeled 'withKubeConfig: Configure Kubernetes CLI (kubectl)'. An orange arrow points from the text above to the 'withKubeConfig' option in the dropdown. Below the dropdown, there are several input fields: 'Credentials' (set to 'none'), 'Kubernetes server endpoint', 'Cluster name', 'Context name', and 'Namespace'.

On the “**credentials**”, click on the drop down and select the Kubernetes token

This screenshot is similar to the previous one, but the 'Credentials' dropdown now contains the value 'k8-cred'. An orange arrow points from the text above to the 'k8-cred' entry in the dropdown. The other fields ('Kubernetes server endpoint', 'Cluster name', 'Context name', 'Namespace') remain the same as in the first screenshot.

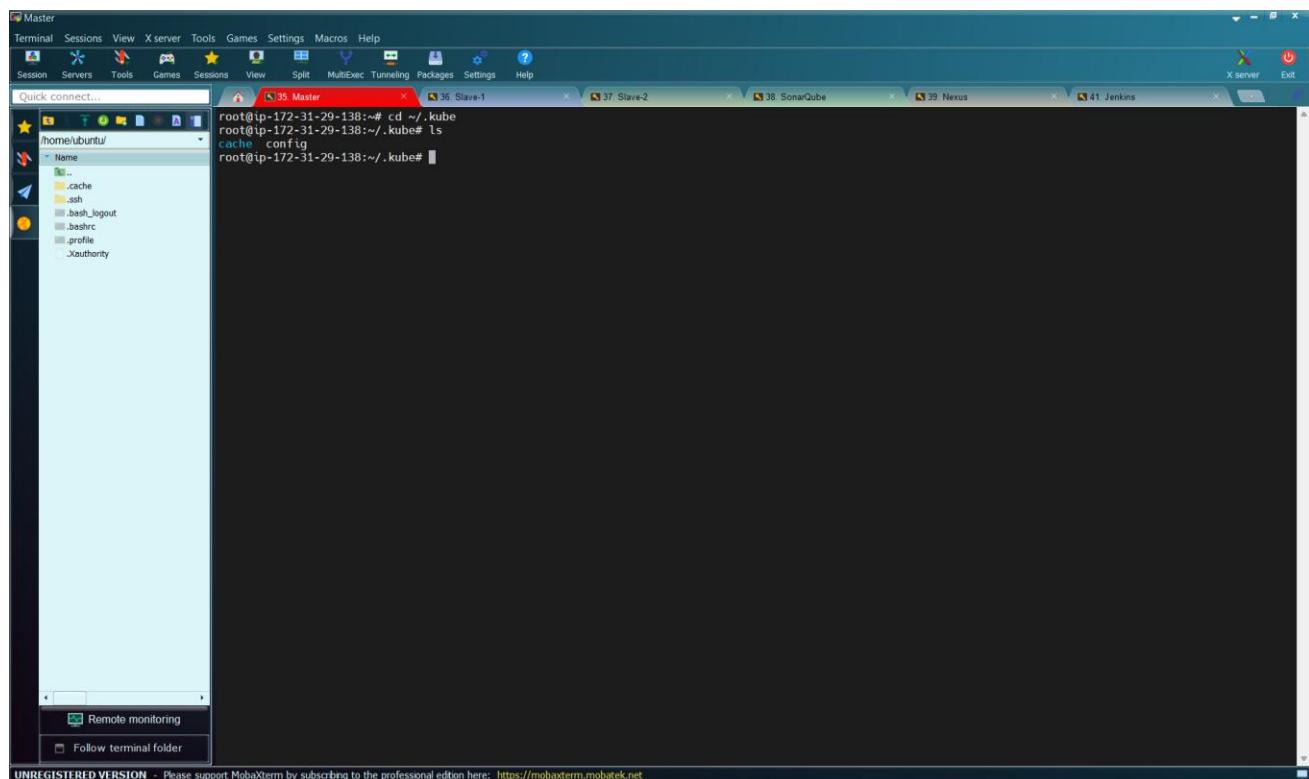
Now, we have to get the Kubernetes server endpoint as follows, go back to the terminal of the “**Master**” server and run the command:

```
cd ~/.kube
```



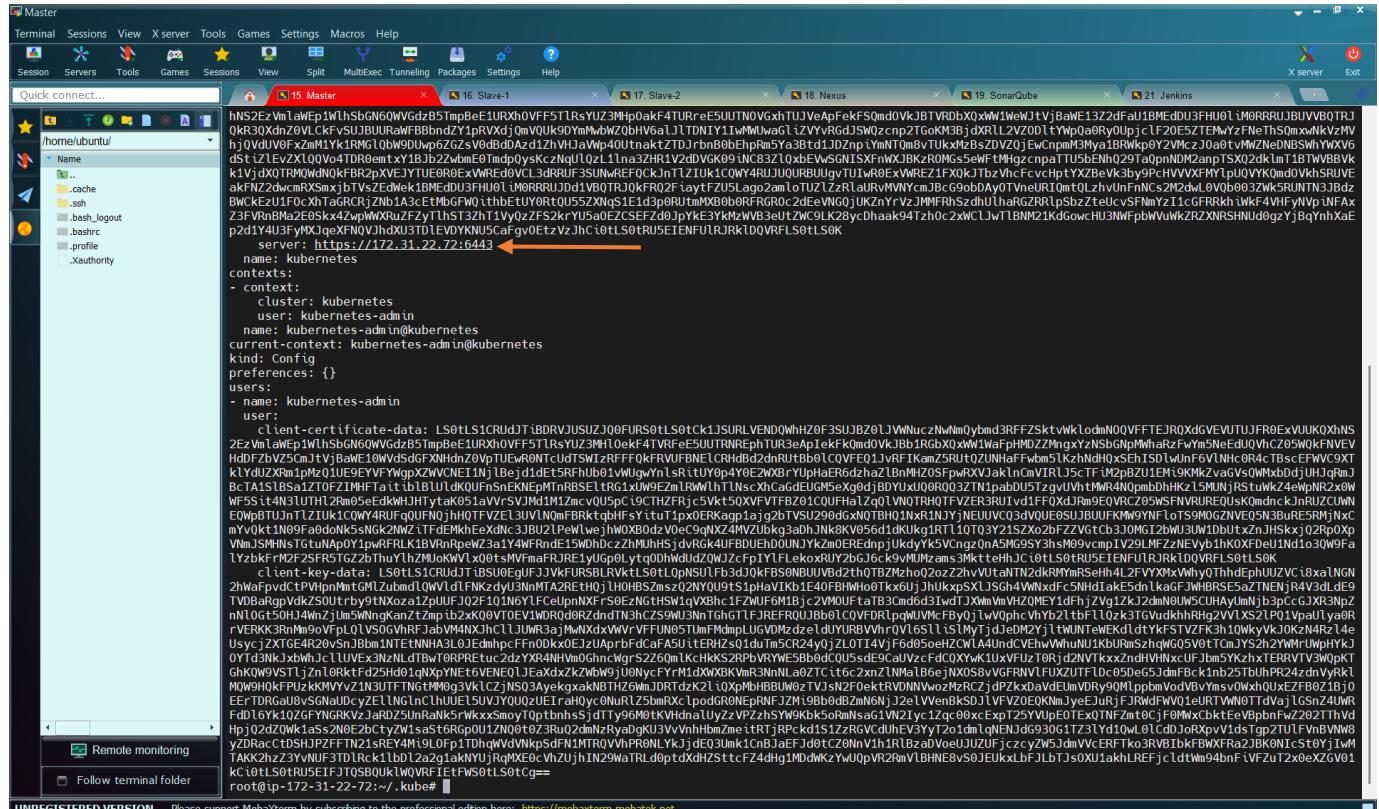
Then, run the command to see the content of the folder:

```
ls
```



Then, run the command to open the “config” file:

cat config



That is the endpoint. Copy it.

<https://172.31.22.72:6443>

Paste it in the “**Pipeline Syntax**” page

Not secure 54.146.239.233:8080/job/BoardGame/pipeline-syntax/ Summarize ⚡ ☆ ⚡ Chat

Jenkins / BoardGame / Pipeline Syntax

Online Documentation Examples Reference IntelliJ IDEA GDSL

Sample Step

withKubeConfig: Configure Kubernetes CLI (kubectl)

withKubeConfig ?

Credentials

k8-cred

+ Add

Kubernetes server endpoint ?

https://172.31.29.138:6443

Cluster name ?

Context name ?

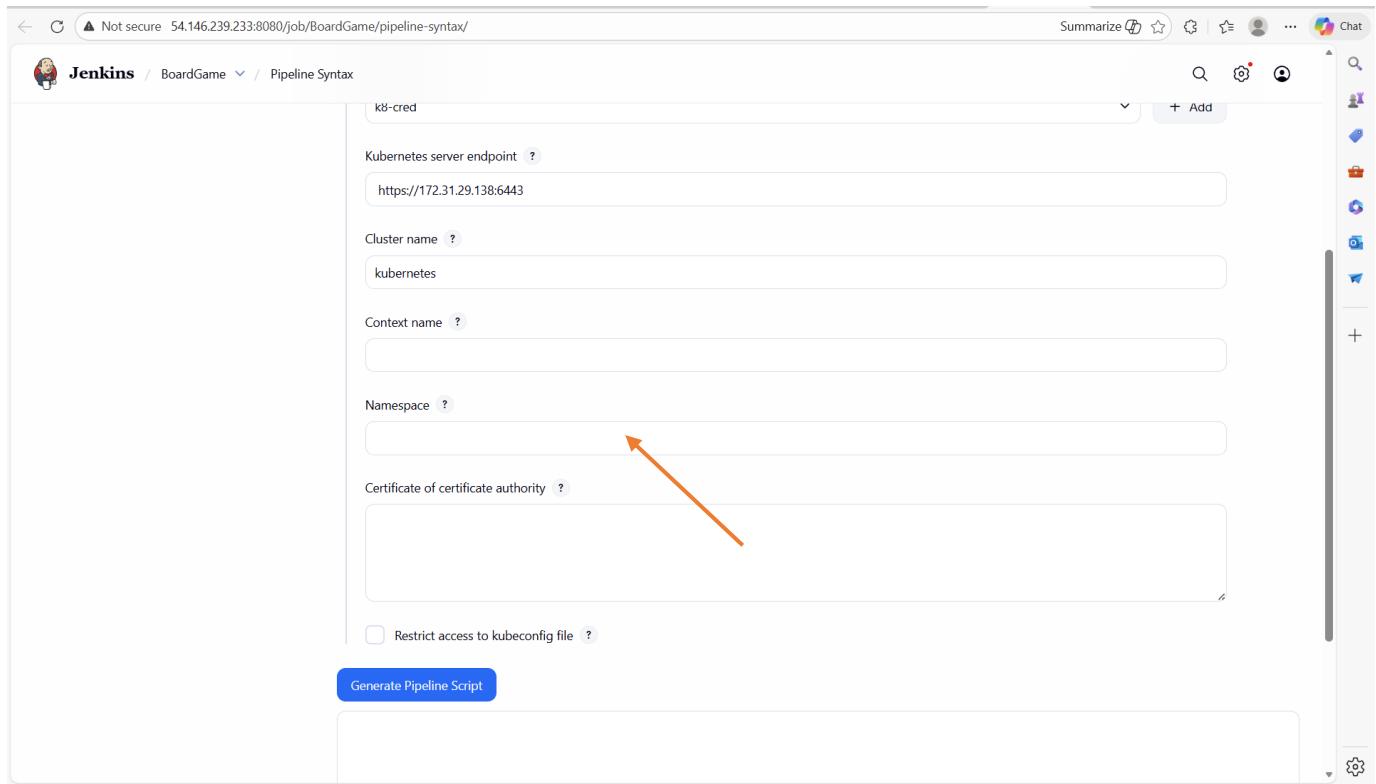
Namespace ?

Certificate of certificate authority ?



We can also get the cluster name from the config file.

The “Cluster Name” is “**kubernetes**”



Jenkins / BoardGame / Pipeline Syntax

k8s-cred

Kubernetes server endpoint ?  
https://172.31.29.138:6443

Cluster name ?  
kubernetes

Context name ?

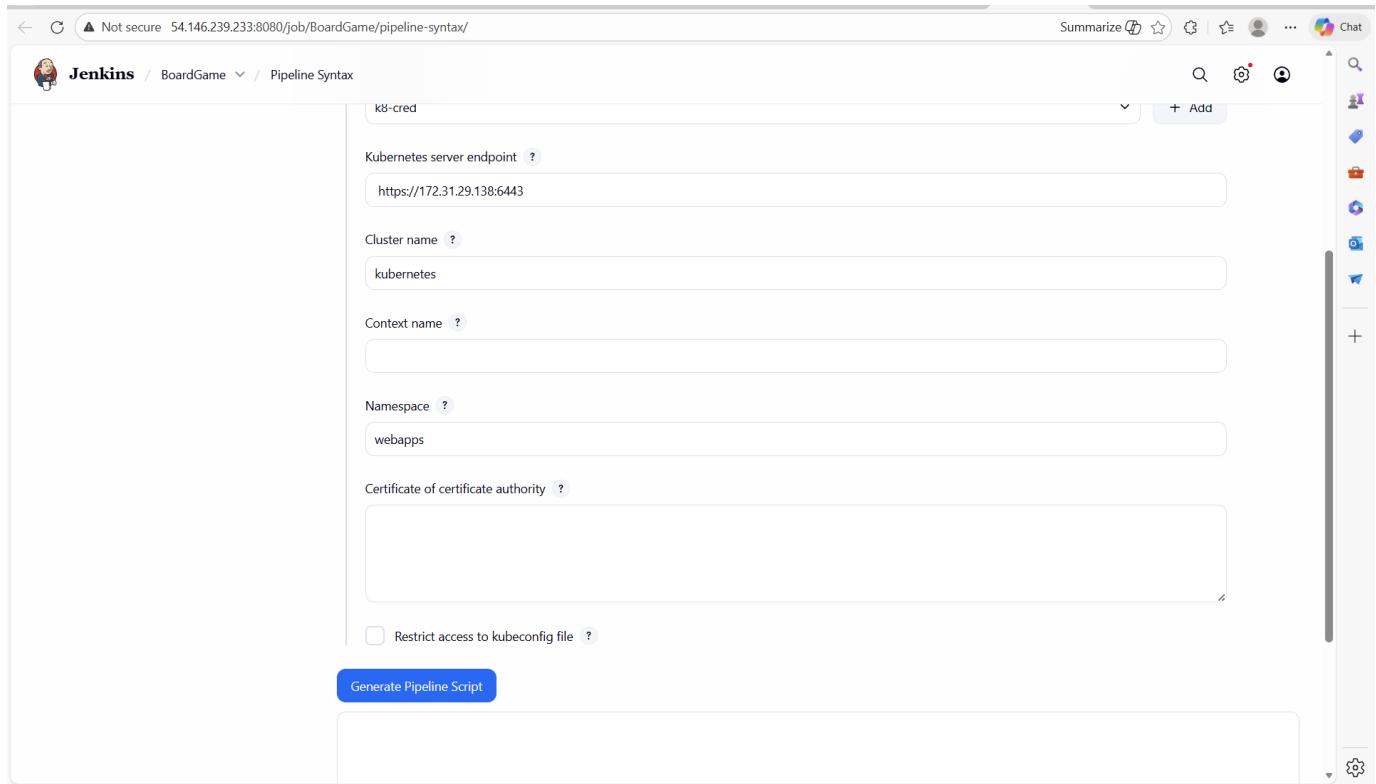
Namespace ?

Certificate of certificate authority ?

Restrict access to kubeconfig file ?

Generate Pipeline Script

Then, enter the namespace, that is “**webapps**”



Jenkins / BoardGame / Pipeline Syntax

k8s-cred

Kubernetes server endpoint ?  
https://172.31.29.138:6443

Cluster name ?  
kubernetes

Context name ?

Namespace ?  
webapps

Certificate of certificate authority ?

Restrict access to kubeconfig file ?

Generate Pipeline Script

Click on “**Generate Pipeline Script**”

Context name ?

Namespace ?  
webapps

Certificate of certificate authority ?

Restrict access to kubeconfig file ?

Generate Pipeline Script

```
withKubeConfig(caCertificate: '', clusterName: 'kubernetes', contextName: '', credentialsId: 'k8-cred', namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl: 'https://172.31.22.72:6443') {
    // some block
}
```

Global Variables

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the [Global Variables Reference](#) for details.

Copy the block of code and paste in the “step” of the stage

Configure

- General
- Triggers
- Pipeline
- Advanced

```
79
80
81
82    }
83    stage('Deploy To Kubernetes') {
84        steps {
85            withKubeConfig(caCertificate: '', clusterName: 'kubernetes', contextName: '', credentialsId: 'k8-cred', namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl: 'https://172.31.22.72:6443') {
86                // some block
87            }
88        }
89    }
90
91
```

Use Groovy Sandbox ?

Pipeline Syntax

Advanced

Advanced ▾

Save Apply

REST API Jenkins 2.528.3

Now, let us enter the shell command in the block. We will use the command:

Sh "kubectl apply -f deployment-service.yaml"

The complete for this stage will be as follows:

```
stage('Deploy To Kubernetes') {
    steps {
        withKubeConfig(caCertificate: '', clusterName: 'kubernetes', contextName: '',
credentialsId: 'k8-cred', namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl:
'https://172.31.22.72:6443') {
            sh "kubectl apply -f deployment-service.yaml"
        }
    }
}
```

The screenshot shows the Jenkins Pipeline configuration page for a job named 'BoardGame'. The left sidebar has 'Pipeline' selected under 'Configure'. The main area contains a Groovy script block with code for deploying to Kubernetes. A checkbox for 'Use Groovy Sandbox' is checked. Below the script is a 'Pipeline Syntax' link. At the bottom, there is an 'Advanced' dropdown set to 'Advanced', a 'Save' button highlighted with an orange arrow, and an 'Apply' button.

Now click on “Save”.

Then, the next thing to do is to create a “**Manifest**” file on our GitHub repository for this project. We will call the file “**deployment-service.yaml**” with the code below:

```
apiVersion: apps/v1
kind: Deployment # Kubernetes resource kind we are creating
metadata:
  name: boardgame-deployment
spec:
  selector:
    matchLabels:
      app: boardgame
  replicas: 2 # Number of replicas that will be created for this deployment
  template:
    metadata:
      labels:
        app: boardgame
    spec:
      containers:
        - name: boardgame
          image: ebotsidneysmith/demo:latest # Image that will be used to containers in the
cluster
          imagePullPolicy: Always
        ports:
          - containerPort: 8080 # The port that the container is running on in the cluster
```

---

```
apiVersion: v1 # Kubernetes API version
kind: Service # Kubernetes resource kind we are creating
metadata: # Metadata of the resource kind we are creating
  name: boardgame-ssvc
spec:
  selector:
    app: boardgame
  ports:
    - protocol: "TCP"
      port: 80
      targetPort: 8080
  type: LoadBalancer # type of the service.
```

The screenshot shows the Jenkins Pipeline interface for the 'BoardGame' project. The pipeline has failed at the 'SonarQube Quality Gate' stage. A red arrow points to the 'Build Now' button in the sidebar. The pipeline stages and their execution times are listed below:

Stage	Time
Declarative: Tool Install	175ms
Git Checkout	568ms
Compile	3s
Test	15s
File System Scan	1s
SonarQube Analysis	13s
Quality Gate	423ms (paused for 1s)
Build	16s
Publish To Nexus	19s
Build & Tag Docker Image	1s
Docker Image Scan	14s
Push Docker Image	7s

Then, build the pipeline by clicking on “Build Now”

Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan	Push Docker Image	Deploy To Kubernetes
186ms	578ms	3s	15s	1s	13s	426ms	16s	19s	1s	10s	5s	1s
207ms	600ms	3s	15s	1s	13s	434ms (paused for 1s)	16s	19s	1s	561ms	2s	1s
180ms	593ms	3s	15s	1s	13s	442ms (paused for 1s)	16s	18s	1s	594ms	7s	
171ms	543ms	3s	15s	1s	13s	404ms (paused for 795ms)	15s	19s	1s	28s		

The build is successful.

#### 5.4.2.10 Configure Kubernetes to Verify the Deployment

In this stage, we will verify if the deployment of the Docker image of the application to the Kubernetes. We will call this stage “**Verify the Deployment**”.

```

86         }
87     }
88 }
89 stage('Verify the Deployment') {
90     steps {
91         withKubeConfig(caCertificate: '', clusterName: 'kubernetes', contextName: '', credentialsId: 'k8s')
92             sh "kubectl apply -f deployment-service.yaml"
93     }
94 }
95 }
96 }
97 }
98

```

Use Groovy Sandbox ?

[Pipeline Syntax](#)

[Advanced](#)

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

Now, let us modify the “**step**” in this stage. This “**stage**” will be almost same as the “**Deploy To Kubernetes**”, we just have to change the “**steps**” as follows:

```
sh "kubectl get pods -n webapps"  
sh "kubectl get svc -n webapps"
```

And the complete stage will be as follows:

```
stage('Verify the Deployment') {  
    steps {  
        withKubeConfig(caCertificate: '', clusterName: 'kubernetes', contextName: '',  
        credentialsId: 'k8-cred', namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl:  
        'https://172.31.22.72:6443') {  
            sh "kubectl get pods -n webapps"  
            sh "kubectl get svc -n webapps"  
        }  
    }  
}
```

The screenshot shows the Jenkins Pipeline configuration interface. On the left, there's a sidebar with options: General, Triggers, Pipeline (which is selected), and Advanced. The main area contains Groovy code for a pipeline stage:

```
88  
89 }  
90 stage('Verify the Deployment') {  
91     steps {  
92         withKubeConfig(caCertificate: '', clusterName: 'kubernetes', contextName: '', credentialsId: 'k8-  
93             sh "kubectl get pods -n webapps"  
94             sh "kubectl get svc -n webapps"  
95         }  
96     }  
97 }  
98 }  
99 }  
100
```

Below the code, there's a checkbox labeled "Use Groovy Sandbox". At the bottom, there are "Save" and "Apply" buttons. The status bar at the bottom right indicates "REST API" and "Jenkins 2.528.3".

Save the Pipeline by clicking on “**Save**”

Not secure 54.226.220.79:8080/job/BoardGame/

**BoardGame**

**Last Successful Artifacts**

- database\_service\_project-0.0.5-SNAPSHOT.jar (45.89 MB) [view](#)
- database\_service\_project-0.0.5-SNAPSHOT.pom (3.83 KB) [view](#)

**Test Result Trend**

Passed Skipped Failed

**Build Now**

**Declarative Tool Install**

Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan	Push Docker Image	Deploy To Kubernetes	
193ms	596ms	3s	15s	1s	13s	438ms	16s	19s	1s	577ms	5s	1s
207ms	600ms	3s	15s	1s	13s	434ms (paused for 1s)	16s	19s	1s	561ms	2s	1s
180ms	593ms	3s	15s	1s	13s	442ms (paused for 1s)	16s	18s	1s	594ms	7s	

**SonarQube Quality Gate**

Let us build now. Click on “Build Now”

Not secure 54.226.220.79:8080/job/BoardGame/

**BoardGame**

**Last Successful Artifacts**

- database\_service\_project-0.0.5-SNAPSHOT.jar (45.89 MB) [view](#)
- database\_service\_project-0.0.5-SNAPSHOT.pom (3.83 KB) [view](#)

**Test Result Trend**

Passed Skipped Failed

**Build Now**

**Git Checkout**

Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan	Push Docker Image	Deploy To Kubernetes	Verify the Deployment	
543ms	3s	15s	1s	13s	445ms	16s	19s	1s	585ms	4s	1s	1s
438ms	3s	15s	973ms	13s	461ms (paused for 2s)	15s	19s	1s	600ms	2s	716ms	1s
600ms	3s	15s	1s	13s	434ms (paused for 1s)	16s	19s	1s	561ms	2s	1s	
593ms	3s	15s	1s	13s	442ms (paused for 1s)	16s	18s	1s	594ms	7s		

**SonarQube Quality Gate**

The build is successful

#### 5.4.2.11 Configure the Mail Notifications

In this stage, we will configure the mail notification. In this we need to have port 465 enabled, remember we have already enabled this port.

Go back to “**Manage Jenkins**” on Jenkins browser

The screenshot shows the Jenkins Manage Jenkins interface. At the top, there is a banner about Java 17 end-of-life. Below it, the 'System Configuration' section is visible, containing links for System, Tools, Nodes, Appearance, Managed files, Plugins, Clouds, Security, Credentials, and Credential Providers. An orange arrow points from the text "Click on ‘System’" to the 'System' link in the configuration section.

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

**Java 17 end of life in Jenkins**

You are running Jenkins on Java 17, support for which will end on or after Mar 31, 2026. Refer to [the documentation](#) for more details.

**System Configuration**

- System** Configure global settings and paths.
- Tools** Configure tools, their locations and automatic installers.
- Nodes** Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Docker** Plugin for launching build Agents as Docker containers
- Appearance** Configure the look and feel of Jenkins
- Managed files** e.g. settings.xml for maven, central managed scripts, custom files, ...
- Plugins** Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Clouds** Add, remove, and configure cloud instances to provision agents on-demand.

**Security**

- Security** Secure Jenkins: define who is allowed to access the Jenkins instance.
- Credentials** Configure credentials.
- Credential Providers** Configure the credential providers and tokens.

Click on “**System**”

The screenshot shows the Jenkins System configuration page. It includes fields for the Home directory (set to /var/lib/jenkins), a System Message area, Maven Project Configuration (with fields for Global MAVEN\_OPTS and Local Maven Repository), and a section for # of executors. At the bottom are Save and Apply buttons.

## Scroll down to “Extended Email Notification”

The screenshot shows the Jenkins System configuration page with the Extended E-mail Notification section expanded. It includes fields for SMTP server, SMTP Port (set to 25), and Default user e-mail suffix. Below these are Advanced and Default Content Type sections. At the bottom are Save and Apply buttons. An orange arrow points to the SMTP server input field.

Enter the “**SMTP Server**”. Since we are using gmail, enter the gmail SMTP server that is “**smtp.gmail.com**”

Not secure 54.146.239.233:8080/manage/configure

Jenkins / Manage Jenkins / System

Extended E-mail Notification

SMTP server: smtp.gmail.com

SMTP Port: 25

Default user e-mail suffix: ?

Default Content Type: Plain Text (text/plain)

List ID: ?

**Save** **Apply**

An orange arrow points from the text "For the ‘SMTP Port’, enter ‘465’" to the "SMTP Port" input field, which currently contains the value "25".

For the “SMTP Port”, enter “465”

Not secure 54.146.239.233:8080/manage/configure

Jenkins / Manage Jenkins / System

Extended E-mail Notification

SMTP server: smtp.gmail.com

SMTP Port: 465

Advanced ▾

Default user e-mail suffix: ?

Default Content Type: Plain Text (text/plain)

List ID: ?

**Save** **Apply**

An orange arrow points from the text "Click on the drop down on ‘Advanced’" to the "Advanced" dropdown menu, which is currently expanded.

Click on the drop down on “Advanced”

Jenkins / Manage Jenkins / System

Extended E-mail Notification

SMTP server: smtp.gmail.com

SMTP Port: 465

Advanced ^

Credentials: - none -

+ Add

Use SSL

Use TLS

Use OAuth 2.0

Advanced Email Properties

Save Apply

Click on the drop down on “**Credentials**” and select our Email notification credentials

Jenkins / Manage Jenkins / System

Extended E-mail Notification

SMTP server: smtp.gmail.com

SMTP Port: 465

Advanced ^

Credentials: ebotsidneysmith@gmail.com/\*\*\*\*\* (mail-cred)

+ Add

⚠ For security when using authentication it is recommended to enable either TLS or SSL

Use SSL

Use TLS

Use OAuth 2.0

Advanced Email Properties

Save Apply

Check the box on “**Use SSL**”

Not secure 54.146.239.233:8080/manage/configure

Jenkins / Manage Jenkins / System

SMTP Port: 465

Advanced ^ Edited

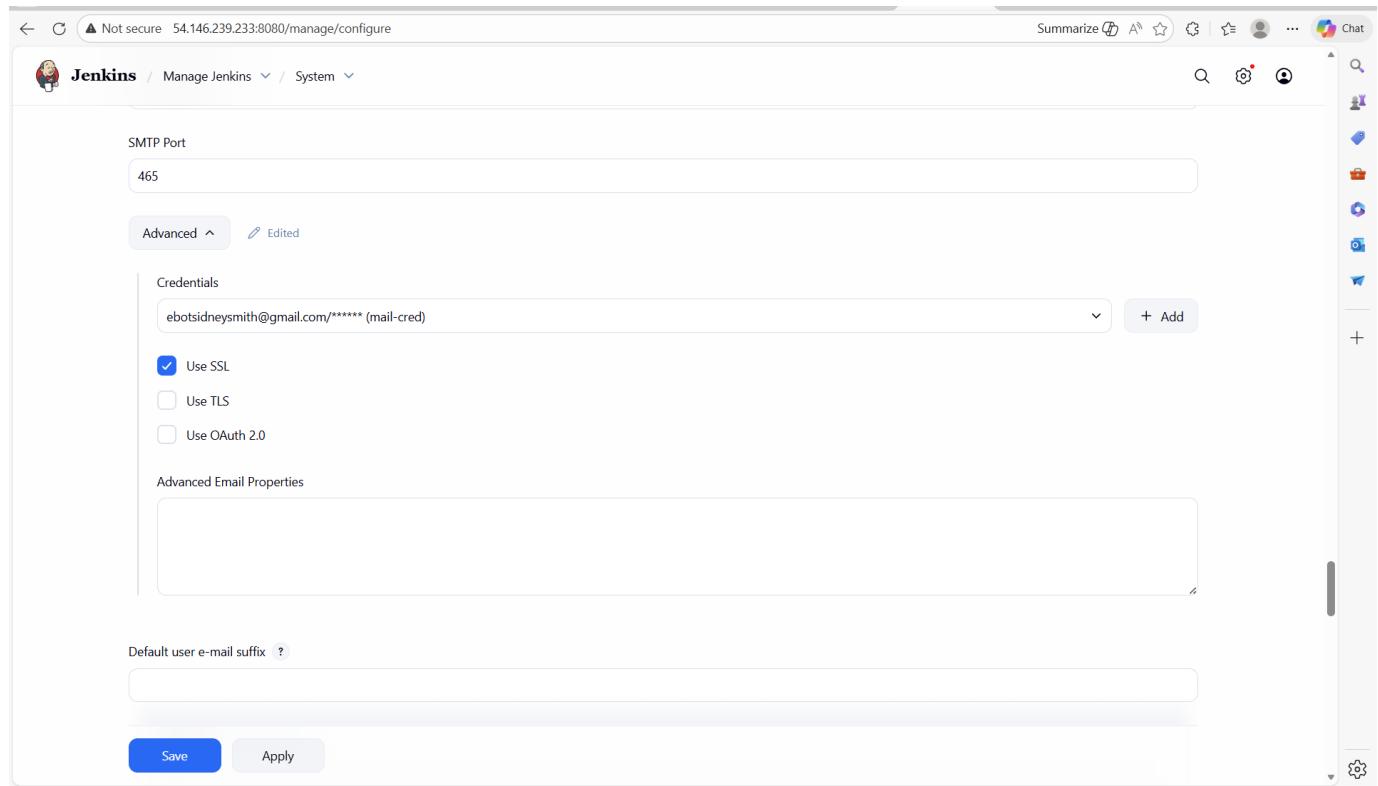
Credentials: ebotsidneysmith@gmail.com/\*\*\*\*\* (mail-cred)

Use SSL  
 Use TLS  
 Use OAuth 2.0

Advanced Email Properties

Default user e-mail suffix: ?

Save Apply



Then, scroll down to “Email Notification”

Not secure 54.146.239.233:8080/manage/configure

Jenkins / Manage Jenkins / System

Require Administrator for Template Testing ?  
 Enable watching for jobs ?  
 Allow sending to unregistered users ?

Default Triggers

Content Token Reference ?

E-mail Notification

SMTP server

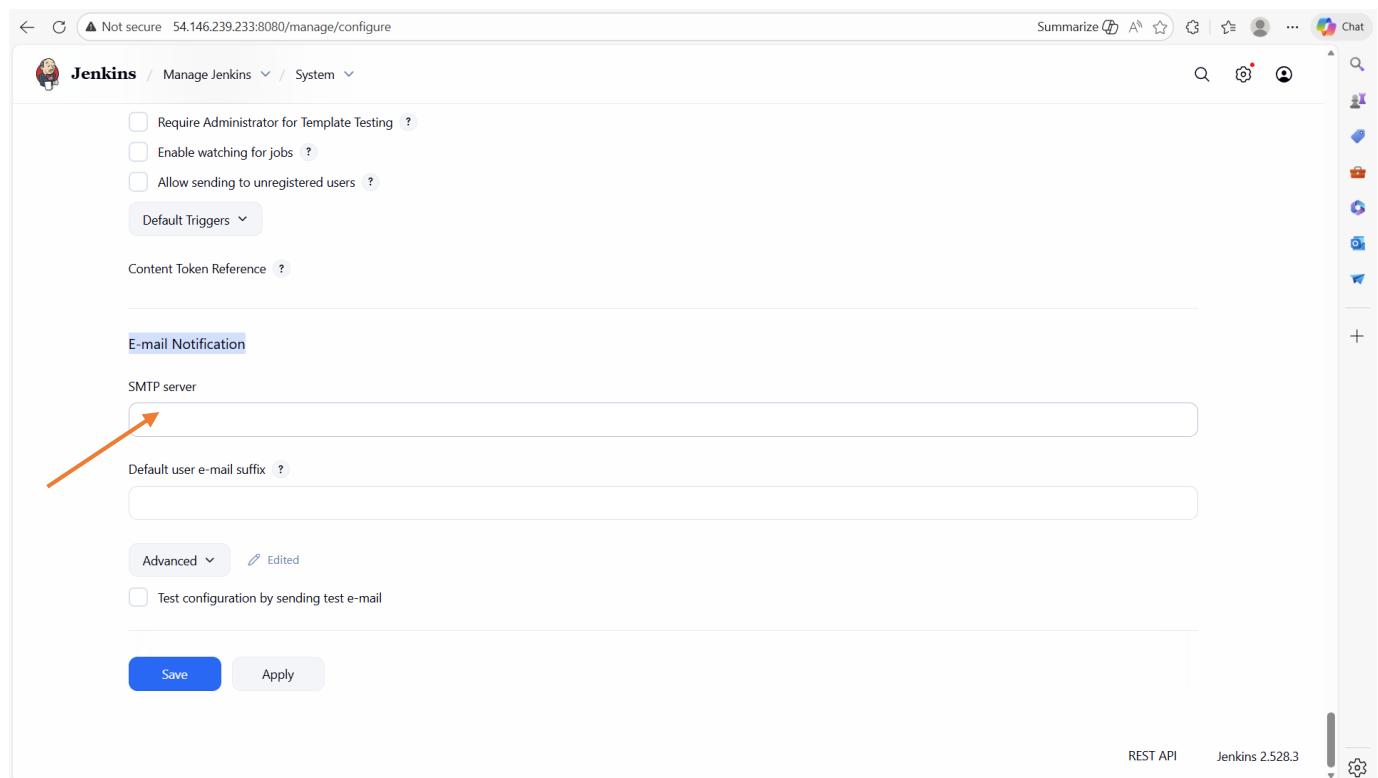
Default user e-mail suffix: ?

Advanced ^ Edited

Test configuration by sending test e-mail

Save Apply

REST API Jenkins 2.528.3



Then, enter the Gmail SMTP Server “**smtp.gmail.com**” again

The screenshot shows the Jenkins System Configuration page under the 'E-mail Notification' section. At the top, there are three checkboxes: 'Require Administrator for Template Testing', 'Enable watching for jobs', and 'Allow sending to unregistered users'. Below them is a 'Default Triggers' dropdown. A 'Content Token Reference' link is also present. The 'E-mail Notification' section includes fields for 'SMTP server' (set to 'smtp.gmail.com') and 'Default user e-mail suffix'. An 'Advanced' dropdown menu is open, showing options like 'Use SMTP Authentication', 'Use SSL', and 'Use TLS'. Two buttons at the bottom are 'Save' (blue) and 'Apply' (grey). A red arrow points from the text 'Click on the drop down on "Advanced"' to the 'Advanced' dropdown menu.

Click on the drop down on “Advanced”

The screenshot shows the same Jenkins System Configuration page, but the 'Advanced' dropdown has been expanded. It now displays additional settings: 'Use SMTP Authentication' (unchecked), 'Use SSL' (unchecked), and 'Use TLS' (unchecked). Below these are fields for 'SMTP Port' (with a question mark icon), 'Reply-To Address' (a text input field), and 'Charset' (a dropdown menu). The 'Save' and 'Apply' buttons remain at the bottom.

Select “Use SMTP Authentication”

Not secure 54.146.239.233:8080/manage/configure

Jenkins / Manage Jenkins / System

Advanced ▾ Edited

Use SMTP Authentication ?

User Name

⚠ For security when using authentication it is recommended to enable either TLS or SSL

Password

Use SSL ?

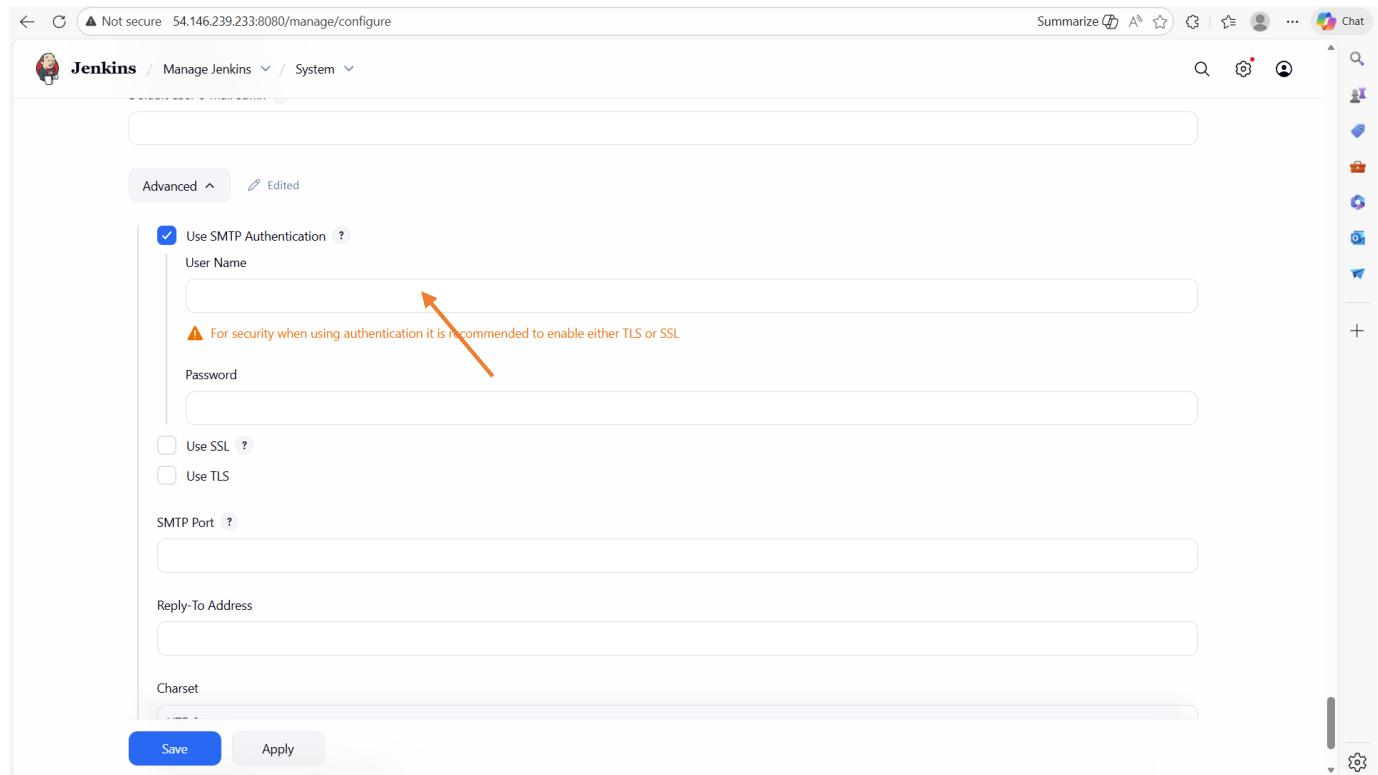
Use TLS

SMTP Port ?

Reply-To Address

Charset

Save Apply



Enter your email address, in my case it is “**ebotsidneysmith@gmail.com**”

Not secure 54.146.239.233:8080/manage/configure

Jenkins / Manage Jenkins / System

Advanced ▾ Edited

Use SMTP Authentication ?

User Name

⚠ For security when using authentication it is recommended to enable either TLS or SSL

Password

Use SSL ?

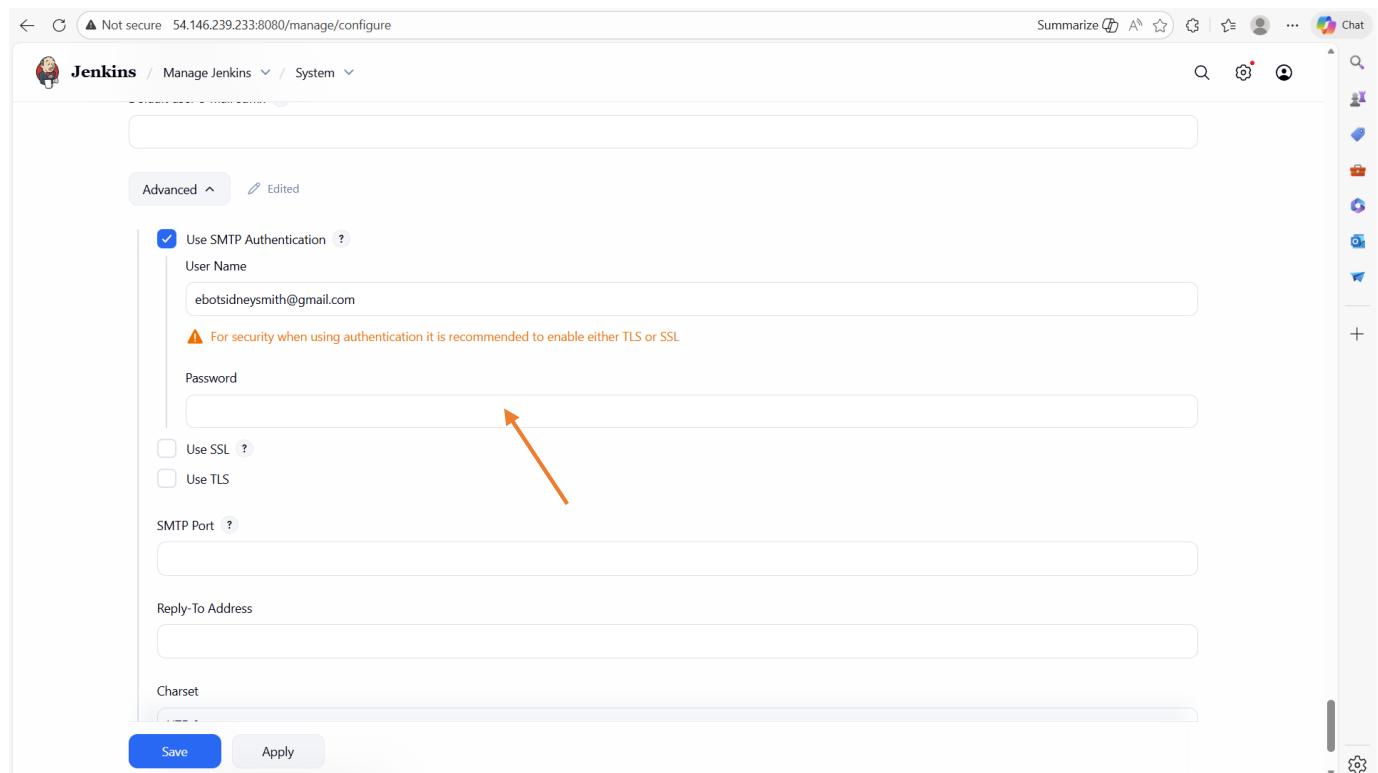
Use TLS

SMTP Port ?

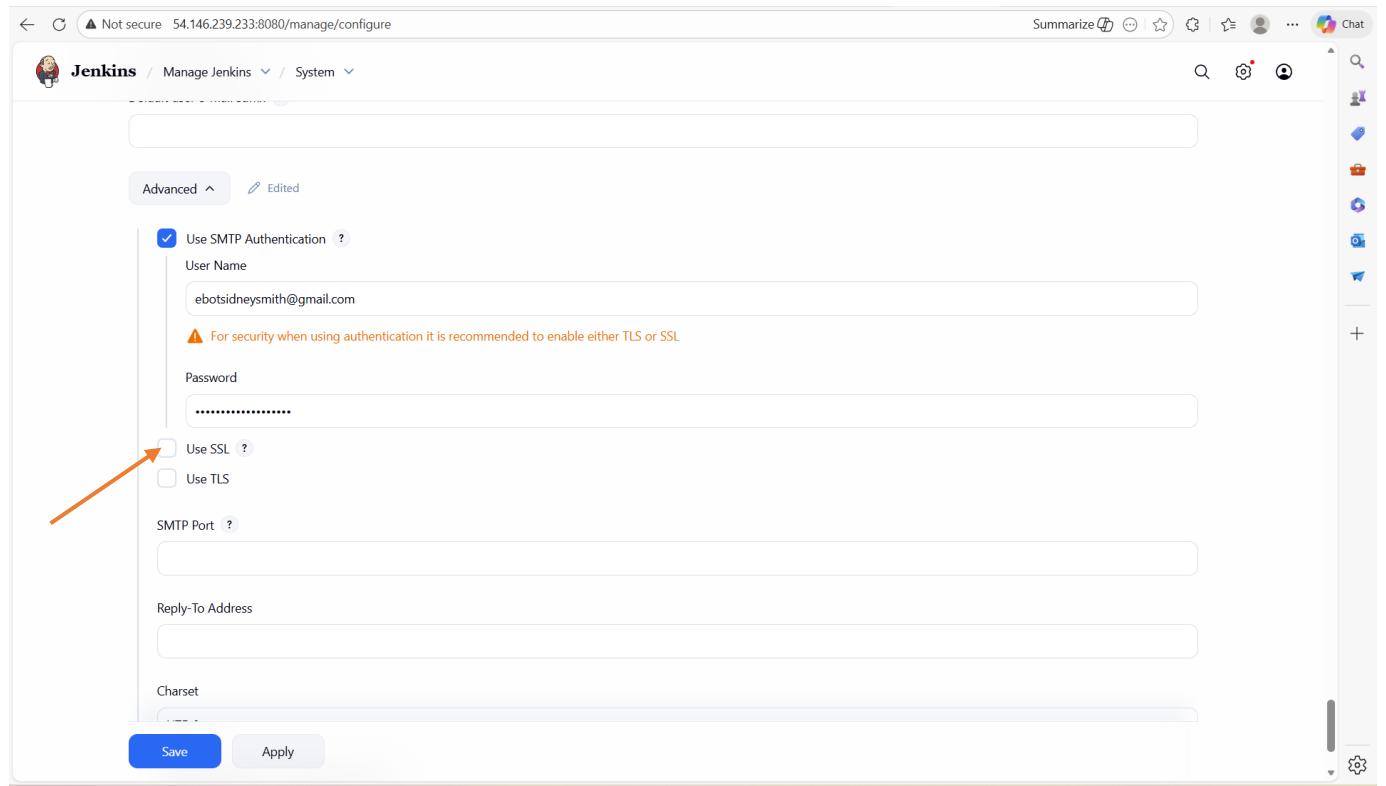
Reply-To Address

Charset

Save Apply



For the password, we will enter the password we generated on Gmail “**thcc ahus girv mvwj**”



Jenkins / Manage Jenkins / System

Advanced ▾ Edited

Use SMTP Authentication ?

User Name  
ebotsidneysmith@gmail.com

⚠ For security when using authentication it is recommended to enable either TLS or SSL

>Password  
.....

Use SSL ? (selected)

Use TLS

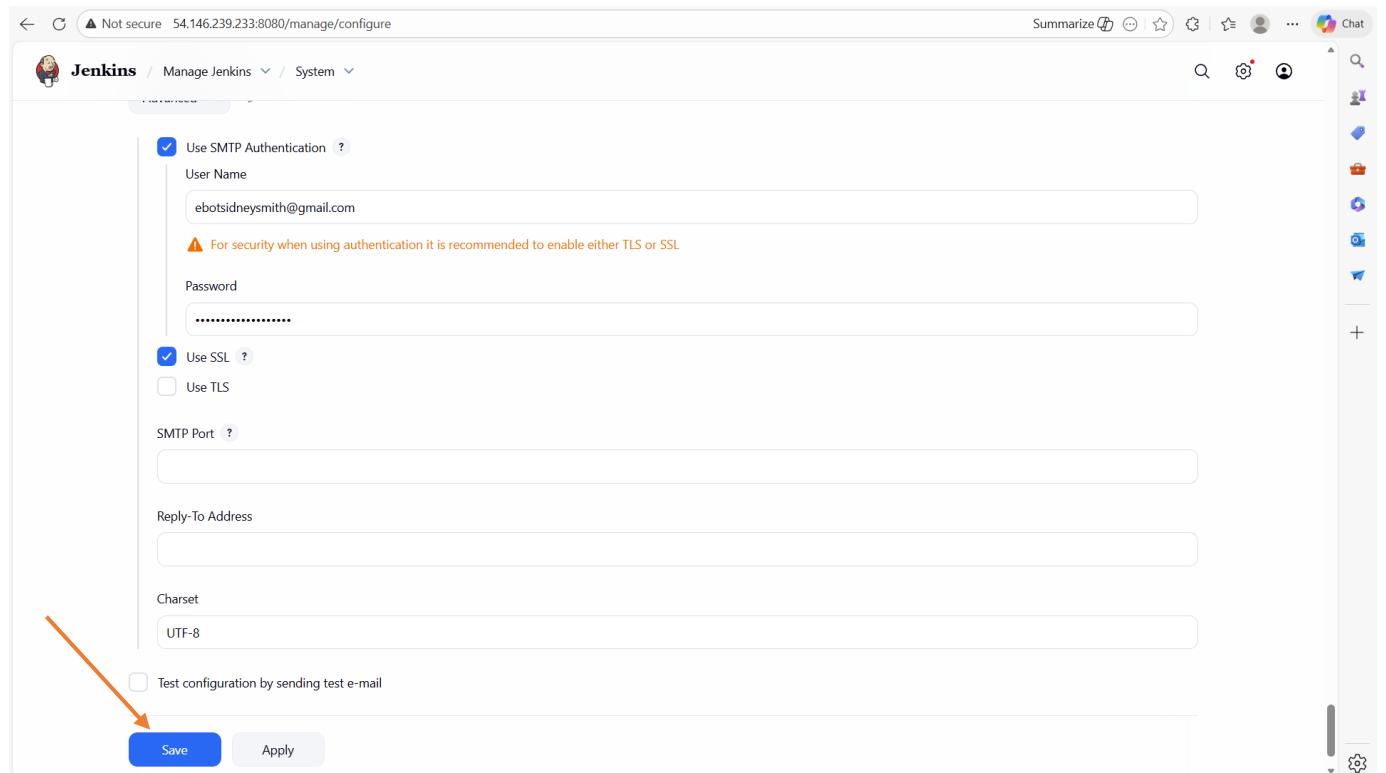
SMTP Port ?  
25

Reply-To Address  
ebotsidneysmith@gmail.com

Charset  
UTF-8

**Save** **Apply**

Select “SSL”



Jenkins / Manage Jenkins / System

Advanced ▾ Edited

Use SMTP Authentication ?

User Name  
ebotsidneysmith@gmail.com

⚠ For security when using authentication it is recommended to enable either TLS or SSL

Password  
.....

Use SSL ? (selected)

Use TLS

SMTP Port ?  
25

Reply-To Address  
ebotsidneysmith@gmail.com

Charset  
UTF-8

Test configuration by sending test e-mail

**Save** **Apply**

Then click on “apply” followed by “Save”

The screenshot shows the Jenkins dashboard at <http://54.146.239.233:8080>. The main area displays a table of build items. One item, 'BoardGame', is listed with the following details:

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	BoardGame	2 hr 9 min #30	N/A	1 min 0 sec

Below the table, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). The right sidebar includes links for 'REST API' and 'Jenkins 2.528.3'.

Let us test the Email notification configuration. Go back to “**Manage Jenkins**” dashboard

The screenshot shows the 'Manage Jenkins' dashboard at <http://54.146.239.233:8080/manage/>. The page features a prominent warning about Java 17 end-of-life:

**Java 17 end of life in Jenkins**  
You are running Jenkins on Java 17, support for which will end on or after Mar 31, 2026. Refer to [the documentation](#) for more details.

Below this, the 'System Configuration' section contains several options:

- System** (highlighted with an orange arrow)
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Appearance**: Configure the look and feel of Jenkins
- Tools**: Configure tools, their locations and automatic installers.
- Docker**: Plugin for launching build Agents as Docker containers
- Managed files**: e.g. settings.xml for maven, central managed scripts, custom files, ...
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Security**: Secure Jenkins: define who is allowed to access the Jenkins instance.
- Credentials**: Configure credentials
- Credential Providers**: Configure the credential providers and how

Click on “**System**”

The screenshot shows the Jenkins System configuration page. Under 'Maven Project Configuration', the 'Global MAVEN\_OPTS' field is empty. The 'Local Maven Repository' dropdown is set to 'Default ("~/m2/repository", or the value of 'localRepository' in Maven's settings file, if defined)'. The '# of executors' field is also empty. At the bottom are 'Save' and 'Apply' buttons.

Scroll down to the end

The screenshot shows the Jenkins System configuration page. Under 'E-mail Notification', the 'SMTP server' field contains 'smtp.gmail.com'. The 'Default user e-mail suffix' field is empty. Below it, there is an 'Advanced' dropdown set to 'Edited' and a checkbox labeled 'Test configuration by sending test e-mail'. A red arrow points to this checkbox. At the bottom are 'Save' and 'Apply' buttons.

Check the box “**Test Configuration by sending test e-mail**”

The screenshot shows the Jenkins System configuration page under the 'E-mail Notification' section. The 'SMTP server' is set to 'smtp.gmail.com'. In the 'Advanced' tab, the 'Test configuration by sending test e-mail' checkbox is checked. An orange arrow points from the text "Enter an email, I will enter ‘ebotsidneysmith@gmail.com’" to the 'Test e-mail recipient' input field, which currently contains an empty placeholder.

Enter an email, I will enter “**ebotsidneysmith@gmail.com**”

The screenshot shows the same Jenkins System configuration page after the email has been entered. The 'Test e-mail recipient' field now contains 'ebotsidneysmith@gmail.com'. A second orange arrow points from the text "Then, click on ‘Test Configuration’" to the 'Test configuration' button, which is highlighted with a light blue background.

Then, click on “**Test Configuration**”

The screenshot shows the Jenkins configuration interface for system settings. Under 'E-mail Notification', the 'SMTP server' is set to 'smtp.gmail.com'. A test configuration has been run, and a success message 'Email was successfully sent' is displayed with an orange arrow pointing to it. The 'Test configuration' button is also visible.

The email has been sent. Let me check my Gmail now.

The screenshot shows a Gmail inbox with one new email. The subject of the email is 'Test email #1 - This is test email #1 sent from Jenkins', with an orange arrow pointing to it. The email is from 'address not configured yet <ebotsidneysmith@gmail.com>'.

You can see the email sent. Open the email

The screenshot shows the same Gmail inbox with the previously received email now open. The subject is 'Test email #1' and the body contains the text 'This is test email #1 sent from Jenkins'. The 'Reply', 'Forward', and 'Delete' buttons are visible at the bottom of the email preview.

You can see the email is sent from Jenkins. So, the email notification configuration is correct. We can now ahead and continue to write the Pipeline.

The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The "Pipeline" tab is selected in the left sidebar. The main area contains a Groovy script editor with line numbers from 87 to 101. An orange arrow points to line 99, which is the closing brace of a stage block. Below the script is a checkbox for "Use Groovy Sandbox". At the bottom are "Save" and "Apply" buttons.

The email notification has to come below “**stages**” and the final code will be as follows:

```

post {
    always {
        script {
            def jobName = env.JOB_NAME
            def buildNumber = env.BUILD_NUMBER
            def pipelineStatus = currentBuild.result ?: 'UNKNOWN'
            def bannerColor = pipelineStatus.toUpperCase() == 'SUCCESS' ? 'green' : 'red'

            def body = """
                <html>
                <body>
                <div style="border: 4px solid ${bannerColor}; padding: 10px;">
                <h2>${jobName} - Build ${buildNumber}</h2>
                <div style="background-color: ${bannerColor}; padding: 10px;">
                <h3 style="color: white;">Pipeline Status: ${pipelineStatus.toUpperCase()}</h3>
                </div>
                <p>Check the <a href="${BUILD_URL}">console output</a>.</p>
                </body>
                </html>
            """

            emailext (
                subject: "${jobName} - Build ${buildNumber} - ${pipelineStatus.toUpperCase()}",
                body: body,
                to: 'ebotsmith@gmail.com',
                from: 'jenkins@example.com',
                replyTo: 'jenkins@example.com',
                mimeType: 'text/html',
                attachmentsPattern: 'trivy-image-report.html'
            )
        }
    }
}

```

The screenshot shows the Jenkins Pipeline configuration page for a job named "BoardGame". The left sidebar has "Pipeline" selected. The main area contains Groovy code for an email notification step:

```

122     subject: "${jobName} - Build ${buildNumber} - ${pipelineStatus.toUpperCase()}",  

123     body: body,  

124     to: 'ebotsmith@gmail.com',  

125     from: 'jenkins@example.com',  

126     replyTo: 'jenkins@example.com',  

127     mimeType: 'text/html',  

128     attachmentsPattern: 'trivy-image-report.html'  

129   )  

130 }  

131 }  

132 }  

133 }

```

Use Groovy Sandbox ?

Below the code, there are "Save" and "Apply" buttons. An orange arrow points from the text "Then, click on ‘apply’ and ‘Save’" to the "Save" button.

REST API Jenkins 2.528.3

Then, click on “apply” and “Save”

The screenshot shows the Jenkins Pipeline execution page for the "BoardGame" job. The left sidebar has "Status" selected. The main area shows the build status as "Last Successful Artifacts" and a "Test Result Trend" chart. Below that is a "Stage View" table showing the duration of various stages for two recent builds (#21 and #20). An orange arrow points from the text "Then build by clicking on ‘Build Now’" to the "Build Now" link in the sidebar.

	Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image
#21	168ms	519ms	3s	15s	1s	13s	447ms	16s	19s	1s
#20	129ms	438ms	3s	15s	973ms	13s	461ms (paused for 2s)	15s	19s	1s

Average stage times: (full run time: ~1min 19s)

Builds

Today

#21 1:12 AM

#20 1:03 AM

SonarQube Quality Gate

Then build by clicking on “Build Now”

The screenshot shows the Jenkins pipeline interface for a project named "BoardGame". On the left, there are navigation links for "Jenkins", "Builds", "Pipeline Syntax", and "Maven". The main area features a summary table with columns for npile, Test, File System Scan, SonarQube Analysis, Quality Gate, Build, Publish To Nexus, Build & Tag Docker Image, Docker Image Scan, Push Docker Image, Deploy To Kubernetes, Verify the Deployment, and Declarative: Post Actions. Below the table, a section titled "SonarQube Quality Gate" displays the status "Passed" for the "BoardGame" project, indicating server-side processing was successful. A "Latest Test Result (no failures)" message is also present. On the right side, there is a vertical sidebar with various Jenkins-related icons.

Let us check the gmail address “[ebotsmith@gmail.com](mailto:ebotsmith@gmail.com)” to see if the email notification has been sent.

The screenshot shows an email in the Gmail inbox from "ebotsidneysmith@gmail.com" with the subject "BoardGame - Build 22 - SUCCESS". The email body contains the message "BoardGame - Build 22" and "Pipeline Status: SUCCESS". It also includes a link to "Check the [console output](#)". Below the message, it says "One attachment · Scanned by Gmail" and shows a thumbnail for "trivy-image-repo...". The inbox sidebar on the left lists categories like Starred, Snoozed, Important, Sent, Drafts, All Mail, Spam, Trash, Purchases, Social, Updates, Forums, Promotions, and More.

You can see that we have received the email. Open the text file

← → ⌂ mail.google.com/mail/u/0/#inbox/5Mfcg2QfBPzCMkhvJcpjMpxZmFmGKgxr?projector=1&messagePartId=0.1

X trivy-image-report.html Search mail

Compose

Inbox

Starred

Snoozed

Important

Sent

Drafts

All Mail

Spam

Trash

Purchases

Social

Updates

Forums

Promotions

More

Labels

Accommodation

AMAZON

Anonimizations

Upgrade

Report Summary

Target	Type	Vulnerabilities	Secrets
ebotsidneysmith/demo:latest (alpine 3.22.2)	alpine	33	-
app/app.jar	jar	81	-

Legend:  
- '-' Not scanned  
- '0' Clean (no security findings detected)

For OSS Maintainers: VEX Notice

If you're an OSS maintainer and Trivy has detected vulnerabilities in your project that you believe are not actually exploitable, consider issuing a VEX (Vulnerability Exploitability Exchange) statement. VEX allows you to communicate the actual status of vulnerabilities in your project, improving security transparency and reducing false positives for your users.

Learn more and start using VEX: <https://trivy.dev/docs/v0.68/guide/supply-chain/vex#publishing-vex-documents>

To disable this notice, set the `TRIVY_DISABLE_VEX_NOTICE` environment variable.

ebotsidneysmith/demo:latest (alpine 3.22.2)  
=====

Total: 33 (UNKNOWN: 0, LOW: 3, MEDIUM: 16, HIGH: 14, CRITICAL: 0)

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
busybox users can launch	CVE-2024-58251	MEDIUM	fixed	1.37.0-r19	1.37.0-r20	In netstat in BusyBox through 1.37.0, local of networ...
						<a href="https://avd.aquasec.com/nvd/cve-2024-58251">https://avd.aquasec.com/nvd/cve-2024-58251</a>
archive can have	CVE-2025-46394	LOW				In tar in BusyBox through 1.37.0, a TAR filenames...

- +

You can see the Trivy scan report.

## 6 Phase 4: Monitoring

In this phase, we are going to set up the monitoring tools to monitor the application. To do this, we have to install Blackbox exporter which is going to help us to monitor the website.

### 6.1 Install Blackbox

Let us install the blackbox. Go to the Prometheus website on <https://prometheus.io/download/>

The screenshot shows the Prometheus Download page. At the top, there is a navigation bar with links for Docs, Download (which is active), Community, Support & Training, and Blog. A search bar is also present. On the right side, there is a sidebar titled "On this page" with a list of exporter names: prometheus, alertmanager, blackbox\_exporter (which has an orange arrow pointing to it), consul\_exporter, graphite\_exporter, memcached\_exporter, mysqld\_exporter, node\_exporter, promlens, pushgateway, and statsd\_exporter. Below the sidebar, there are dropdown menus for "Operating System" (set to "popular") and "Architecture" (set to "popular"). A section for the "prometheus" exporter is shown, featuring a brief description, a "3.9.1 / 2026-01-07 LATEST" button, and a table with columns: File name, OS, Arch, Size, and SHA256 Checksum.

Click on “Blackbox Exporter”

[prometheus.io/download/#blackbox\\_exporter](https://prometheus.io/download/#blackbox_exporter)

Prometheus

## blackbox\_exporter

Blackbox prober exporter

0.28.0 / 2025-12-04 **LATEST**

File name	OS	Arch	Size	SHA256 Checksum
blackbox_exporter-0.28.0.darwin-amd64.tar.gz	darwin	amd64	15.91 MiB	12d7a3010235862d073bb11b997870a50070bcd3b912bca8f0095cfda23c6
blackbox_exporter-0.28.0.darwin-arm64.tar.gz	darwin	arm64	14.98 MiB	ec6c70ccca92e209dd22be76a4fa244f4bd31afda3ddb2bb082144100ec52bb
blackbox_exporter-0.28.0.linux-amd64.tar.gz	linux	amd64	15.40 MiB	cdf5d242fb1cf6d5cb678f3f799f22703d4fafa26b03dcbbd7e1f1825e06329
blackbox_exporter-0.28.0.windows-amd64.zip	windows	amd64	15.62 MiB	f79af4599679d05a4976e78596c8bfed6a088eeef3baced100128921fdee3ac5

[Release notes](#)

## consul\_exporter

Exporter for Consul metrics

0.13.0 / 2024-11-06 **LATEST**

File name	OS	Arch	Size	SHA256 Checksum
consul_exporter-0.13.0.darwin-amd64.tar.gz	darwin	amd64	9.44 MiB	c60739251dc50cbc9bf3fdeeb9e91a46abd50653d7a5df9091836ce02d7f91e0
consul_exporter-0.13.0.darwin-arm64.tar.gz	darwin	arm64	8.99 MiB	7b6d68a2a2222489416b3f1c9c2219956bdded7bcd456808c33c6997854b6920
consul_exporter-0.13.0.linux-amd64.tar.gz	linux	amd64	9.59 MiB	2a8da4147330c6e19c9665deca1c419d507e100de6c8b7c58c0715ff25453773
consul_exporter-0.13.0.windows-amd64.zip	windows	amd64	9.76 MiB	4c378b2827ba2631b4be6551784deef089527f6d6882dd1bc2976d5f58a3d614

[Release notes](#)

## graphite\_exporter

### 6.1.1 Download Blackbox Exporter

Let us download the blackbox exporter

[prometheus.io/download/#blackbox\\_exporter](https://prometheus.io/download/#blackbox_exporter)

Prometheus

## blackbox\_exporter

Blackbox prober exporter

0.28.0 / 2025-12-04 **LATEST**

File name	OS	Arch	Size	SHA256 Checksum
blackbox_exporter-0.28.0.darwin-amd64.tar.gz	darwin	amd64	15.91 MiB	12d7a3010235862d073bb11b997870a50070bcd3b912bca8f0095cfda23c6
blackbox_exporter-0.28.0.darwin-arm64.tar.gz	darwin	arm64	14.98 MiB	ec6c70ccca92e209dd22be76a4fa244f4bd31afda3ddb2bb082144100ec52bb
blackbox_exporter-0.28.0.linux-amd64.tar.gz	linux	amd64	15.40 MiB	cdf5d242fb1cf6d5cb678f3f799f22703d4fafa26b03dcbbd7e1f1825e06329
blackbox_exporter-0.28.0.windows-amd64.zip	windows	amd64	15.62 MiB	f79af4599679d05a4976e78596c8bfed6a088eeef3baced100128921fdee3ac5

[Release notes](#)



## consul\_exporter

Exporter for Consul metrics

0.13.0 / 2024-11-06 **LATEST**

File name	OS	Arch	Size	SHA256 Checksum
consul_exporter-0.13.0.darwin-amd64.tar.gz	darwin	amd64	9.44 MiB	c60739251dc50cbc9bf3fdeeb9e91a46abd50653d7a5df9091836ce02d7f91e0
consul_exporter-0.13.0.darwin-arm64.tar.gz	darwin	arm64	8.99 MiB	7b6d68a2a2222489416b3f1c9c2219956bdded7bcd456808c33c6997854b6920
consul_exporter-0.13.0.linux-amd64.tar.gz	linux	amd64	9.59 MiB	2a8da4147330c6e19c9665deca1c419d507e100de6c8b7c58c0715ff25453773
consul_exporter-0.13.0.windows-amd64.zip	windows	amd64	9.76 MiB	4c378b2827ba2631b4be6551784deef089527f6d6882dd1bc2976d5f58a3d614

[Release notes](#)

## graphite\_exporter

Right-click on “**Blackbox\_exporter**”

https://github.com/prometheus/blackbox\_exporter/releases/download/v0.28.0/blackbox\_exporter-0.28.0.linux-amd64.tar.gz

Select “Copy link address”

Then run the command on the Monitor server to download it:

```
wget https://github.com/prometheus/blackbox_exporter/releases/download/v0.28.0/blackbox_exporter-0.28.0.linux-amd64.tar.gz
```

```
root@ip-172-31-18-51:~# sudo dpkg -i grafana-enterprise_12.3.1_20271043721_linux_amd64.deb
Selecting previously unselected package grafana-enterprise.
(Reading database ... 71886 files and directories currently installed.)
Preparing to unpack grafana-enterprise_12.3.1_20271043721_linux_amd64.deb ...
Unpacking grafana-enterprise (12.3.1) ...
Setting up grafana-enterprise (12.3.1) ...
info: Selecting UID from range 100 to 999 ...
info: Adding system user 'grafana' (UID 111) ...
info: Adding new user 'grafana' (UID 111) with group 'grafana' ...
info: Not creating home directory '/usr/share/grafana'.
### NOT starting on installation, please execute the following statements to configure grafana to start automatically using systemd
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable grafana-server
# # On start grafana-server by executing
sudo /bin/systemctl start grafana-server
root@ip-172-31-18-51:~# C
root@ip-172-31-18-51:~# sudo /bin/systemctl start grafana-server
root@ip-172-31-18-51:~# wget https://github.com/prometheus/blackbox_exporter/releases/download/v0.28.0/blackbox_exporter-0.28.0.linux-amd64.tar.gz
--2026-01-11 00:02:39 - https://github.com/prometheus/blackbox_exporter/releases/download/v0.28.0/blackbox_exporter-0.28.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 149.82.113.4
Connecting to github.com (github.com)|149.82.113.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://release-assets.githubusercontent.com/github-production-release-asset/41964498/d1e23250-73a8-4d26-abc6-ed04d2507545?perCommit=2018-11-08&refId=571-4301-aeddb-ab1947aa7ad08sk1t#398a6654-997b-47e9-b12b-9515b8964d6esk#2026-01-10T23:35:53Z&size=2026-01-1109:34:5633a0176sk5sk5kv2018-11-09:5633geyprV
LuyWnviSPLPhnD0tWkhAP10Xma1m5w904tB3D5jvtseYXo1iJKV10iLChbhc01JUj1NjI9_eYJpc3M0lJnaoRodTu29t1iwYXVkijoicmVs2WFz2WFcz2v0CH1vHvjdgVb15b91LmNcmmlud21uZg9jcy5uZ01f0_zKzhPm0ds
6n_Cj40z3SyvKkA6zfjap0945Hsdoq87w&response-content-disposition=attachment%3Bfilename%3Dblackbox_exporter-0.28.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2026-01-11 00:02:39 - https://release-assets.githubusercontent.com/github-production-release-asset/41964498/d1e23250-73a8-4d26-abc6-ed04d2507545?perCommit=2018-11-08&refId=571-4301-aeddb-ab1947aa7ad08sk1t#398a6654-997b-47e9-b12b-9515b8964d6esk#2026-01-10T23:35:53Z&size=2026-01-1109:34:5633a0176sk5sk5kv2018
8aWV_0003geyprV
Resolving release-assets.githubusercontent.com (release-assets.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16148437 (15M) [application/octet-stream]
Saving to: 'blackbox_exporter-0.28.0.linux-amd64.tar.gz'

blackbox_exporter-0.28.0.linux-amd64.tar 100%[=====] 15.40M -->KB/s in 0.1s
2026-01-11 00:02:39 (114 MB/s) - 'blackbox_exporter-0.28.0.linux-amd64.tar.gz' saved [16148437/16148437]

root@ip-172-31-18-51:~#
```

Then, run the command to extract it:

```
tar -xvf blackbox_exporter-0.28.0.linux-amd64.tar.gz
```

The screenshot shows a MobaXterm window titled "Monitor". The terminal session displays the command "tar -xvf blackbox\_exporter-0.28.0.linux-amd64.tar.gz" being run by root at IP 172.31.18.51. The output shows the extraction of files from the tar archive, including "blackbox\_exporter-0.28.0.linux-amd64", "grafana-enterprise\_12.3.1\_20271043721\_linux\_amd64.deb", and "prometheus-3.9.1.linux-amd64". The session also lists other user sessions like Jenkins, Master, Nexus, Slave1, Slave2, and SonarQube.

```
root@ip-172-31-18-51:~# tar -xvf blackbox_exporter-0.28.0.linux-amd64.tar.gz
blackbox_exporter-0.28.0.linux-amd64/
blackbox_exporter-0.28.0.linux-amd64/LICENSE
blackbox_exporter-0.28.0.linux-amd64/.NOTICE
blackbox_exporter-0.28.0.linux-amd64/blackbox.yaml
blackbox_exporter-0.28.0.linux-amd64/blackbox_exporter
root@ip-172-31-18-51:~#
```

Then, run the command to see the content:

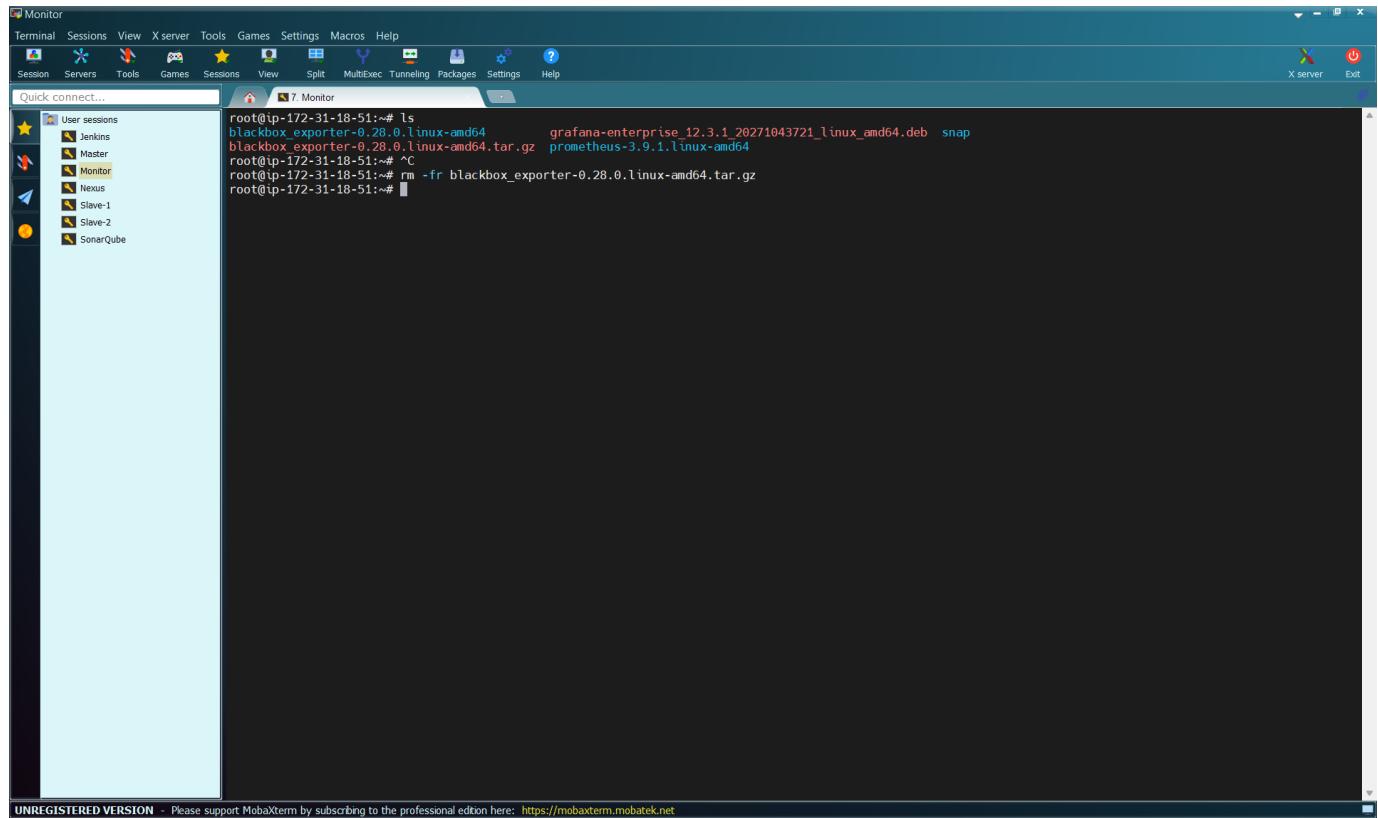
```
ls
```

The screenshot shows the same MobaXterm window after the tar file has been extracted. The terminal command "ls" is run, and the output shows the contents of the "blackbox\_exporter-0.28.0.linux-amd64" directory, which includes "blackbox\_exporter-0.28.0.linux-amd64", "grafana-enterprise\_12.3.1\_20271043721\_linux\_amd64.deb", and "prometheus-3.9.1.linux-amd64". An orange arrow points from the text "You can see we still have the tar file." in the previous step to the "blackbox\_exporter-0.28.0.linux-amd64" entry in the ls output.

```
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64
blackbox_exporter-0.28.0.linux-amd64.tar.gz
root@ip-172-31-18-51:~#
```

You can see we still have the tar file. Let us remove it by running the command:

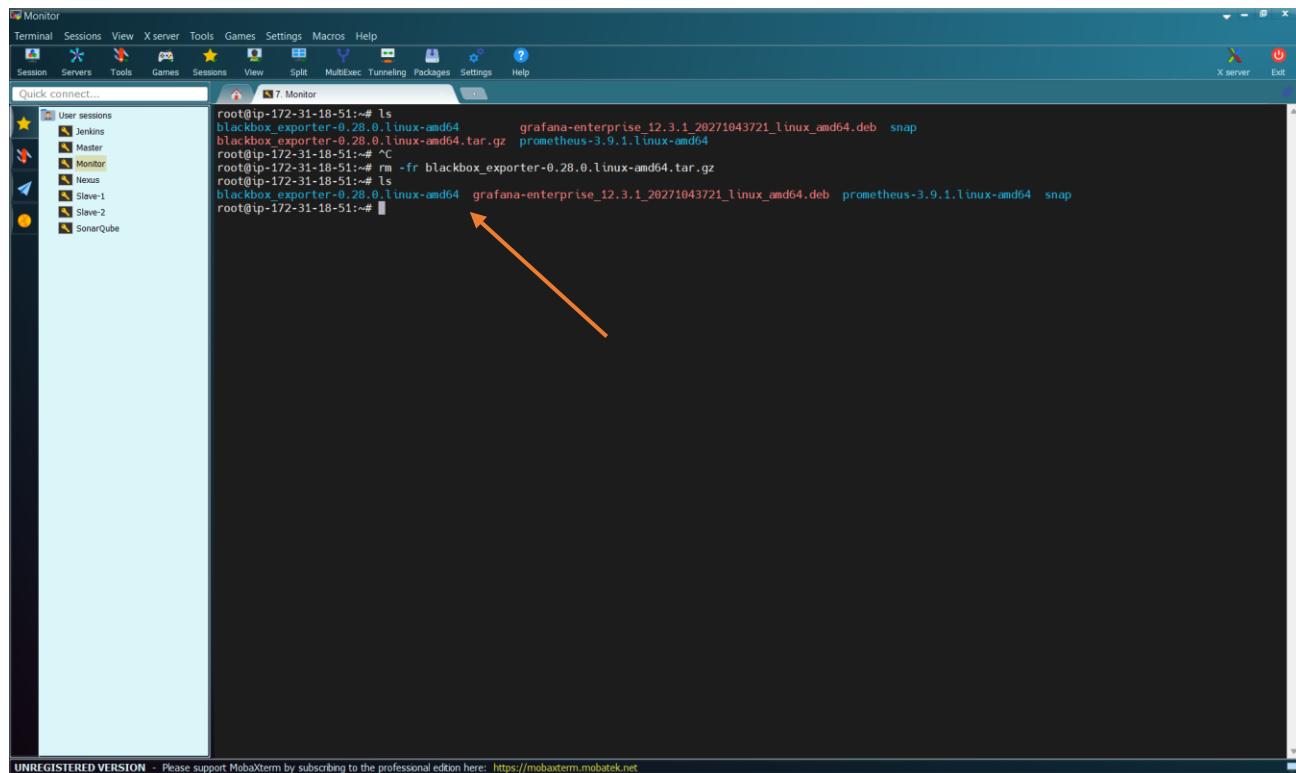
```
rm -fr blackbox_exporter-0.28.0.linux-amd64.tar.gz
```



```
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  snap
blackbox_exporter-0.28.0.linux-amd64.tar.gz  prometheus-3.9.1.linux-amd64
root@ip-172-31-18-51:~# ^C
root@ip-172-31-18-51:~# rm -fr blackbox_exporter-0.28.0.linux-amd64.tar.gz
root@ip-172-31-18-51:~#
```

Run the command to see the content again:

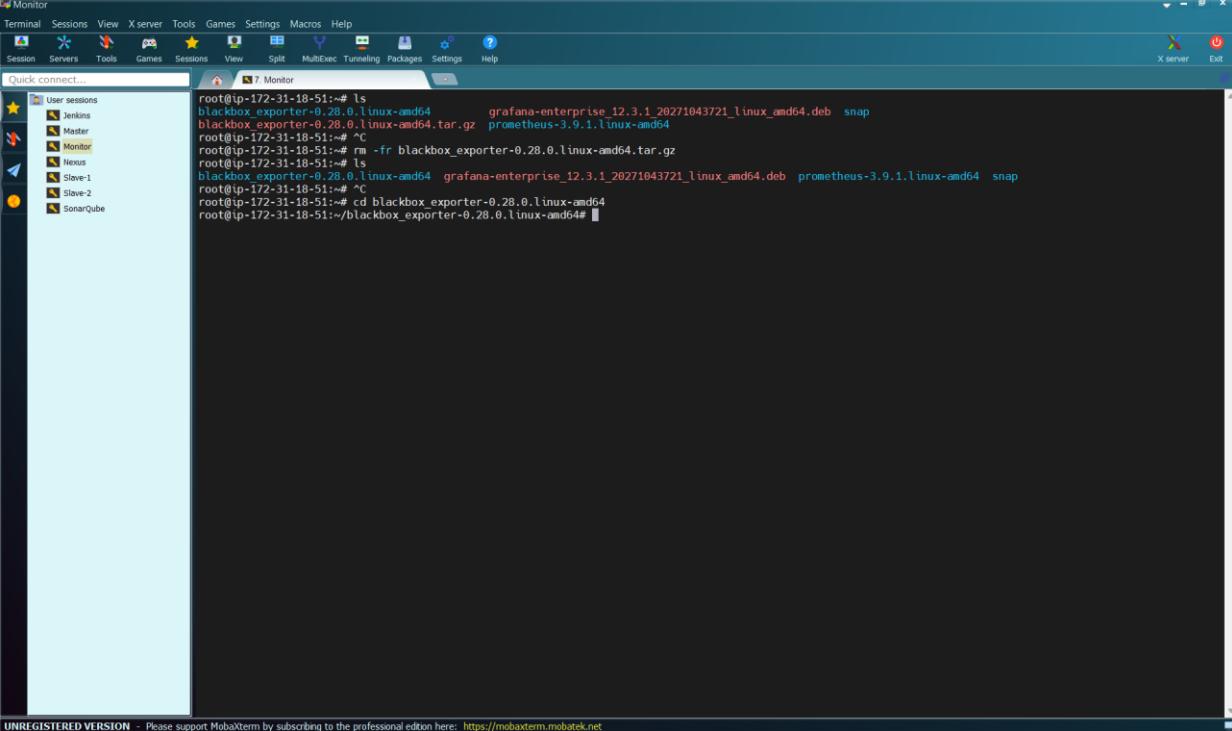
```
ls
```



```
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  snap
blackbox_exporter-0.28.0.linux-amd64.tar.gz  prometheus-3.9.1.linux-amd64
root@ip-172-31-18-51:~# ^C
root@ip-172-31-18-51:~# rm -fr blackbox_exporter-0.28.0.linux-amd64.tar.gz
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  prometheus-3.9.1.linux-amd64  snap
root@ip-172-31-18-51:~#
```

Then, let us run the command to go into the blackbox exporter folder:

```
cd blackbox_exporter-0.28.0.linux-amd64
```

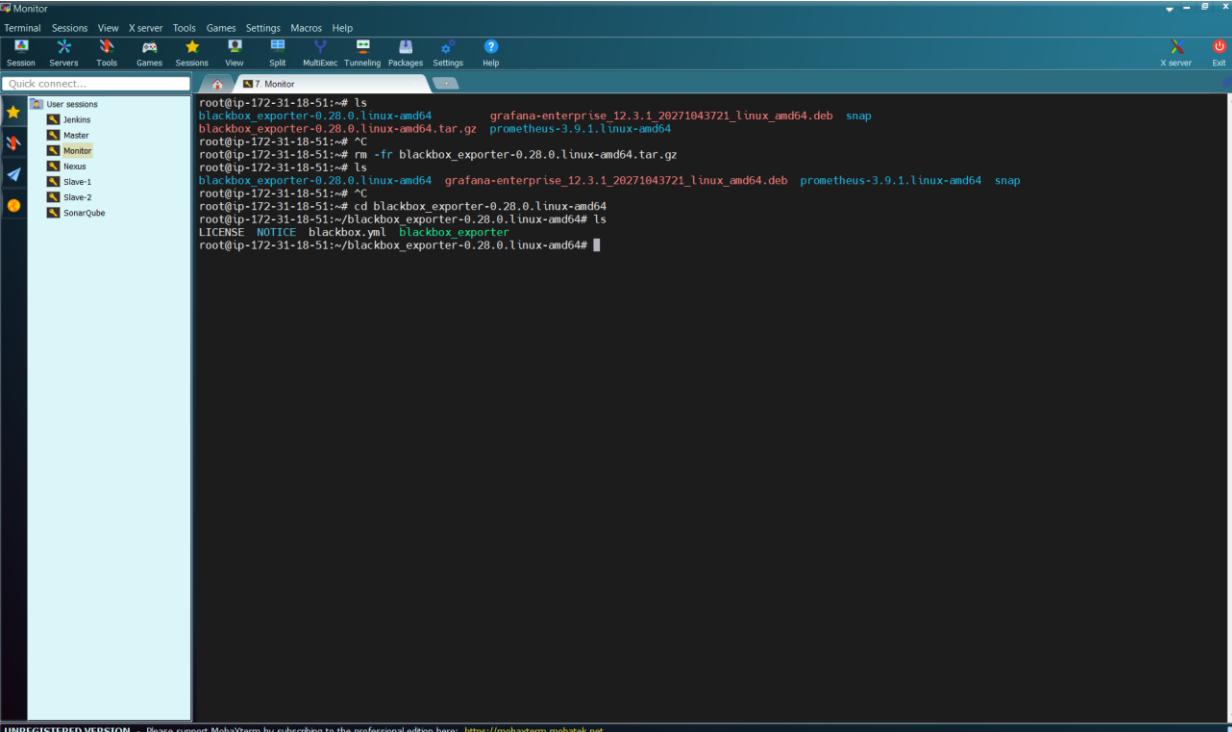


The screenshot shows a MobaXterm window titled "Monitor". The terminal session is connected to a root shell on a Linux system. The user has run the command "cd blackbox\_exporter-0.28.0.linux-amd64" to change into the specified directory. The terminal output shows the directory listing and the command being run.

```
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64      grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  snap
blackbox_exporter-0.28.0.linux-amd64.tar.gz  prometheus-3.9.1.linux-amd64
root@ip-172-31-18-51:~# ^C
root@ip-172-31-18-51:~# rm -fr blackbox_exporter-0.28.0.linux-amd64.tar.gz
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  prometheus-3.9.1.linux-amd64  snap
root@ip-172-31-18-51:~# ^C
root@ip-172-31-18-51:~# cd blackbox_exporter-0.28.0.linux-amd64
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64#
```

Then, run the command to see the content of the folder:

```
ls
```

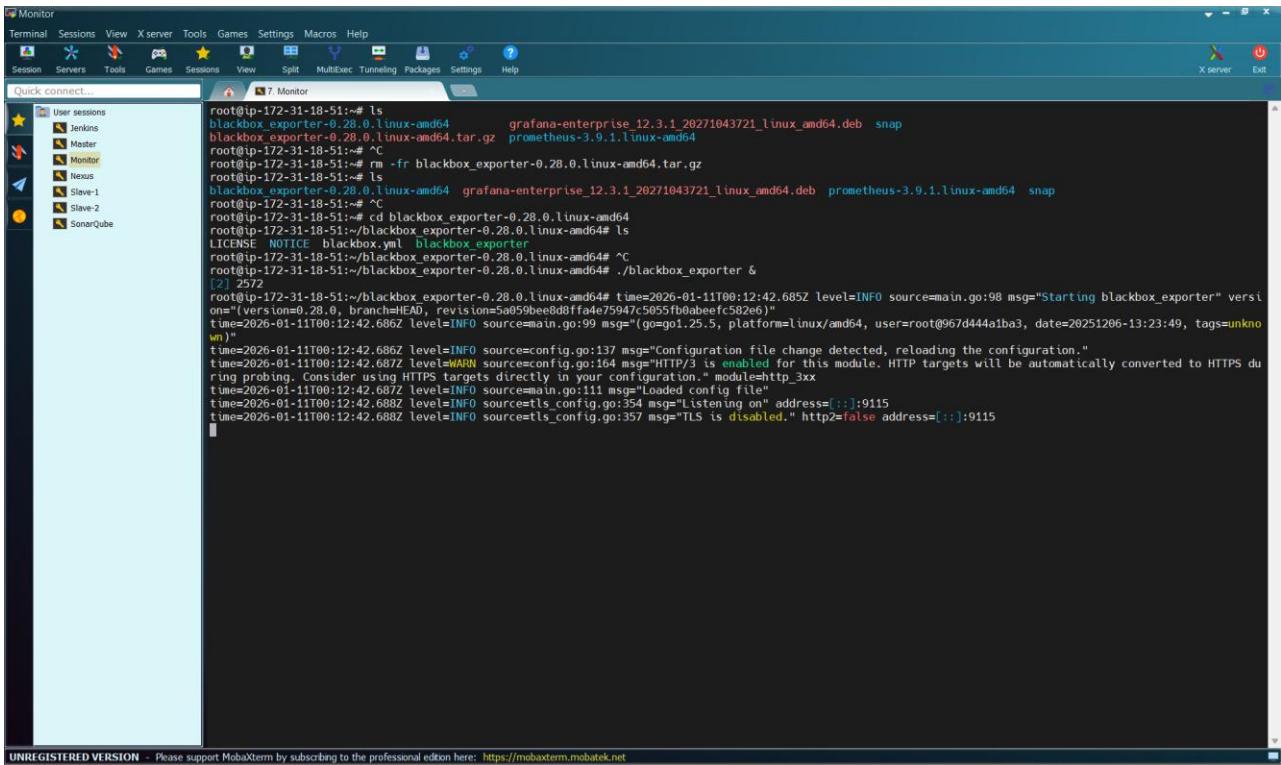


The screenshot shows a MobaXterm window titled "Monitor". The terminal session is connected to a root shell on a Linux system. The user has run the command "ls" to list the contents of the current directory, which is the "blackbox\_exporter-0.28.0.linux-amd64" folder. The terminal output shows the files and subdirectories present in the folder.

```
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ls
blackbox_exporter-0.28.0.linux-amd64      grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  snap
blackbox_exporter-0.28.0.linux-amd64.tar.gz  prometheus-3.9.1.linux-amd64
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ^C
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# rm -fr blackbox_exporter-0.28.0.linux-amd64.tar.gz
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  prometheus-3.9.1.linux-amd64  snap
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ^C
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ls
LICENSE  NOTICE  blackbox.yml  blackbox_exporter
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64#
```

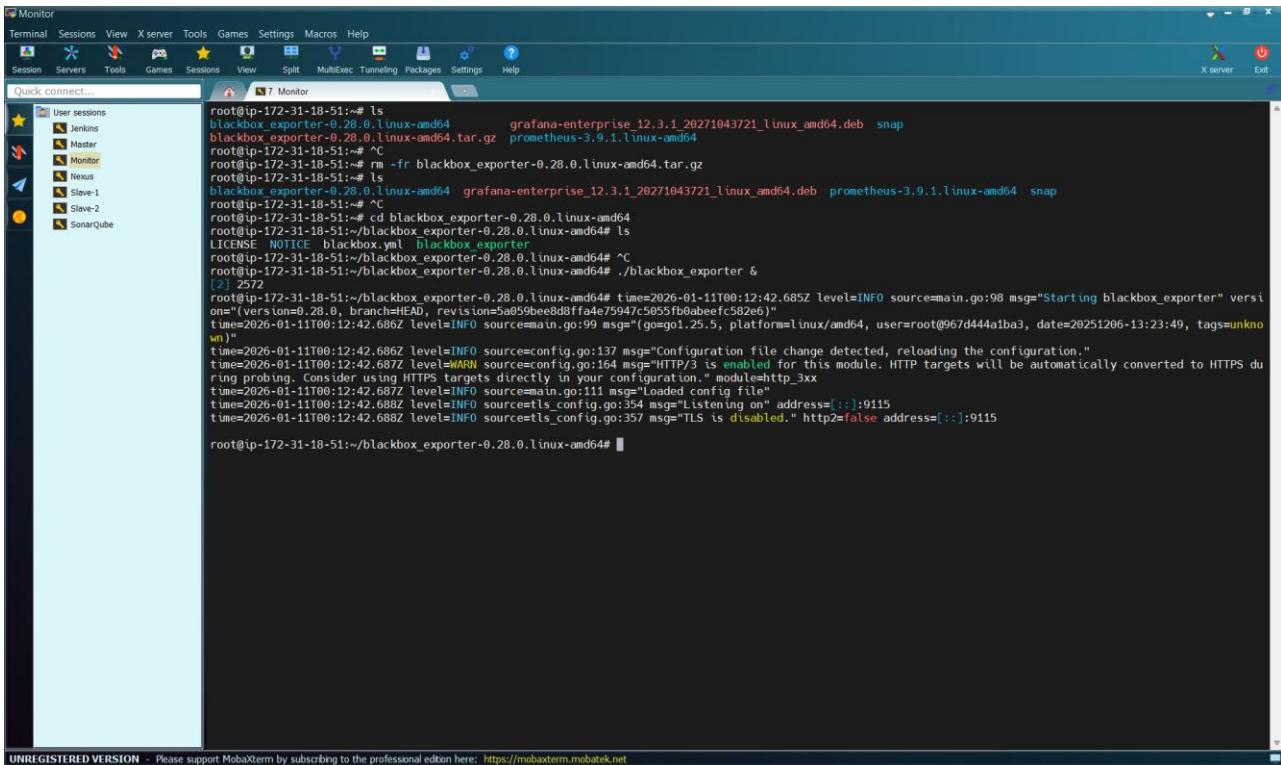
Then, let us run the executable file in the background using the command:

```
./blackbox_exporter &
```



```
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  snap
blackbox_exporter-0.28.0.linux-amd64.tar.gz  prometheus-3.9.1.linux-amd64
root@ip-172-31-18-51:~# ^C
root@ip-172-31-18-51:~# rm -fr blackbox_exporter-0.28.0.linux-amd64.tar.gz
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  prometheus-3.9.1.linux-amd64  snap
root@ip-172-31-18-51:~# ^C
root@ip-172-31-18-51:~# cd blackbox_exporter-0.28.0.linux-amd64
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ls
LICENSE  NOTICE  blackbox.yml  blackbox_exporter
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ^C
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ./blackbox_exporter &
[2] 2572
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# time=2026-01-11T00:12:42.685Z level=INFO source=main.go:98 msg="Starting blackbox_exporter" version="version=0.28.0, branch=HEAD, revision=5a059beed8dff4e75947c5055fb0abefc582e6"
time=2026-01-11T00:12:42.687Z level=INFO source=config.go:164 msg="HTTP/3 is enabled for this module. HTTP targets will be automatically converted to HTTPS during probing. Consider using HTTPS targets directly in your configuration." module=http_3xx
time=2026-01-11T00:12:42.687Z level=INFO source=main.go:111 msg="Loaded config file"
time=2026-01-11T00:12:42.688Z level=INFO source=tls_config.go:354 msg="Listening on address=[::]:9115"
time=2026-01-11T00:12:42.688Z level=INFO source=tls_config.go:357 msg="TLS is disabled." http=false address=[::]:9115
```

Press “Enter”



```
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  snap
blackbox_exporter-0.28.0.linux-amd64.tar.gz  prometheus-3.9.1.linux-amd64
root@ip-172-31-18-51:~# ^C
root@ip-172-31-18-51:~# rm -fr blackbox_exporter-0.28.0.linux-amd64.tar.gz
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  prometheus-3.9.1.linux-amd64  snap
root@ip-172-31-18-51:~# ^C
root@ip-172-31-18-51:~# cd blackbox_exporter-0.28.0.linux-amd64
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ls
LICENSE  NOTICE  blackbox.yml  blackbox_exporter
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ^C
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ./blackbox_exporter &
[2] 2572
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# time=2026-01-11T00:12:42.685Z level=INFO source=main.go:98 msg="Starting blackbox_exporter" version="version=0.28.0, branch=HEAD, revision=5a059beed8dff4e75947c5055fb0abefc582e6"
time=2026-01-11T00:12:42.687Z level=INFO source=config.go:137 msg="Configuration file change detected, reloading the configuration."
time=2026-01-11T00:12:42.687Z level=INFO source=config.go:164 msg="HTTP/3 is enabled for this module. HTTP targets will be automatically converted to HTTPS during probing. Consider using HTTPS targets directly in your configuration." module=http_3xx
time=2026-01-11T00:12:42.687Z level=INFO source=main.go:111 msg="Loaded config file"
time=2026-01-11T00:12:42.688Z level=INFO source=tls_config.go:354 msg="Listening on address=[::]:9115"
time=2026-01-11T00:12:42.688Z level=INFO source=tls_config.go:357 msg="TLS is disabled." http=false address=[::]:9115
```

This has started running in the background an by default it is running on port 9115.

### 6.1.2 Access Blackbox Exporter on browser

Let us now access blackbox exporter in the browser. It is running on port 9115. We will do this as follows:

`http://<Public IP address of Monitor server>:9115`

That is

`http://44.222.224.185:9115`

The screenshot shows a web browser window with the following details:

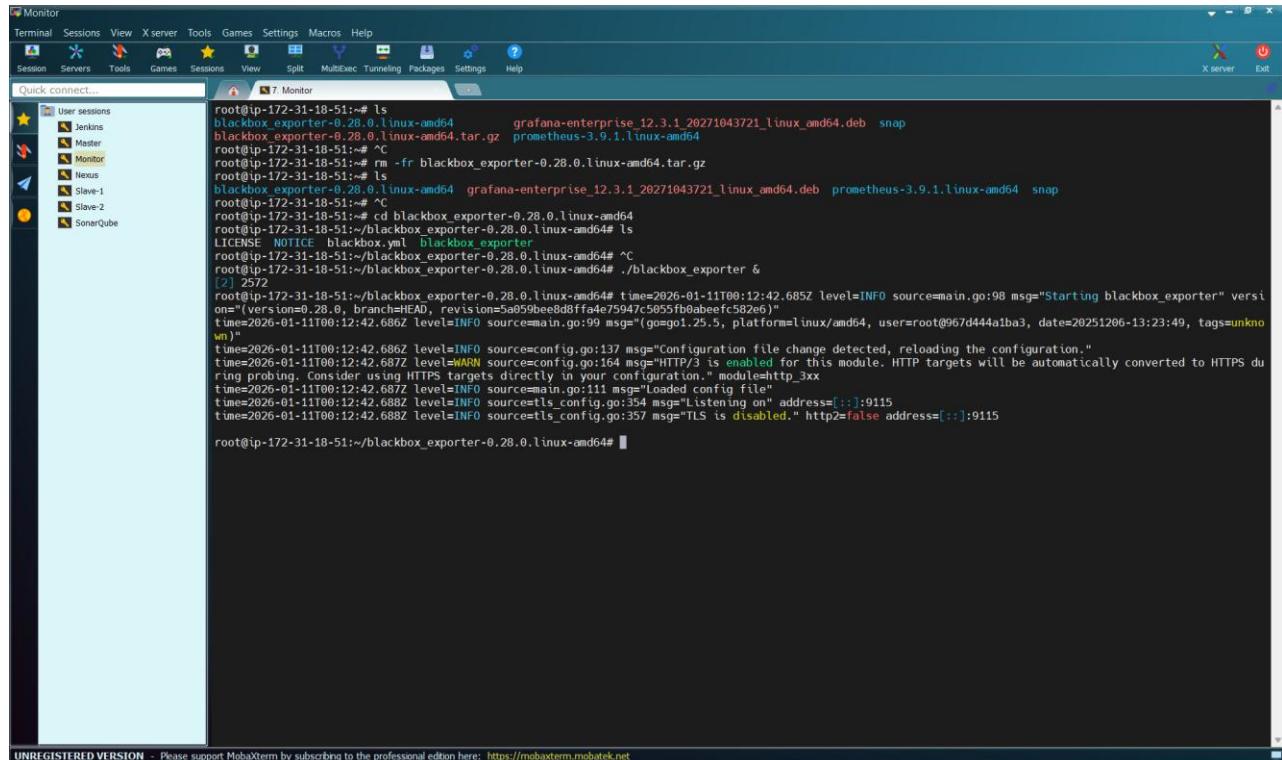
- Title bar: "Blackbox Exporter" (highlighted in red)
- Address bar: "Not secure 44.220.145.88:9115" (highlighted in red)
- Links:
  - [Probe probe.prometheus.io for http\\_2xx](#)
  - [Debug probe.prometheus.io for http\\_2xx](#)
  - [Metrics](#)
  - [Configuration](#)
- Section: "Recent Probes"
- Table:

Module	Target	Result	Debug
probe	probe.prometheus.io	OK	OK
- Right sidebar: Includes icons for search, user, file, folder, database, gear, and a plus sign.

You can see that it is running.

## 6.2 Configure Prometheus

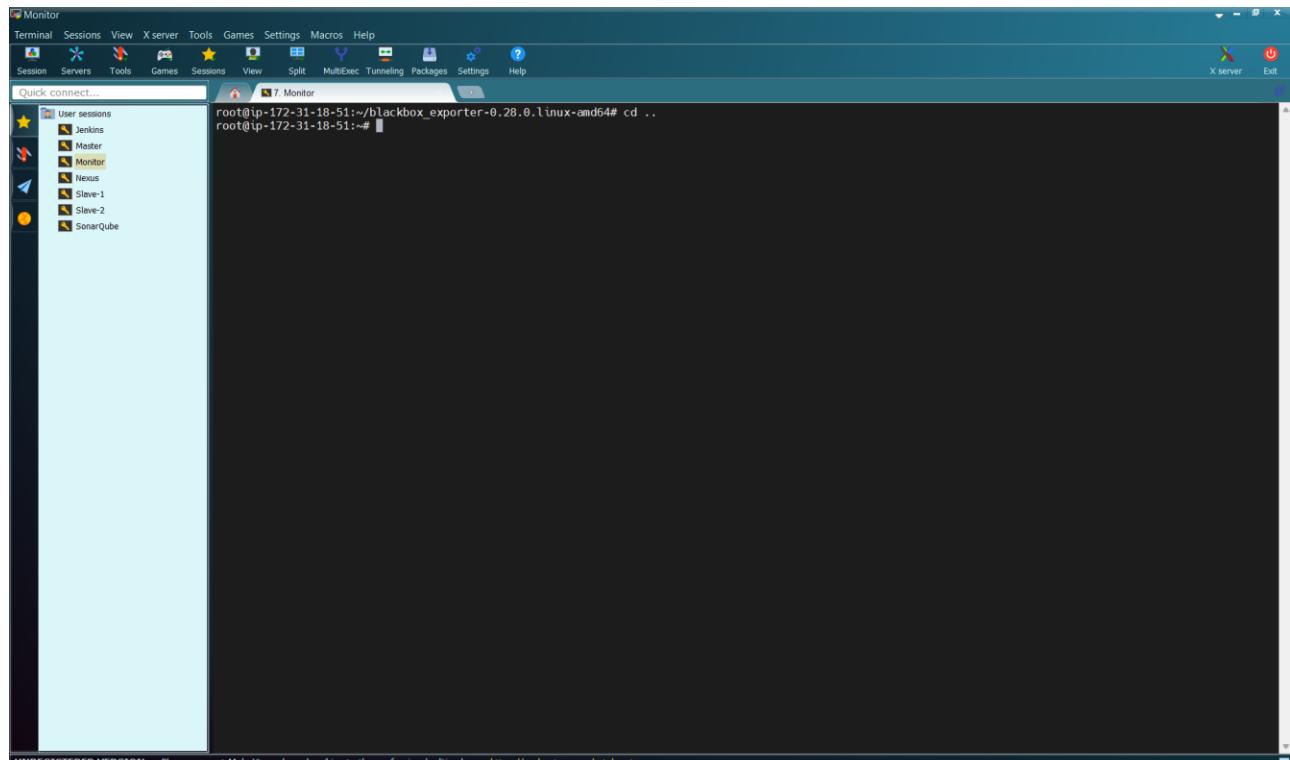
We have to configure prometheus. To do this, we will make some changes in the file “**prometheus.yaml**”. Let us find the file



```
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  snap
blackbox_exporter-0.28.0.linux-amd64.tar.gz  prometheus-3.9.1.linux-amd64
root@ip-172-31-18-51:~# rm -fr blackbox_exporter-0.28.0.linux-amd64.tar.gz
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  prometheus-3.9.1.linux-amd64  snap
root@ip-172-31-18-51:~# cd blackbox_exporter-0.28.0.linux-amd64
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ls
LICENSE  NOTICE  blackbox.yaml  Blackbox_exporter
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ^C
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# ./blackbox_exporter &
[2] 2572
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# time=2026-01-11T00:12:42.685Z level=INFO source=main.go:98 msg="Starting blackbox_exporter" version="(version=0.28.0, branch=HEAD, revision=5a059be8edff4a4e75947c5055fb0a0beefc582e6)"
time=2026-01-11T00:12:42.686Z level=INFO source=main.go:99 msg="(goos=linux, goarch=amd64, user=root@967d444a1ba3, date=20251206-13:23:49, tags=unknown)"
time=2026-01-11T00:12:42.686Z level=INFO source=config.go:137 msg="Configuration file change detected, reloading the configuration."
time=2026-01-11T00:12:42.687Z level=WARNING source=config.go:164 msg="HTTP/3 is enabled for this module. HTTP targets will be automatically converted to HTTPS during proxying. Consider using HTTPS targets directly in your configuration." module=http_3x
time=2026-01-11T00:12:42.687Z level=INFO source=main.go:111 msg="Loaded config file"
time=2026-01-11T00:12:42.688Z level=INFO source=tls_config.go:354 msg="Listening on " address=:9115
time=2026-01-11T00:12:42.688Z level=INFO source=tls_config.go:357 msg="TLS is disabled." http2=false address=:9115
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64#
```

Run the command to move one step backward:

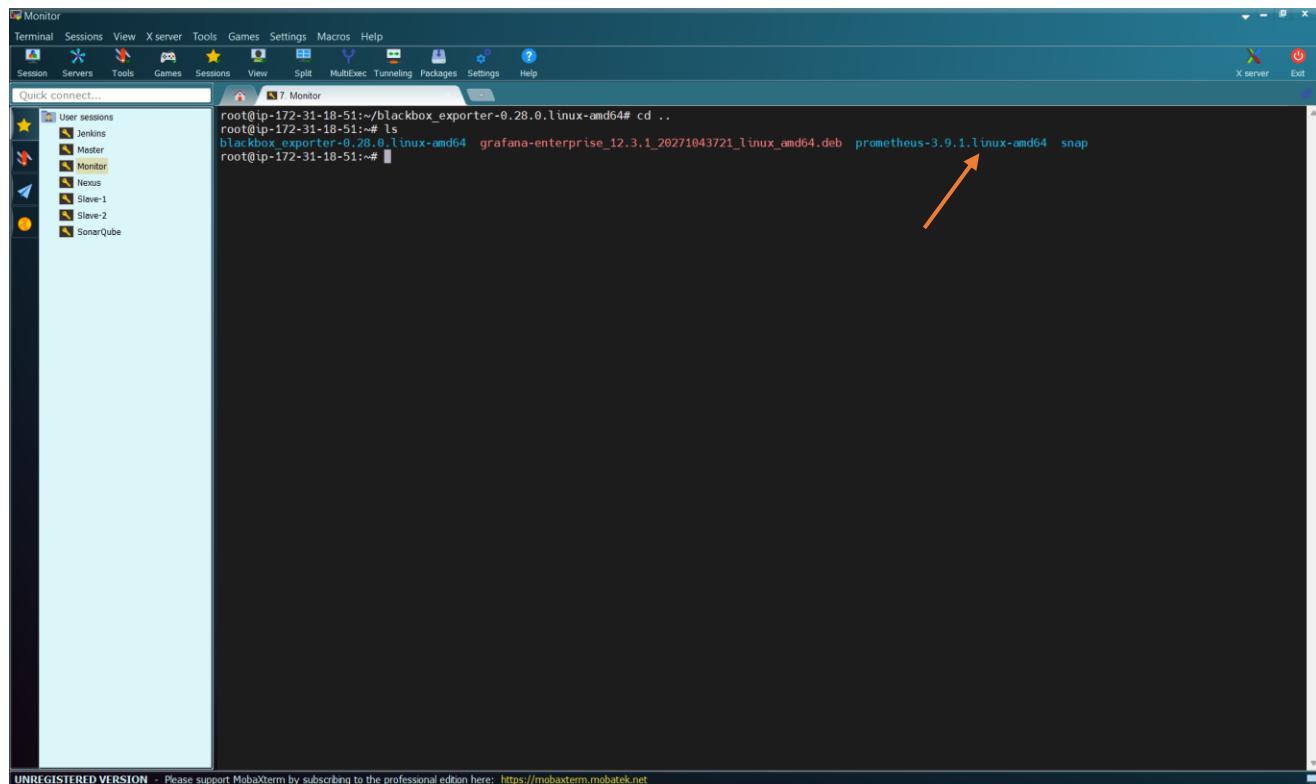
```
cd ..
```



```
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# cd ..
root@ip-172-31-18-51:~#
```

Run the command to see the content of this folder:

```
ls
```

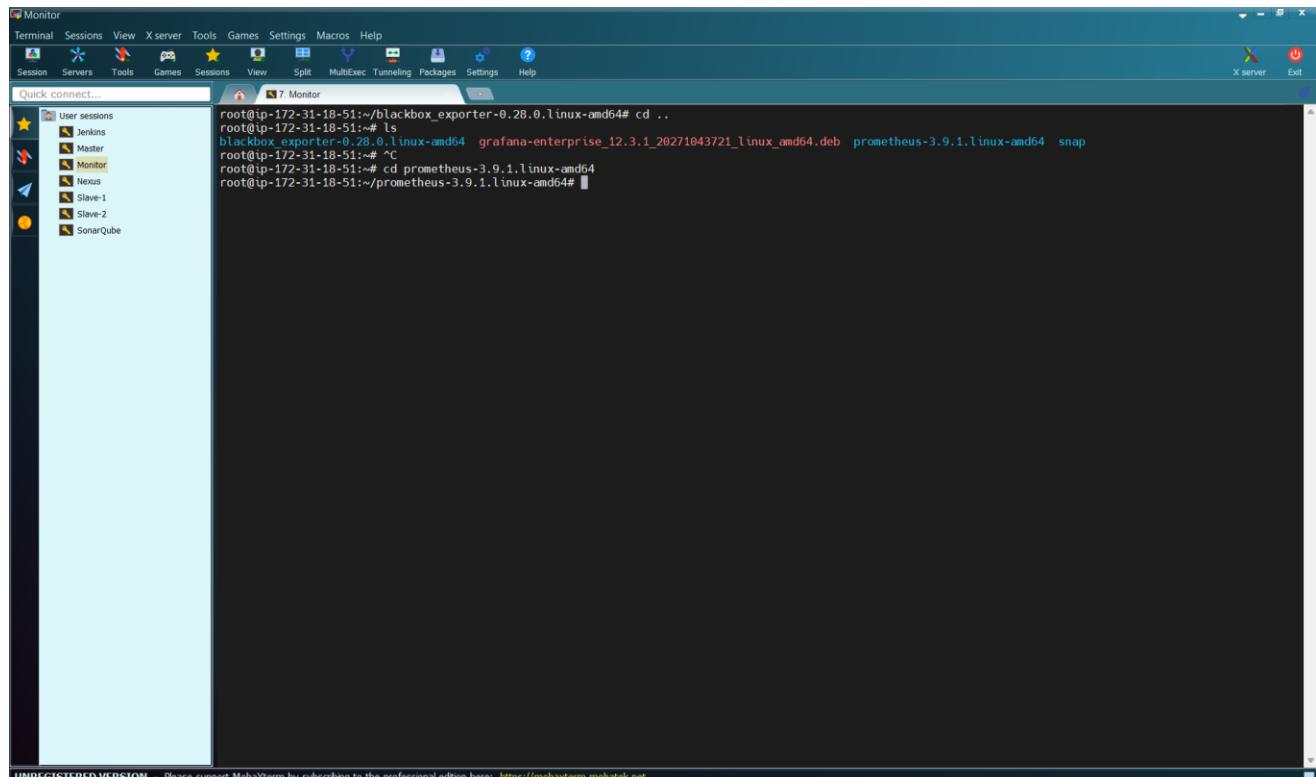


A screenshot of the MobaXterm application interface. The title bar says "Monitor". The menu bar includes "Terminal", "Sessions", "View", "X server", "Tools", "Games", "Settings", "Macros", and "Help". The toolbar has icons for "Session", "Servers", "Tools", "Games", "Sessions", "View", "Split", "MultiExec", "Tunneling", "Packages", "Settings", and "Help". A sidebar titled "Quick connect..." lists "User sessions": Jenkins, Master, Monitor, Nexus, Slave-1, Slave-2, and SonarQube. The main terminal window shows a root shell on a Linux system. The user runs the command "ls" which lists several files: blackbox\_exporter-0.28.0.linux-amd64, grafana-enterprise\_12.3.1\_20271043721\_linux\_amd64.deb, prometheus-3.9.1.linux-amd64, and snap. An orange arrow points from the word "ls" in the command line to the "ls" command in the terminal output.

```
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# cd ..
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  prometheus-3.9.1.linux-amd64  snap
```

Then, run the command to go into the Prometheus folder:

```
cd prometheus-3.9.1.linux-amd64
```

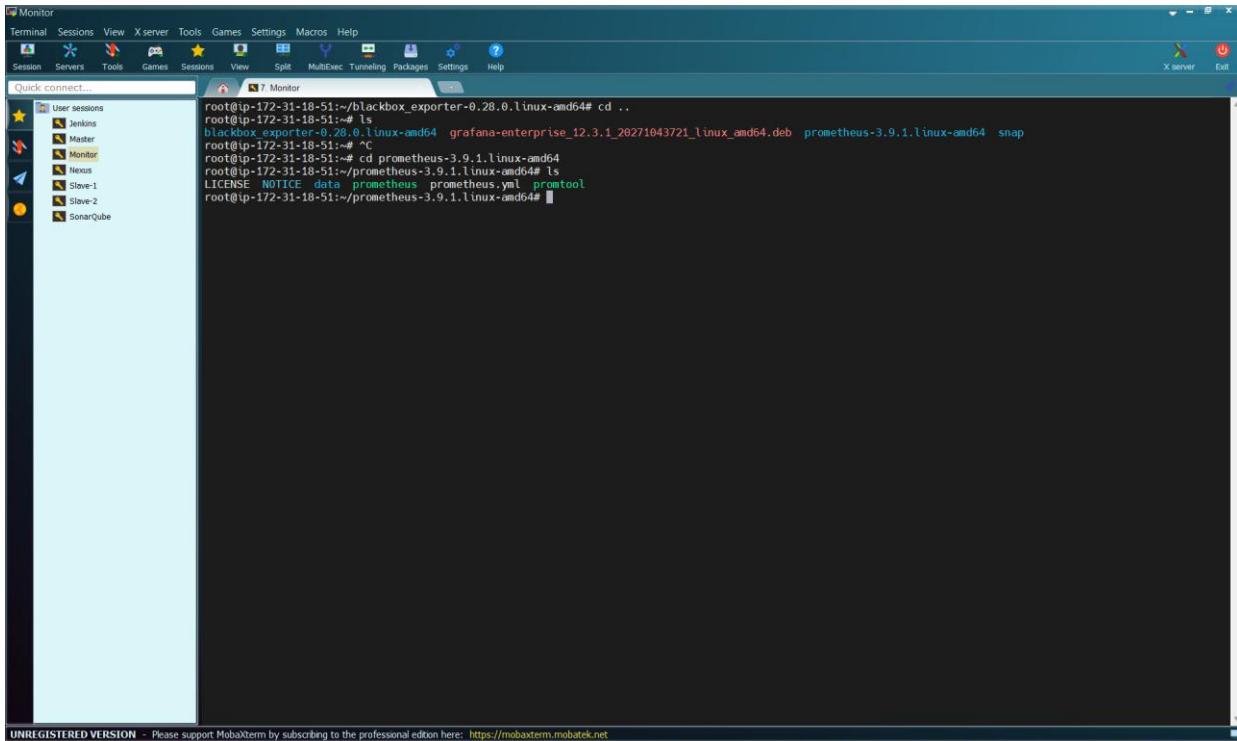


A screenshot of the MobaXterm application interface, identical to the previous one but with a different command in the terminal. The user runs "cd prometheus-3.9.1.linux-amd64". The terminal output shows the directory has changed to "/prometheus-3.9.1.linux-amd64". An orange arrow points from the "cd" command in the input field to the "cd" command in the terminal output.

```
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# cd ..
root@ip-172-31-18-51:~# ls
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  prometheus-3.9.1.linux-amd64  snap
root@ip-172-31-18-51:~# cd prometheus-3.9.1.linux-amd64
root@ip-172-31-18-51:/prometheus-3.9.1.linux-amd64#
```

Run the command to see the content:

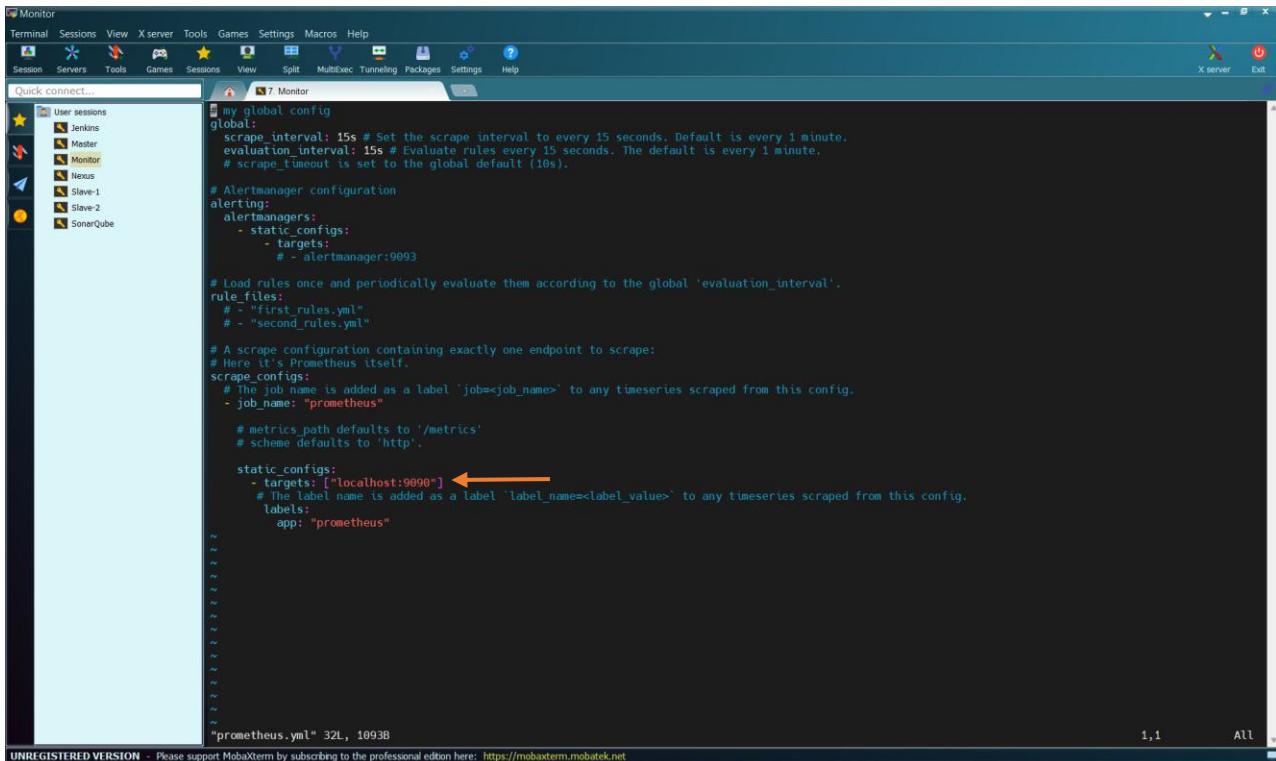
ls



```
root@ip-172-31-18-51:~/blackbox_exporter-0.28.0.linux-amd64# cd ..  
root@ip-172-31-18-51:~# ls  
blackbox_exporter-0.28.0.linux-amd64  grafana-enterprise_12.3.1_20271043721_linux_amd64.deb  prometheus-3.9.1.linux-amd64  snap  
root@ip-172-31-18-51:~# c  
root@ip-172-31-18-51:~# cd prometheus-3.9.1.linux-amd64  
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# ls  
LICENSE  NOTICE  data  prometheus  prometheus.yml  protool  
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64#
```

Then, let us edit the file “prometheus.yml”. We will first open the file using the command:

vi prometheus.yml



```
my global config  
global:  
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.  
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.  
  # scrape_timeout is set to the global default (10s).  
  
# Alertmanager configuration  
alerting:  
  alertmanagers:  
    - static_configs:  
      - targets:  
        # - alertmanager:9093  
  
# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.  
rule_files:  
  # - "first_rules.yml"  
  # - "second_rules.yml"  
  
# A scrape configuration containing exactly one endpoint to scrape:  
# Here it's Prometheus itself.  
scrape_configs:  
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.  
  - job_name: "prometheus"  
    # metrics_path defaults to '/metrics'  
    # scheme defaults to 'http'.  
  
    static_configs:  
      - targets: ["localhost:9090"] ←  
        # The label name is added as a label 'label_name=<label_value>' to any timeseries scraped from this config.  
        labels:  
          app: "prometheus"  
  
"
```

Go to [https://github.com/prometheus/blackbox\\_exporter](https://github.com/prometheus/blackbox_exporter)

The screenshot shows the GitHub repository page for `prometheus/blackbox_exporter`. The repository has 691 commits in the `master` branch. The sidebar on the right provides information about the repository, including its purpose as a Blackbox prober exporter, its license (Apache-2.0), and its status (5.5k stars, 1.2k forks).

Scroll down

The screenshot shows the GitHub repository page for `prometheus/blackbox_exporter`, specifically the `README` file. It contains configuration examples and notes about HTTP probes accepting additional parameters.

```
scrape_configs:
  job_name: 'blackbox'
  metrics_path: /probe
  params:
    module: [http_2xx] # Look for a HTTP 200 response.
  static_configs:
    - targets:
        - http://prometheus.io # Target to probe with http.
        - https://prometheus.io # Target to probe with https.
        - http://example.com:8080 # Target to probe with http on port 8080.
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: 127.0.0.1:9115 # The blackbox exporter's real hostname:port.
  job_name: 'blackbox_exporter' # collect blackbox exporter's operational metrics.
  static_configs:
    - targets: ['127.0.0.1:9115']
```

HTTP probes can accept an additional `hostname` parameter that will set `Host` header and TLS SNI. This can be especially useful with `dns_sd_config`:

Copy the job section

```

- job_name: 'blackbox'
  metrics_path: /probe
  params:
    module: [http_2xx] # Look for a HTTP 200 response.
  static_configs:
    - targets:
        - http://prometheus.io      # Target to probe with http.
        - https://prometheus.io     # Target to probe with https.
        - http://example.com:8080 # Target to probe with http on port 8080.
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: 127.0.0.1:9115 # The blackbox exporter's real hostname:port.
- job_name: 'blackbox_exporter' # collect blackbox exporter's operational metrics.
  static_configs:
    - targets: ['127.0.0.1:9115']

```

Then, paste just after “-target: [“localhost:9090”]”

```

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            - - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
  - job_name: 'blackbox'
    metrics_path: /probe
    params:
      module: [http_2xx] # Look for a HTTP 200 response.
    static_configs:
      - targets:
          - http://prometheus.io      # Target to probe with http.
          - https://prometheus.io     # Target to probe with https.
          - http://example.com:8080 # Target to probe with http on port 8080.
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: 127.0.0.1:9115 # The blackbox exporter's real hostname:port.
  - job_name: 'blackbox_exporter' # collect blackbox exporter's operational metrics.
    static_configs:
      - targets: ['127.0.0.1:9115']

    # The label name is added as a label `label_name=<label_value>` to any timeseries scraped from this config.
-- INSERT --

```

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

Remove the part above

```

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
    - job_name: 'blackbox'
    metrics_path: /probe
    params:
      module: [http_2xx] # Look for a HTTP 200 response.
    static_configs:
      - targets:
          - http://prometheus.io # Target to probe with http.
          - https://prometheus.io # Target to probe with https.
          - http://example.com:8080 # Target to probe with http on port 8080.
    relabel_configs:
      - source_labels: [__address__]
      target_label: __param_target
      - source_labels: [__param_target]
      target_label: instance
      - target_label: __address__
      replacement: 127.0.0.1:9115 # The blackbox exporter's real hostname:port.

    # The label name is added as a label 'label_name=<label_value>' to any timeseries scraped from this config.
    labels:
      app: "prometheus"
~
~
-- INSERT --

```

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

Then, we have to change this IP address to the Public IP address of where the blackbox exporter is running, that is our “Monitor” server.

```

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

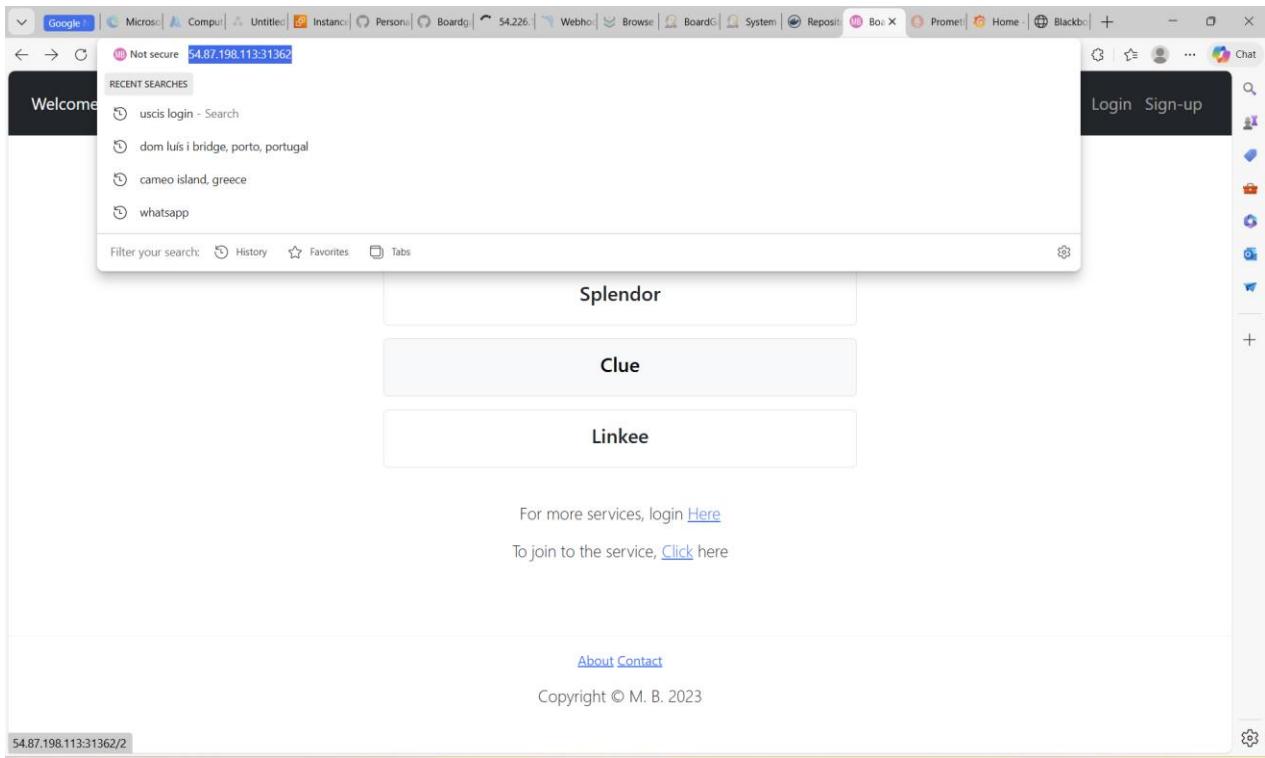
    static_configs:
      - targets: ["localhost:9090"]
    - job_name: 'blackbox'
    metrics_path: /probe
    params:
      module: [http_2xx] # Look for a HTTP 200 response.
    static_configs:
      - targets:
          - http://prometheus.io # Target to probe with http.
          - https://prometheus.io # Target to probe with https.
          - http://example.com:8080 # Target to probe with http on port 8080.
    relabel_configs:
      - source_labels: [__address__]
      target_label: __param_target
      - source_labels: [__param_target]
      target_label: instance
      - target_label: __address__
      replacement: 44.220.145.88:9115 # The blackbox exporter's real hostname:port.

    # The label name is added as a label 'label_name=<label_value>' to any timeseries scraped from this config.
    labels:
      app: "prometheus"
~
~
-- INSERT --

```

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

Above are the targets we want to monitor. We want to monitor the browser of our application. So, we will add its url there.



Copy the url

<http://54.87.198.113:31362/>

Paste it on the prometheus.yml file. With this we can monitor our application.

```

Monitor
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunnelling Packages Settings Help
Quick connect...
7. Monitor
scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
# scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  - "first_rules.yml"
  - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
    - job_name: 'blackbox'
    metrics_path: /probe
    params:
      module: [http_2xx] # Look for a HTTP 200 response.
    static_configs:
      - targets:
          - http://prometheus.io      # Target to probe with http.
          - http://54.87.198.113:31362
    relabel_configs:
      - source_labels: [_address_]
        target_label: param_target
      - source_labels: [_param_target]
        target_label: instance
      - target_label: address
        replacement: 44.220.145.88:9115 # The blackbox exporter's real hostname:port.

```

Save and exit the file.

A screenshot of the MobaXterm interface. The title bar says "Monitor". The menu bar includes Terminal, Sessions, View, Xserver, Tools, Games, Settings, Macros, Help, Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, Help, and X server. The status bar at the bottom says "UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>". The main window shows a terminal session titled "7. Monitor". The command "root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# vi prometheus.yml" is being run. The terminal window has a dark background and light text.

Run the command to see the content again:

```
ls
```

A screenshot of the MobaXterm interface, identical to the previous one but with a different command. The title bar says "Monitor". The menu bar and status bar are the same. The main window shows a terminal session titled "7. Monitor". The command "root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# ls" is being run. The output shows files: LICENSE, NOTICE, data, prometheus, prometheus.yml, and promtool. The terminal window has a dark background and light text.

Let us now check the port where Prometheus is running by running the command:

```
pgrep prometheus
```

```
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# vi prometheus.yml
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# ls
LICENSE NOTICE data prometheus prometheus.yml promtool
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# pgrep prometheus
1815
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64#
```

This is the ID where it is running. Let us kill the ID using the command:

```
kill 1815
```

```
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# vi prometheus.yml
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# ls
LICENSE NOTICE data prometheus prometheus.yml promtool
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# pgrep prometheus
1815
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# kill 1815
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# time=2026-01-11T00:56:58.447Z level=WARN source=main.go:1118 msg="Received an OS signal, exiting gracefully..." signal=terminated
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1157 msg="Stopping scrape discovery manager..."
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1157 msg="Stopping notify discovery manager..."
time=2026-01-11T00:56:58.447Z level=INFO source=manager.go:216 msg="Stopping rule manager..." component="rule manager"
time=2026-01-11T00:56:58.447Z level=INFO source=manager.go:232 msg="Rule manager stopped" component="rule manager"
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1194 msg="Stopping scrape manager..."
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1153 msg="Notify discovery manager stopped"
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1139 msg="Scrape discovery manager stopped"
time=2026-01-11T00:56:58.459Z level=INFO source=main.go:1186 msg="Scrape manager stopped"
time=2026-01-11T00:56:58.469Z level=INFO source=manager.go:561 msg="Stopping notification manager..." component=notifier
time=2026-01-11T00:56:58.469Z level=INFO source=manager.go:301 msg="Draining any remaining notifications..." component=notifier
time=2026-01-11T00:56:58.469Z level=INFO source=manager.go:307 msg="Remaining notifications drained" component=notifier
time=2026-01-11T00:56:58.469Z level=INFO source=manager.go:234 msg="Notification manager stopped" component=notifier
time=2026-01-11T00:56:58.469Z level=INFO source=main.go:1466 msg="Notifier manager stopped"
time=2026-01-11T00:56:58.469Z level=INFO source=main.go:1480 msg="See you next time!"
```

Press "enter"

```

root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# vi prometheus.yml
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# ls
LICENSE NOTICE data prometheus prometheus.yml promtool
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# pgrep prometheus
1815
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# kill 1815
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# time=2026-01-11T00:56:58.447Z level=WARN source=main.go:1118 msg="Received an OS signal, exiting gracefully..." signal=terminated
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1143 msg="Stopping scrape discovery manager..."
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1157 msg="Stopping notify discovery manager..."
time=2026-01-11T00:56:58.447Z level=INFO source=manager.go:216 msg="Stopping rule manager..." component="rule manager"
time=2026-01-11T00:56:58.447Z level=INFO source=manager.go:232 msg="Rule manager stopped" component="rule manager"
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1194 msg="Stopping scrape manager..."
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1153 msg="Notify discovery manager stopped"
time=2026-01-11T00:56:58.447Z level=INFO source=main.go:1139 msg="Scrape discovery manager stopped"
time=2026-01-11T00:56:58.459Z level=INFO source=main.go:1186 msg="Scrape manager stopped"
time=2026-01-11T00:56:58.469Z level=INFO source=manager.go:561 msg="Stopping notification manager..." component=notifier
time=2026-01-11T00:56:58.469Z level=INFO source=manager.go:301 msg="Draining any remaining notifications..." component=notifier
time=2026-01-11T00:56:58.469Z level=INFO source=manager.go:307 msg="Remaining notifications drained" component=notifier
time=2026-01-11T00:56:58.469Z level=INFO source=manager.go:234 msg="Notification manager stopped" component=notifier
time=2026-01-11T00:56:58.469Z level=INFO source=main.go:1466 msg="Notifier manager stopped"
time=2026-01-11T00:56:58.469Z level=INFO source=main.go:1480 msg="See you next time!"

[1]- Done . ./prometheus
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64#

```

Now, let us restart prometheus using the command:

./prometheus &

```

root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# ./prometheus &
[3] 3561
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# time=2026-01-11T01:05:03.095Z level=INFO source=main.go:1589 msg="updated GOGC" old=100 new=75
time=2026-01-11T01:05:03.096Z level=INFO source=main.go:704 msg="Leaving GOMAXPROCS=2: CPU quota undefined" component=automaxprocs
time=2026-01-11T01:05:03.097Z level=INFO source=memlimit.go:198 msg="GOMEMLIMIT is updated" component=automemlimit package=github.com/KimMachineGun/automemlim
it/memlimit GOMEMLIMIT=7491332505 previous=9223372036854775807
time=2026-01-11T01:05:03.097Z level=INFO source=main.go:752 msg="No time or size retention was set so using the default time retention" duration=15d
time=2026-01-11T01:05:03.097Z level=INFO source=main.go:803 msg="Starting Prometheus Server" mode=server version="(version=3.9.1, branch=HEAD, revision=9ec59b
affb547e24f146ba3eb2901e58feabdb)" ip-172-31-18-51 (none)
time=2026-01-11T01:05:03.097Z level=INFO source=main.go:808 msg="operational information" build_context="(go=go1.25.5, platform=linux/amd64, user=root@061c3a92
12:9e, date=20260107-16:08:09, tags=netgo,builtinassets)" host_details="(Linux 6.14.0-1015-aws #1~24.04.1-Ubuntu SMP Tue Sep 23 22:44:48 UTC 2025 x86_64 ip-1
72-31-18-51 (none))" fd_limits="soft=1048575, hard=1048576" vm_limits="soft=unlimited, hard=unlimited"
time=2026-01-11T01:05:03.099Z level=INFO source=web.go:684 msg="Start listening for connections" component=web address=0.0.0.0:9090
time=2026-01-11T01:05:03.101Z level=INFO source=main.go:1331 msg="Starting TSDB ..."
time=2026-01-11T01:05:03.103Z level=INFO source=ts_config.go:359 msg="Listening on component=web address=[::]:9090
time=2026-01-11T01:05:03.103Z level=INFO source=ts_config.go:357 msg="TLS is disabled" component=web http2=false address=[::]:9090
time=2026-01-11T01:05:03.106Z level=INFO source=head.go:681 msg="Replaying on-disk memory mappable chunks if any" component=tsdb
time=2026-01-11T01:05:03.107Z level=INFO source=head.go:767 msg="On-disk memory mappable chunks replay completed" component=tsdb duration=303.931µs
time=2026-01-11T01:05:03.107Z level=INFO source=head.go:775 msg="Replaying WAL, this may take a while" component=tsdb
time=2026-01-11T01:05:03.111Z level=INFO source=head.go:848 msg="WAL segment loaded" component=tsdb segment=0 maxSegment=2 duration=4.302895ms
time=2026-01-11T01:05:03.126Z level=INFO source=head.go:848 msg="WAL segment loaded" component=tsdb segment=1 maxSegment=2 duration=14.882178ms
time=2026-01-11T01:05:03.126Z level=INFO source=head.go:848 msg="WAL segment loaded" component=tsdb segment=2 maxSegment=2 duration=162.162ms
time=2026-01-11T01:05:03.126Z level=INFO source=head.go:860 msg="WAL replay completed" component=tsdb checkpoint_replay_duration=39.257µs wal_replay_duration=
19.36137ms wal_replay_duration=57ins chunk_snapshot_load_duration=0µs mmap_chunk_replay_duration=303.931µs total_replay_duration=20.039923ms
time=2026-01-11T01:05:03.128Z level=INFO source=main.go:1352 msg="filesystem information" fs_type=EXT4_SUPER_MAGIC
time=2026-01-11T01:05:03.128Z level=INFO source=main.go:1359 msg="TSDB started"
time=2026-01-11T01:05:03.128Z level=INFO source=main.go:1542 msg="Loading configuration file" filename=prometheus.yml
time=2026-01-11T01:05:03.132Z level=INFO source=main.go:1582 msg="Completed loading of configuration file" db_storage=1.369µs remote_storage=1.5µs web_handler
=862ns query_engine=1.28µs scrape=3.36529ms scrape_sd=69.229µs notify_sd=16.788µs rules=1.911µs tracing=3.152µs filename=prometheus.yml total
Duration=3.88368ms
time=2026-01-11T01:05:03.132Z level=INFO source=main.go:1316 msg="Server is ready to receive web requests."
time=2026-01-11T01:05:03.132Z level=INFO source=manager.go:202 msg="Starting rule manager..." component="rule manager"

```

Press "Enter"

```

Monitor
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
User sessions
  Jenkins
  Master
  Monitor
  Nexus
  Slave-1
  Slave-2
  SonarQube

7. Monitor
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# ./prometheus &
[3] 3561
root@ip-172-31-18-51:~/prometheus-3.9.1.linux-amd64# time=2026-01-11T01:05:03.095Z level=INFO source=main.go:1589 msg="updated GOGC" old=100 new=75
time=2026-01-11T01:05:03.096Z level=INFO source=main.go:704 msg="Leaving GOMAXPROCS=2: CPU quota undefined" component=automaxprocs
time=2026-01-11T01:05:03.097Z level=INFO source=memlimit.go:198 msg="GOMEMLIMIT is updated" component=automemlimit package=github.com/KimMachineGun/automemlim
it/memlimit GOMEMLIMIT=7491332565 previous=9223372036854775807
time=2026-01-11T01:05:03.097Z level=INFO source=main.go:752 msg="No time or size retention was set so using the default time retention" duration=15d
time=2026-01-11T01:05:03.097Z level=INFO source=main.go:803 msg="Starting Prometheus Server" mode=server version="(version=3.9.1, branch=HEAD, revision=9ec59b
affb547e24f146ba53eb2901e58feabd8)"
time=2026-01-11T01:05:03.097Z level=INFO source=main.go:808 msg="operational information" build_context="(go=go1.25.5, platform=linux/amd64, user=root@061c3a92
12c9e, date=20260107-16:08:09, tags=netgo,builtinassets)" host_details="(Linux 6.14.0-1015-aws #15~24.04.1-Ubuntu SMP Tue Sep 23 22:44:48 UTC 2025 x86_64 ip-1
72-31-18-51 {none})" fd_limits="(soft=1048575, hard=1048575)" vm_limits="(soft=unlimited, hard=unlimited)"
time=2026-01-11T01:05:03.099Z level=INFO source=web.go:684 msg="Start listening for connections" component=web address=0.0.0.0:9090
time=2026-01-11T01:05:03.101Z level=INFO source=main.go:1331 msg="Starting TSDB ..."
time=2026-01-11T01:05:03.103Z level=INFO source=tls_config.go:354 msg="Listening on" component=web address=[::]:9090
time=2026-01-11T01:05:03.103Z level=INFO source=tls_config.go:357 msg="TLS is disabled." component=web http=false address=[::]:9090
time=2026-01-11T01:05:03.106Z level=INFO source=head.go:681 msg="Replaying on-disk memory mappable chunks if any" component=tsdb
time=2026-01-11T01:05:03.107Z level=INFO source=head.go:767 msg="On-disk memory mappable chunks replay completed" component=tsdb duration=303.931μs
time=2026-01-11T01:05:03.111Z level=INFO source=head.go:775 msg="Replaying WAL, this may take a while" component=tsdb
time=2026-01-11T01:05:03.111Z level=INFO source=head.go:841 msg="WAL segment loaded" component=tsdb segment=0 maxSegment=2 duration=4.302895ms
time=2026-01-11T01:05:03.126Z level=INFO source=head.go:841 msg="WAL segment loaded" component=tsdb segment=1 maxSegment=2 duration=14.802178ms
time=2026-01-11T01:05:03.126Z level=INFO source=head.go:848 msg="WAL segment loaded" component=tsdb segment=2 maxSegment=2 duration=162.162μs
time=2026-01-11T01:05:03.126Z level=INFO source=head.go:888 msg="WAL replay completed" component=tsdb checkpoint_replay_duration=39.257μs wal_replay_duration=
19.361373ms wbl_replay_duration=571ns chunk_snapshot_load_duration=8s mmap_chunk_replay_duration=303.931μs total_replay_duration=20.039923ms
time=2026-01-11T01:05:03.128Z level=INFO source=main.go:1352 msg="filesystem information" fs_type=EXT4_SUPER_MAGIC
time=2026-01-11T01:05:03.128Z level=INFO source=main.go:1355 msg="TSDB started"
time=2026-01-11T01:05:03.132Z level=INFO source=main.go:1542 msg="Loading configuration file" filename=prometheus.yml
time=2026-01-11T01:05:03.132Z level=INFO source=main.go:1582 msg="Completed loading of configuration file" db_storage=1.369us remote_storage=1.5us web handler
=862ms query_engine=1.28us scrape=3.36529ms scrape_sd=69.229us notify=146.142μs notify_sd=16.788μs rules=1.911μs tracing=3.152μs filename=prometheus.yml total
Duration=3.88368ms
time=2026-01-11T01:05:03.132Z level=INFO source=main.go:1316 msg="Server is ready to receive web requests."
time=2026-01-11T01:05:03.132Z level=INFO source=manager.go:202 msg="Starting rule manager..." component=rule_manager"
time=2026-01-11T01:05:21.844Z level=WARN source=http.go:490 msg="Received redirect" module=http_2xx target=https://prometheus.io location=https://prometheus.io
/

```

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

Refresh the Prometheus browser page

Click on the drop down on "Status"

Not secure 44.220.145.88:9090/query

Prometheus

Query Alerts Status

Enter expression (press Shift+Enter for newlines)

Table Graph Explain

Evaluation time

No data queried yet

+ Add query

Monitoring status

- Target health
- Rule health
- Service discovery

Server status

- Runtime & build information
- TSDB status
- Command-line flags
- Configuration
- Alertmanager discovery

44.220.145.88:9090/targets

## Select “Target Health”

Not secure 44.220.145.88:9090/targets

Prometheus

Status > Target health

Select scrape pool

Filter by target health

Filter by endpoint or labels

blackbox

Endpoint	Labels	Last scrape	State
http://44.220.145.88:9115/probe	instance="http://54.87.198.113:31362" job="blackbox"	14.351s ago	UP
http://44.220.145.88:9115/probe	instance="http://prometheus.io" job="blackbox"	14.339s ago	UP

prometheus

Endpoint	Labels	Last scrape	State
http://localhost:9090/metrics	instance="localhost:9090" job="prometheus"	13.209s ago	UP

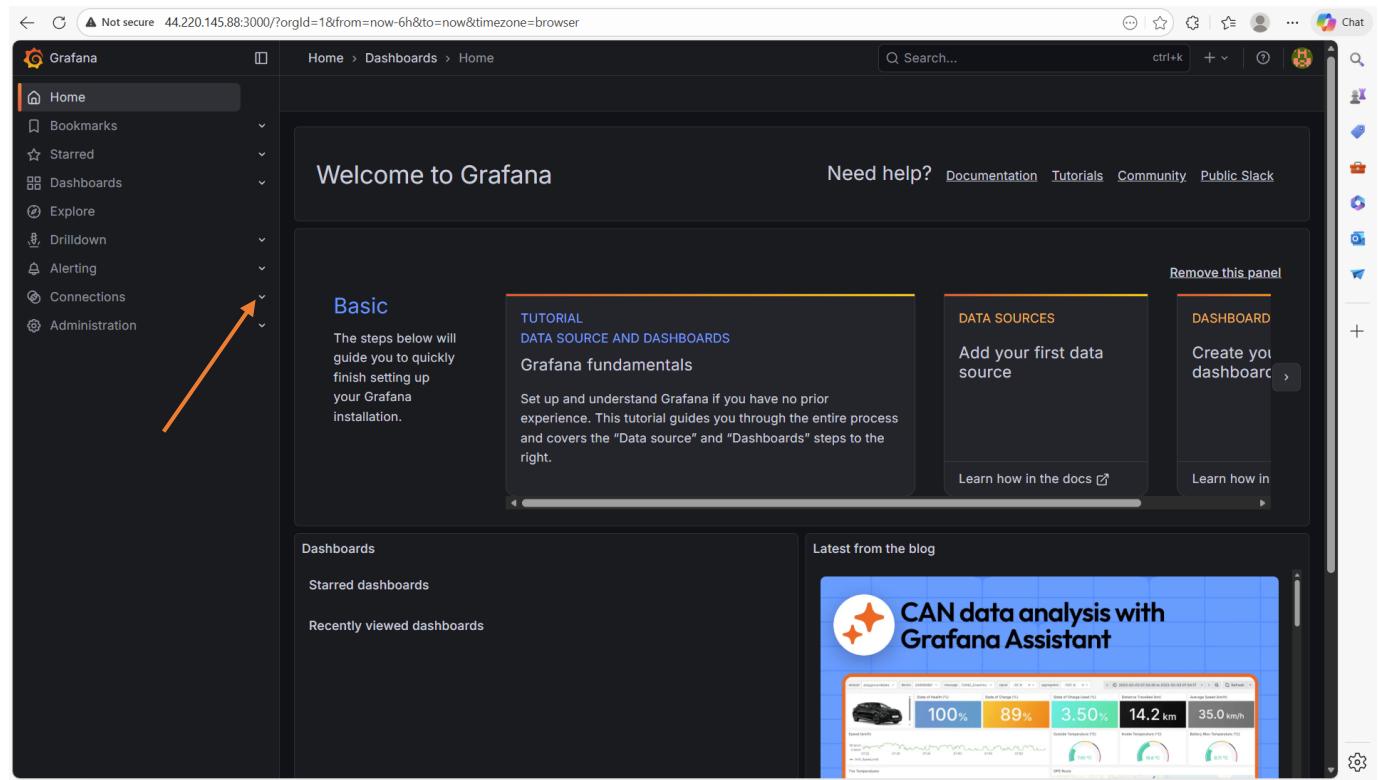
You can see the targets are up and running.

## 6.3 Website Monitoring

Here we are going to use Prometheus and Grafana for website monitoring.

### 6.3.1 Add Prometheus as data Source to Grafana

Now, we have to add Prometheus as data source to Grafana. Go to the Grafana browser



The screenshot shows the Grafana home page. On the left, there is a sidebar with various navigation options: Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, Connections (which is highlighted with an orange arrow), and Administration. The main content area features a "Welcome to Grafana" message and several promotional panels: "Basic" (with steps to quickly set up Grafana), "TUTORIAL DATA SOURCE AND DASHBOARDS" (about Grafana fundamentals), "DATA SOURCES" (to add your first data source), and "DASHBOARD" (to create your dashboard). Below these panels, there are links for "Dashboards", "Starred dashboards", and "Recently viewed dashboards". A "Latest from the blog" section is also visible, featuring a thumbnail for an article about "CAN data analysis with Grafana Assistant".

Click on the drop down on “Connections”

The screenshot shows the Grafana Connections page. On the left sidebar, under the 'Connections' section, the 'Data sources' option is highlighted with an orange arrow pointing to it. The main content area displays a 'Welcome to Connections' message and two buttons: 'Add new connection' and 'View configured data sources'.

Click on “Data Sources”

The screenshot shows the Grafana Data sources page. The 'Data sources' option in the sidebar is highlighted with an orange arrow. The main content area shows a cartoon character and the message 'No data sources defined'. A large orange arrow points to the 'Add data source' button at the bottom of the page.

Click on “Add Data Source”

The screenshot shows the 'Add data source' interface in Grafana. On the left is a sidebar with navigation links like Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, Connections, and Administration. Under 'Connections', 'Data sources' is selected. The main area is titled 'Add data source' and 'Choose a data source type'. A search bar at the top right says 'Search...'. Below it, there are two sections: 'Time series databases' and 'Logging & document databases'. In the 'Time series databases' section, 'Prometheus' is highlighted with an orange arrow pointing to its icon. Other options include Graphite, InfluxDB, and OpenTSDB. In the 'Logging & document databases' section, there is one entry for Loki.

Click on “Prometheus”

The screenshot shows the 'Edit data source' page for Prometheus. The URL in the browser is 44.220.145.88:3000/connections/datasources/edit/df9svld757gga. The left sidebar shows 'Data sources' is selected. The main area has a title 'prometheus' with a Prometheus icon. It shows 'Type: Prometheus' and 'Alerting Supported'. Below this is a settings tab bar with 'Settings' (which is active), 'Dashboards', 'Permissions', 'Insights', and 'Cache'. A modal window titled 'Configure your Prometheus data source below' is open, with a note about using the free forever Grafana Cloud plan. The 'Connection' section contains a 'Prometheus server URL' input field containing 'http://localhost:9090'. An orange arrow points from the previous screenshot's 'Prometheus' selection here. A validation error message 'Please enter a valid URL' is displayed below the input field.

Here, provide the url address of the Prometheus browser.

The screenshot shows the Grafana interface for managing data sources. On the left, a sidebar navigation includes Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, Connections (with 'Add new connection' and 'Data sources' selected), and Administration. The main content area is titled 'prometheus' under the 'Type: Prometheus' section. A prominent callout box at the top right says 'Configure your Prometheus data source below' and 'Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the [free-for-ever Grafana Cloud plan](#)'. Below this, there's a 'Name' field set to 'prometheus', a 'Default' toggle switch, and a note about configuring it in the config file or below. A link to 'view the documentation' is provided. The 'Connection' section contains a 'Prometheus server URL' input field with the value 'http://44.220.145.88:9090'. The 'Authentication' section is currently empty.

Scroll down to the end

This screenshot shows the 'Settings' tab of the Prometheus data source configuration. It includes sections for 'Performance' (Prometheus type set to 'Choose', Cache level 'Low', Incremental querying (beta) off, Disable recording rules (beta) off), 'Other' (Custom query parameters, HTTP method 'POST', Series limit '40000', Use series endpoint off), and 'Exemplars' (+ Add). An orange arrow points from the bottom of the 'Exemplars' section down to the 'Save & test' button. At the bottom of the page, there are 'Delete' and 'Save & test' buttons.

Click on “Save and Test”

The screenshot shows the Grafana interface for managing data sources. The left sidebar is dark-themed and includes sections for Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, Connections (with 'Add new connection' and 'Data sources' selected), and Administration. The main content area is titled 'Connections > Data sources > prometheus'. It shows configuration options for the Prometheus type (set to 'Choose'), Cache level (set to 'Low'), Incremental querying (beta) (disabled), and Disable recording rules (beta) (disabled). Under 'Other', there are fields for Custom query parameters (example: max\_source\_resolution=5m&tin), HTTP method (set to 'POST'), Series limit (set to '40000'), and Use series endpoint (disabled). A section for 'Exemplars' has a '+ Add' button. A green success message box contains the text: 'Successfully queried the Prometheus API.' followed by 'Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#).'. Below the message are 'Delete' and 'Save & test' buttons. An orange arrow points from the text 'Next, you can start to visualize data by ...' towards the 'Save & test' button.

You can see, it is running fine. It is able to be connected.

### 6.3.2 Import the Dashboard

We have to import the dashboard

This screenshot is identical to the one above, showing the successful connection to the Prometheus API. However, an orange arrow points from the top right corner of the screen towards the 'Save & test' button, indicating the next step in the process.

Click on the drop down on (+)

The screenshot shows the Grafana interface for managing data sources. The left sidebar has 'Data sources' selected. The main area is titled 'Connections > Data sources > prometheus'. A dropdown menu is open at the top right, with 'Import dashboard' highlighted by an orange arrow. A success message at the bottom says 'Successfully queried the Prometheus API.'

Select “Import Dashboard”

The screenshot shows the 'Import dashboard' page. The left sidebar has 'Dashboards' selected. The main area has a title 'Import dashboard' and a sub-instruction 'Import dashboard from file or Grafana.com'. It features a 'Upload dashboard JSON file' section with a file input field and a 'Load' button. Below it is a 'Grafana.com dashboard URL or ID' input field with a 'Load' button. At the bottom is a 'Import via dashboard JSON model' code editor containing a JSON snippet, with 'Load' and 'Cancel' buttons at the bottom.

Now, we have to create dashboard. Go to google and search for “blackbox Grafana dashboard”

Google

blackbox Grafana dashboard

All Images Shopping Videos News Short videos More Tools

Grafana https://grafana.com › grafana › dashboards › 7587-prom... :

**Prometheus Blackbox Exporter | Grafana Labs**

The Prometheus Blackbox Exporter dashboard uses the prometheus data source to create a Grafana dashboard with the graph and singlestat panels. [Read more](#)

Grafana https://grafana.com › grafana › dashboards › 13659-blac... :

**Blackbox Exporter (HTTP prober)**

The Blackbox Exporter (HTTP prober) dashboard uses the prometheus data source to create a Grafana dashboard with the graph and table panels. [Read more](#)

Grafana https://grafana.com › grafana › dashboards › 14928-pro... :

**Prometheus Blackbox Exporter | Grafana Labs**

Easily monitor any Prometheus-compatible and publicly accessible metrics URL with Grafana Cloud's out-of-the-box monitoring solution. Learn more. Get this ... [Read more](#)

**Discussions and forums**

[How can I add multiple sources of blackbox as dropdown in grafana ...](#)

Grafana Community - 4 years ago :

Grafana Labs Community Forums · How can I add multiple sources of blackbox as dropdown in grafana dashboard? Grafana · nicegu... [More >](#)

**Simple UP/Down dashboard**

Grafana Community - 5 years ago :

Click on the above link

grafana.com/grafana/dashboards/7587-prometheus-blackbox-exporter/

Grafana Labs Products Open Source Solutions Learn Docs Pricing Downloads Contact us Sign in

(Prometheus)

Grafana Labs solution

Easily monitor any Prometheus-compatible and publicly accessible metrics URL with Grafana Cloud's out-of-the-box monitoring solution.

[Learn more](#)

**Get this dashboard**

1 Sign up for Grafana Cloud [Create free account](#)

2 Import the dashboard template [Copy ID to clipboard](#)

or [Download JSON](#)

Status UP

HTTP Status Code 200

HTTP Version 1.1

SSL YES

SSL Expiry 1 year, 9 months, 2 weeks

Average Probe Duration 1.1 s

Average DNS Lookup 168 ms

HTTP Duration

Probe Duration

SSL Metrics

Global Probe Metrics

The Prometheus Blackbox Exporter dashboard uses the prometheus data source to create a Grafana dashboard with the graph and singlestat panels.

**Revisions**

Revision	Description	Created	Action
3	Prometheus Blackbox Exporter Overview	2018-08-19T16:22:07	<a href="#">Download</a>
2		2018-08-19T10:07:24	<a href="#">Download</a>

Click on “Copy ID to clipboard”

7587

436

The screenshot shows a Grafana dashboard titled "Prometheus Blackbox Exporter Overview". It features several panels: a singlestat panel showing "Status UP", an HTTP Status Code of "200", and an SSL Version of "1.1"; a graph panel titled "HTTP Duration" showing request times; a graph panel titled "Probe Duration" showing probe times; and a singlestat panel showing "SSL YES". Below the dashboard, a note states: "The Prometheus Blackbox Exporter dashboard uses the prometheus data source to create a Grafana dashboard with the graph and singlestat panels." A sidebar on the right provides options to "Get this dashboard" via sign-up or dashboard ID copying.

Paste it on the Grafana browser and paste it on “Find and Import Dashboards”

The screenshot shows the "Import dashboard" page in the Grafana interface. It includes a file upload area for "Dashboard JSON file", a search bar, and a "Load" button highlighted with an orange arrow. The "Load" button is located at the bottom right of the JSON model input field.

Click on “Load”

The screenshot shows the 'Import dashboard' screen in Grafana. On the left is a sidebar with navigation links like Home, Bookmarks, Starred, Dashboards (which is selected), Explore, Drilldown, Alerting, Connections, and Administration. The main area has a title 'Import dashboard' and sub-sections 'Importing dashboard from Grafana.com' (published by sparanoid, updated on 2018-08-19 12:22:07) and 'Options'. Under Options, there's a 'Name' field containing 'Prometheus Blackbox Exporter', a 'Folder' dropdown set to 'Dashboards', and a 'Unique identifier (UID)' section with a text input 'xtkCtBkiz' and a 'Change uid' button. Below these is a dropdown menu titled 'Select a Prometheus data source' with an arrow pointing to it. At the bottom are 'Import' and 'Cancel' buttons.

Click on the drop down on “Prometheus”

This screenshot shows the same 'Import dashboard' screen as the previous one, but with a different focus. An orange arrow points to the 'prometheus' option in the 'Select a Prometheus data source' dropdown menu. The menu also includes 'default' and 'Prometheus'.

Select “Prometheus”

Import dashboard

Importing dashboard from [Grafana.com](#)

Published by sparanoid

Updated on 2018-08-19 12:22:07

Options

Name: Prometheus Blackbox Exporter

Folder: Dashboards

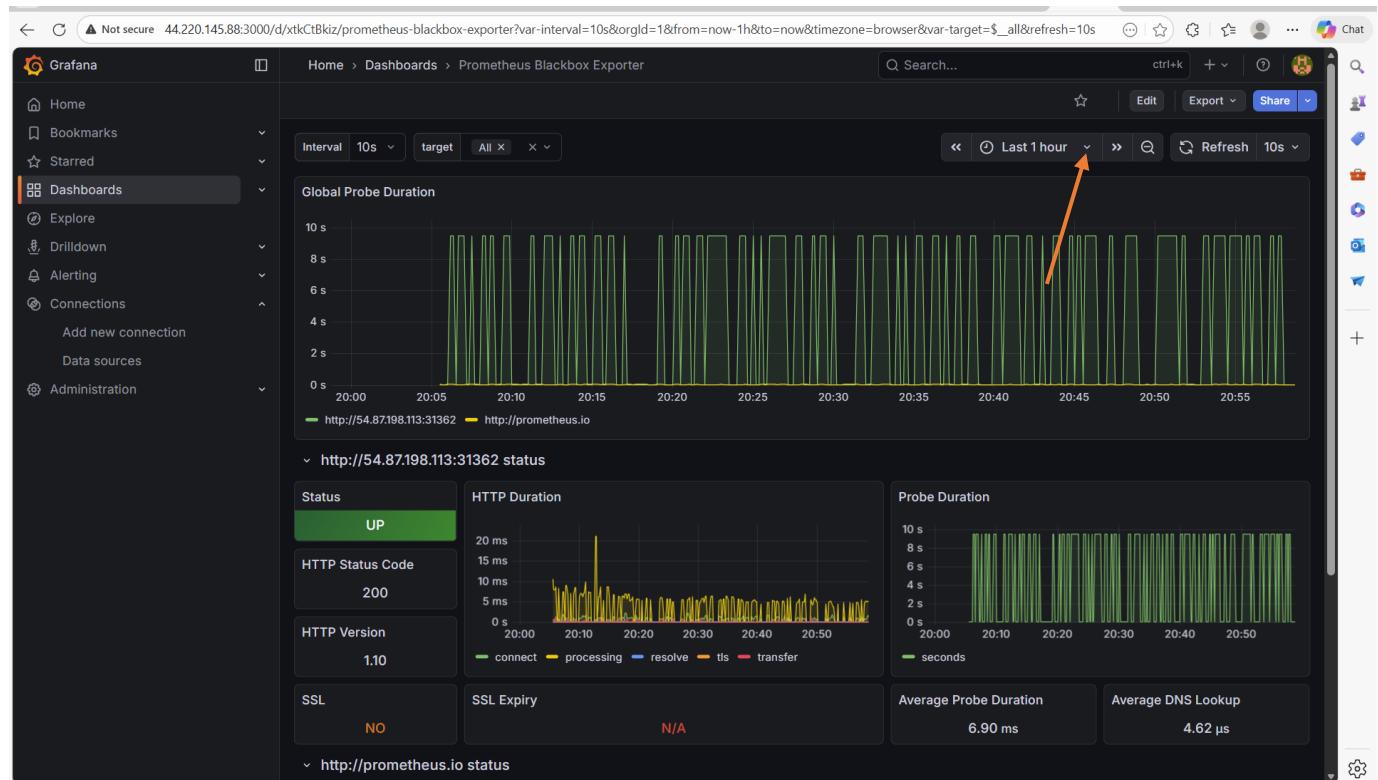
Unique identifier (UID)  
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

xtkCtBkiz [Change uid](#)

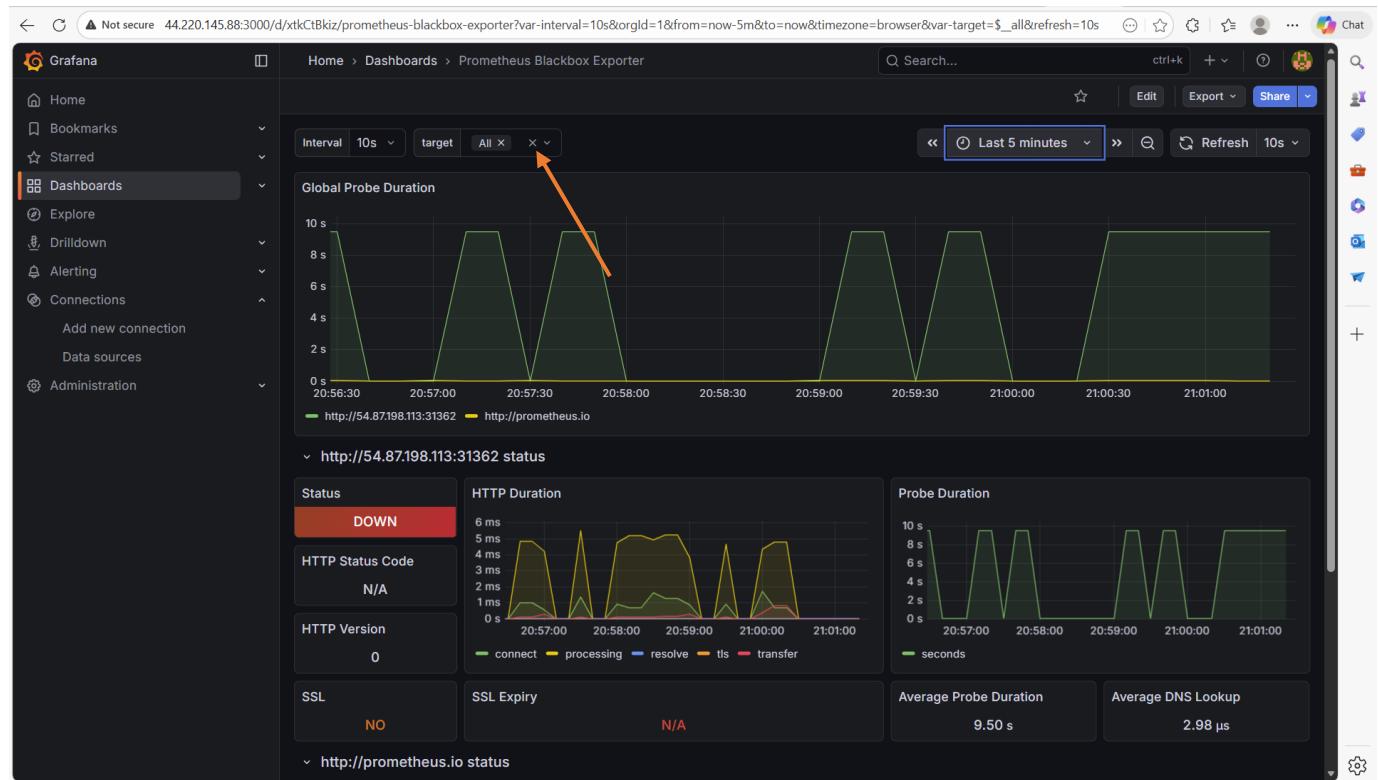
prometheus  
prometheus

[Import](#) [Cancel](#)

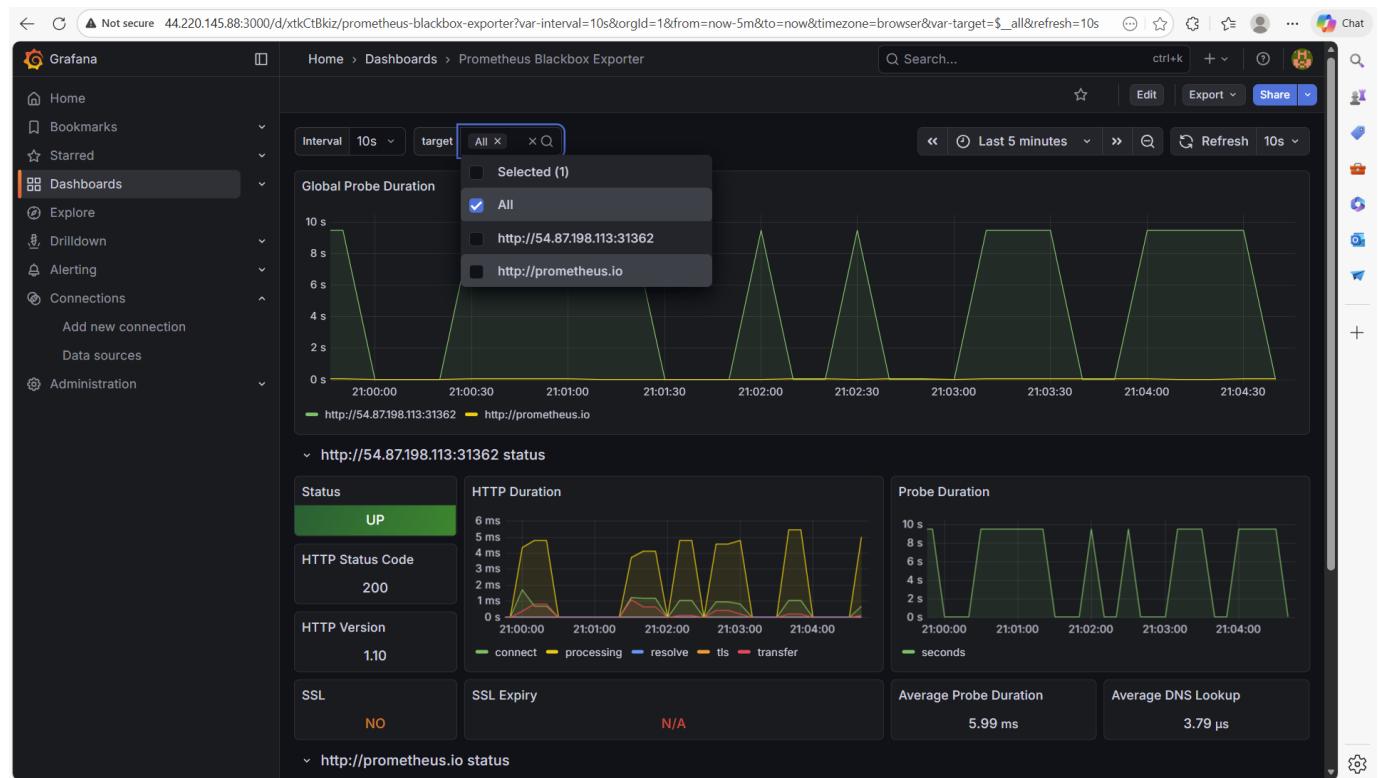
Click on “Import”



Let us change it to the last five minutes. Click on the drop down on “Last 1 Hour” and select “5Minutes”



You can select the target you want to monitor by clicking on the drop down on “All”



## 7 Phase 5: Automate the Process

In this phase, we will automate the pipeline so that whenever any change is made in the GitHub repository and the changes are committed. The Jenkins pipeline is triggered to run automatically.

To achieve this, we have to make two changes. One in Jenkins and the other in GitHub.

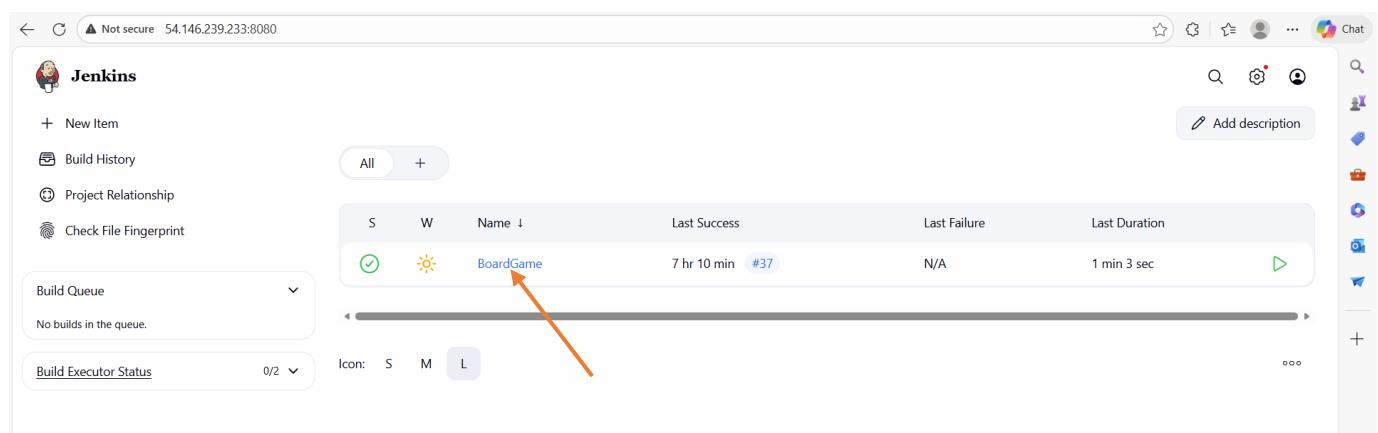
### 7.1 Modification on Jenkins

We are going to modify Jenkins by adding the URL of our GitHub repository and add a trigger such that when changes are made in the repository, the trigger enables the pipeline to start building.

#### 7.1.1 Add the URL of the GitHub Repository on Jenkins

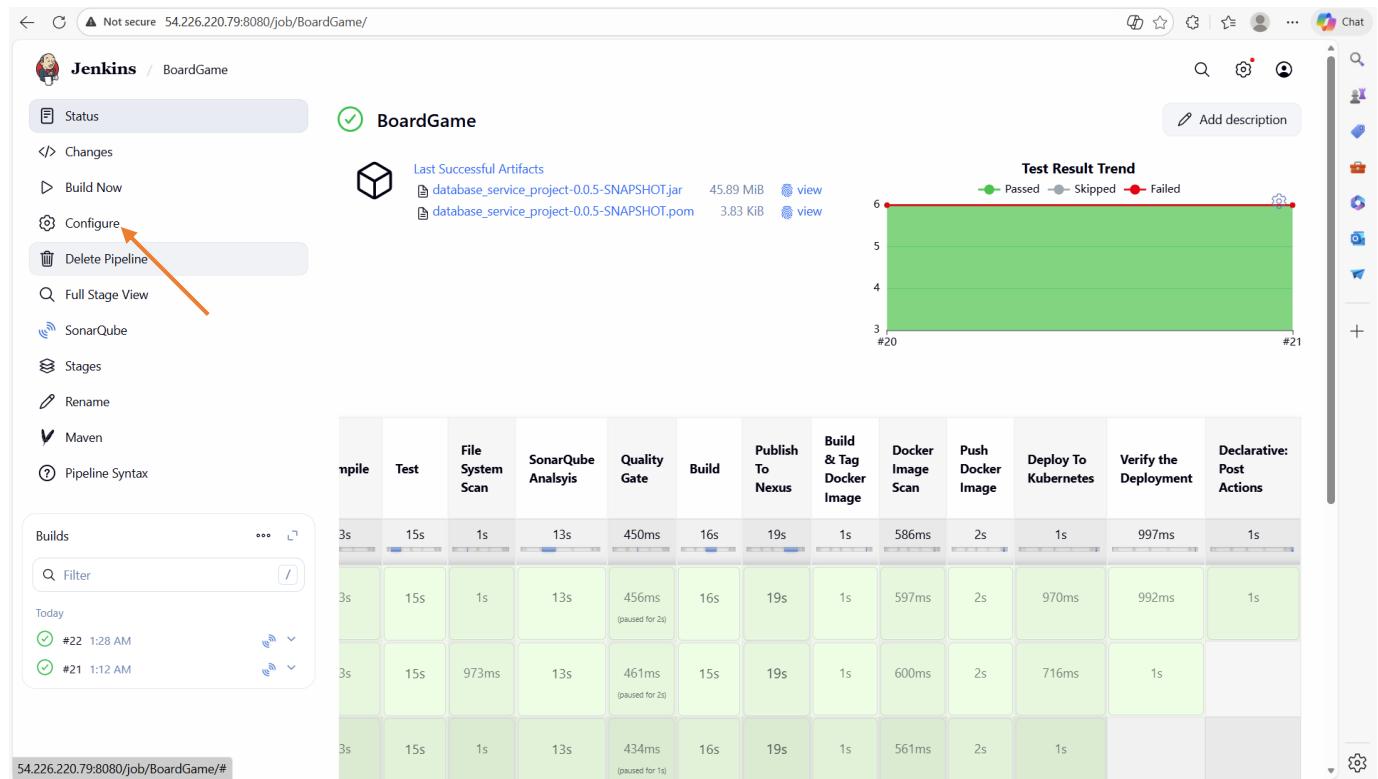
We are going to add the URL of our GitHub repository on Jenkins

Go to Jenkins Dashboard



The screenshot shows the Jenkins dashboard with the 'BoardGame' pipeline listed in the center. The pipeline has a green status icon, a yellow warning icon, and the name 'BoardGame'. To its right, it shows 'Last Success' at 7 hr 10 min, 'Last Failure' at N/A, and 'Last Duration' at 1 min 3 sec. An orange arrow points from the text 'Click on the Pipeline "BoardGame"' to the pipeline name 'BoardGame' in the dashboard table.

Click on the Pipeline “BoardGame”



The screenshot shows the Jenkins pipeline configuration page for 'BoardGame'. On the left, there is a sidebar with options like 'Status', 'Changes', 'Build Now', 'Configure' (which is highlighted with an orange arrow), 'Delete Pipeline', 'Full Stage View', 'SonarQube', 'Stages', 'Rename', 'Maven', and 'Pipeline Syntax'. The main area shows the pipeline stages: 'Compile', 'Test', 'File System Scan', 'SonarQube Analysis', 'Quality Gate', 'Build', 'Publish To Nexus', 'Build & Tag Docker Image', 'Docker Image Scan', 'Push Docker Image', 'Deploy To Kubernetes', 'Verify the Deployment', and 'Declarative: Post Actions'. Below this is a table showing build times for various stages. To the right, there is a 'Last Successful Artifacts' section showing 'database\_service\_project-0.0.5-SNAPSHOT.jar' (45.89 MB) and 'database\_service\_project-0.0.5-SNAPSHOT.pom' (3.83 KB). Further right is a 'Test Result Trend' chart showing a single red dot at the top, indicating all tests passed. Another orange arrow points from the text 'Click on the Pipeline "BoardGame"' to the 'Configure' button in the sidebar.

Click on “Configure”

The screenshot shows the Jenkins configuration interface for a project named 'BoardGame'. The left sidebar has tabs for 'General', 'Triggers', 'Pipeline', and 'Advanced'. The 'General' tab is selected. The main area is titled 'General' and contains a 'Description' field which is currently empty. Below it is a section for 'Log Rotation' with a checked checkbox for 'Discard old builds'. Under 'Days to keep builds', there is a text input field containing '2'. Under 'Max # of builds to keep', there is another text input field also containing '2'. A blue arrow points from the text 'Under "General", select "GitHub project"' to the 'GitHub project' checkbox in the 'Advanced' section. At the bottom are 'Save' and 'Apply' buttons.

Scroll down

This screenshot shows the same Jenkins configuration page for 'BoardGame', but the 'Advanced' section is now expanded. The 'Max # of builds to keep' field still contains '2'. In the 'Advanced' section, there is a list of checkboxes: 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'GitHub project' (which has an orange arrow pointing to it), 'Pipeline speed/durability override', 'Preserve stashes from completed builds', 'This project is parameterized', and 'Throttle builds'. Below this is a 'Triggers' section with two checkboxes: 'Build after other projects are built' and 'Build periodically'. At the bottom are 'Save' and 'Apply' buttons.

Under “General”, select “GitHub project”

The screenshot shows the Jenkins configuration page for a job named "BoardGame". The "General" tab is selected. Under the "GitHub project" section, the "Project url" field contains "https://github.com/ebotsidneysmith/Boardgame.git". An orange arrow points from the text input field towards the "Save" button at the bottom.

Enter the URL of the GitHub repository

The screenshot shows the Jenkins configuration page for a job named "BoardGame". The "General" tab is selected. Under the "GitHub project" section, the "Project url" field contains "https://github.com/ebotsidneysmith/Boardgame.git". An orange arrow points from the "Save" button at the bottom towards the button itself.

Click on “Save”

[Not secure 54.226.220.79:8080/job/BoardGame/](#)

**Jenkins / BoardGame**

**Status** **BoardGame** **Add description**

- </> Changes
- ▷ Build Now
- ⚙ Configure
- 🗑 Delete Pipeline
- 🔍 Full Stage View
- /github GitHub
- .sonar SonarQube
- ⌚ Stages
- ✍ Rename
- ✗ Maven
- Pipeline Syntax

Last Successful Artifacts

- database\_service\_project-0.0.5-SNAPSHOT.jar 45.89 MB [view](#)
- database\_service\_project-0.0.5-SNAPSHOT.pom 3.83 kB [view](#)

**Test Result Trend**

Passed Skipped Failed

#21 #22

mpile	Test	File System Scan	SonarQube Analysys	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan	Push Docker Image	Deploy To Kubernetes	Verify the Deployment	Declarative: Post Actions
3s	15s	1s	13s	458ms	16s	19s	1s	598ms	2s	843ms	997ms	1s
3s	15s	1s	13s	456ms (paused for 2s)	16s	19s	1s	597ms	2s	970ms	992ms	1s
3s	15s	973ms	13s	461ms (paused for 2s)	15s	19s	1s	600ms	2s	716ms	1s	

**SonarQube Quality Gate**

We have successfully added the URL of our GitHub repository to Jenkins.

### 7.1.2 Modify Trigger on Jenkins

Now, we are going to modify the trigger.

[Not secure 54.226.220.79:8080/job/BoardGame/](#)

**Jenkins / BoardGame**

**Status** **BoardGame** **Add description**

- </> Changes
- ▷ Build Now
- ⚙ **Configure**
- 🗑 Delete Pipeline
- 🔍 Full Stage View
- /github GitHub
- .sonar SonarQube
- ⌚ Stages
- ✍ Rename
- ✗ Maven
- Pipeline Syntax

Last Successful Artifacts

- database\_service\_project-0.0.5-SNAPSHOT.jar 45.89 MB [view](#)
- database\_service\_project-0.0.5-SNAPSHOT.pom 3.83 kB [view](#)

**Test Result Trend**

Passed Skipped Failed

#21 #22

mpile	Test	File System Scan	SonarQube Analysys	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan	Push Docker Image	Deploy To Kubernetes	Verify the Deployment	Declarative: Post Actions
3s	15s	1s	13s	458ms	16s	19s	1s	598ms	2s	843ms	997ms	1s
3s	15s	1s	13s	456ms (paused for 2s)	16s	19s	1s	597ms	2s	970ms	992ms	1s
3s	15s	973ms	13s	461ms (paused for 2s)	15s	19s	1s	600ms	2s	716ms	1s	

**SonarQube Quality Gate**

Click on “Configure” again

The screenshot shows the Jenkins configuration page for a job named "BoardGame". The left sidebar has tabs for "General", "Triggers", "Pipeline", and "Advanced", with "General" currently selected. The main area is titled "General" and contains fields for "Description" (empty), "Plain text" and "Preview" buttons, and a checkbox for "Discard old builds". Under "Strategy", there are sections for "Log Rotation" (with dropdown for "Days to keep builds" and "Max # of builds to keep"), and "Advanced" (with a dropdown). At the bottom are "Save" and "Apply" buttons.

Click on “Triggers” on the left-hand side

The screenshot shows the Jenkins configuration page for a job named "BoardGame". The left sidebar has tabs for "General", "Triggers", "Pipeline", and "Advanced", with "Triggers" currently selected. The main area is titled "Triggers" and describes how to set up automated actions based on events like code changes or scheduled times. It lists several triggers: "Build after other projects are built", "Build periodically", "GitHub hook trigger for GITScm polling" (which is highlighted with a red arrow), "Poll SCM", and "Trigger builds remotely (e.g., from scripts)". Below this is a "Pipeline" section with a "Definition" dropdown set to "Pipeline script". A code editor shows Groovy script for defining the pipeline. At the bottom are "Save" and "Apply" buttons.

Under “Triggers”, select “GitHub hook trigger for GITScm polling”

Not secure 54.146.239.233:8080/job/BoardGame/configure

**Configure**

**Triggers**

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- Build after other projects are built ?
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?
- Trigger builds remotely (e.g., from scripts) ?

**Pipeline**

Define your Pipeline using Groovy directly or pull it from source control.

**Definition**

Pipeline script

```
Script ?
1~ pipeline {
2     agent any
3
4~     tools {
5         jdk 'jdk17'
6         maven "maven3"
7     }
8 }
```

**Buttons:** Save | Apply

Click on “Save”

Not secure 54.226.220.79:8080/job/BoardGame/

**BoardGame**

- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- GitHub
- SonarQube
- Stages
- Rename
- Maven
- Pipeline Syntax
- GitHub Hook Log

**Builds**

Build	Test	File System Scan	SonarQube Analysys	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan	Push Docker Image	Deploy To Kubernetes	Verify the Deployment	Declarative: Post Actions
#22 1:28 AM	3s	15s	1s	13s	458ms	16s	19s	1s	590ms	2s	843ms	997ms
#21 1:12 AM	3s	15s	973ms	13s	461ms (paused for 2s)	15s	19s	1s	600ms	2s	716ms	1s

**SonarQube Quality Gate**

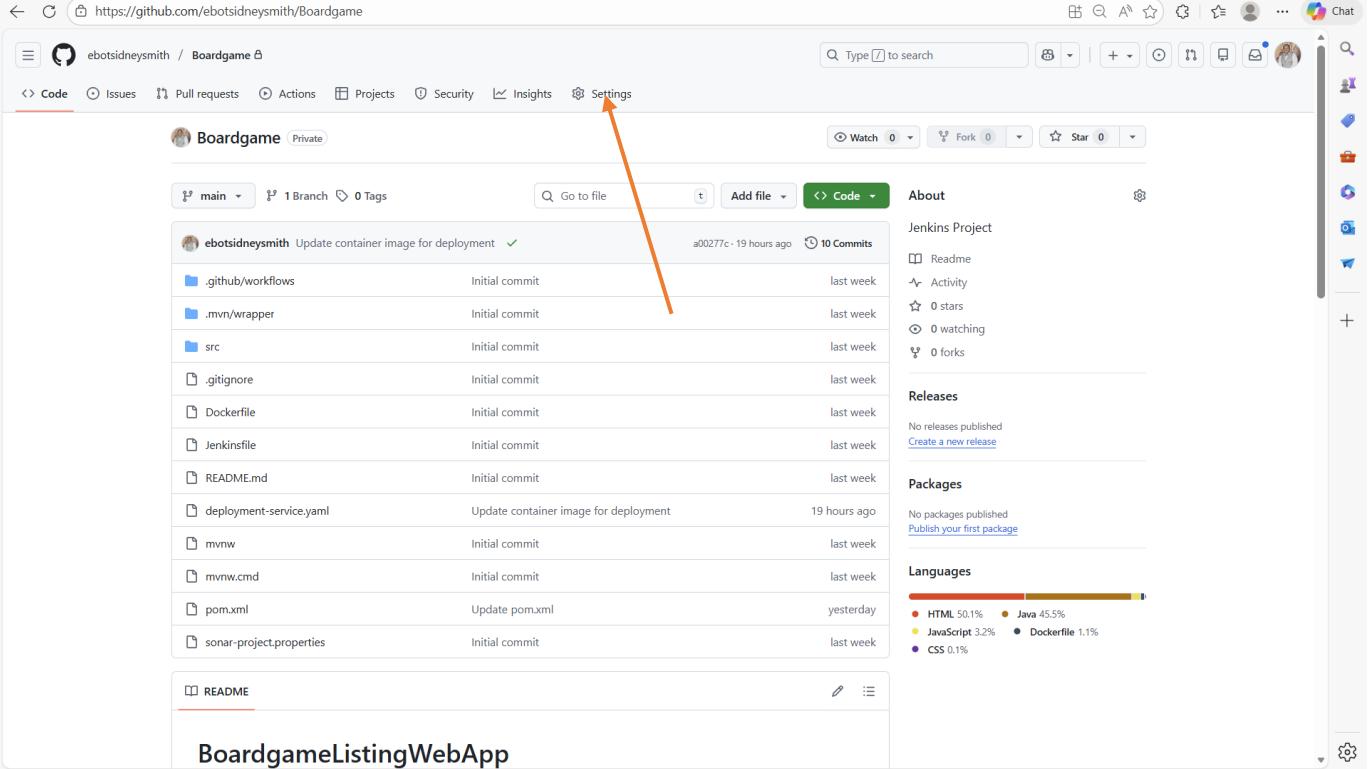
BoardGame Passed

server-side processing: Success

We have added the trigger to our Jenkins pipeline.

### 7.1.3 Make Modification on GitHub

Now, we have to modify the GitHub repository. Go to the GitHub repository



https://github.com/ebotsidneysmith/Boardgame

ebotsidneysmith / Boardgame

Code Issues Pull requests Actions Projects Security Insights Settings

Boardgame Private

main 1 Branch 0 Tags

Go to file Add file Code

ebotsidneysmith Update container image for deployment ✓ a00277c - 19 hours ago 10 Commits

.github/workflows Initial commit last week

.mvnw Initial commit last week

src Initial commit last week

.gitignore Initial commit last week

Dockerfile Initial commit last week

Jenkinsfile Initial commit last week

README.md Initial commit last week

deployment-service.yaml Update container image for deployment 19 hours ago

mvnw Initial commit last week

mvnw.cmd Initial commit last week

pom.xml Update pom.xml yesterday

sonar-project.properties Initial commit last week

README

BoardgameListingWebApp

About Jenkins Project

Readme Activity 0 stars 0 watching 0 forks

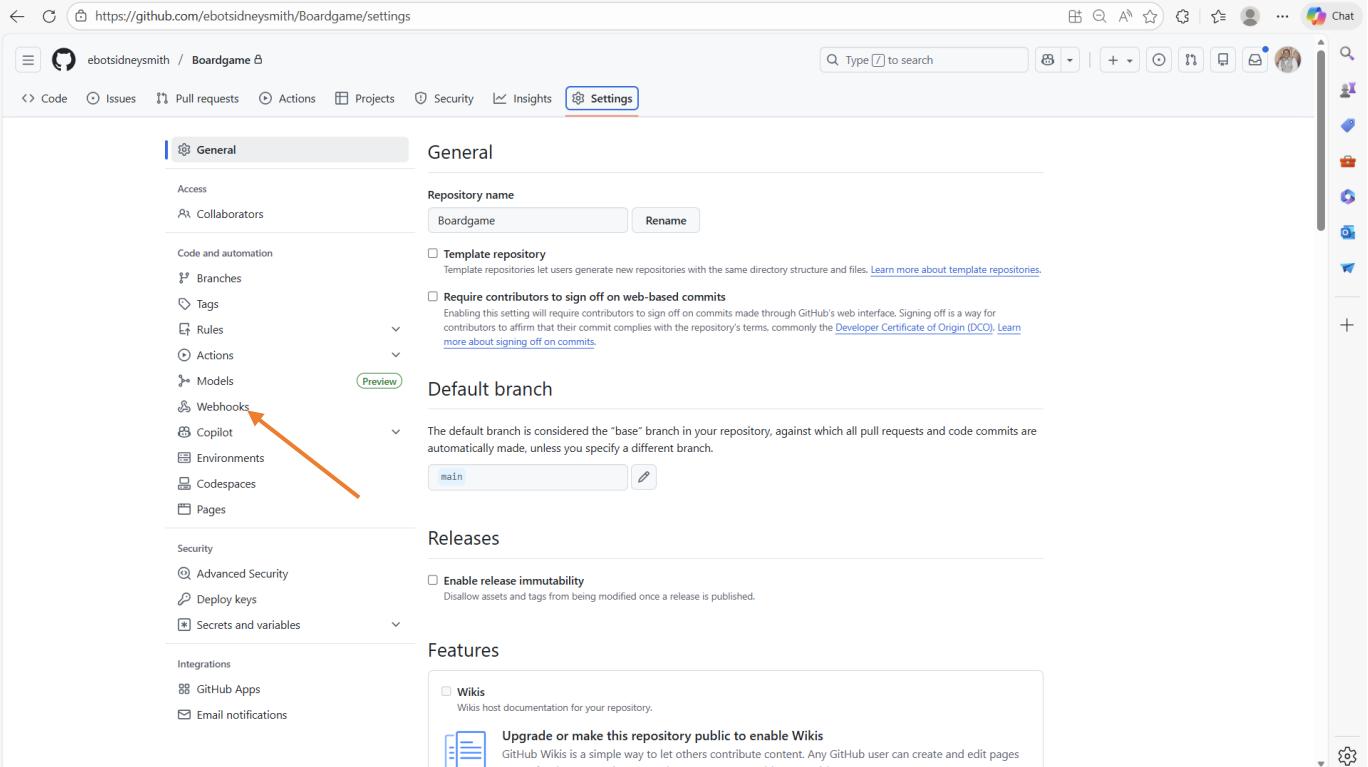
Releases No releases published Create a new release

Packages No packages published Publish your first package

Languages

HTML 50.1% Java 45.5% JavaScript 3.2% Dockerfile 1.1% CSS 0.1%

Click on “Settings”



https://github.com/ebotsidneysmith/Boardgame/settings

ebotsidneysmith / Boardgame

Code Issues Pull requests Actions Projects Security Insights Settings

General

Access Collaborators

Code and automation

Branches Tags Rules Actions Models Webhooks Copilot Environments Codespaces Pages

General

Repository name Boardgame Rename

Template repository Repository templates let users generate new repositories with the same directory structure and files. Learn more about template repositories.

Require contributors to sign off on web-based commits Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the [Developer Certificate of Origin \(DCO\)](#). Learn more about signing off on commits.

Default branch

The default branch is considered the “base” branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

main

Releases

Enable release immutability Disallow assets and tags from being modified once a release is published.

Features

Wikis Wikis host documentation for your repository.

Upgrade or make this repository public to enable Wikis GitHub Wikis is a simple way to let others contribute content. Any GitHub user can create and edit pages to use for documentation, examples, support, or anything you wish.

Click on “webhook” on the left-hand side

A screenshot of the GitHub settings interface for a repository named "Boardgame". The left sidebar shows various settings categories like General, Access, Collaborators, and Code and automation. Under "Code and automation", the "Webhooks" option is selected and highlighted with a green preview badge. At the top right of the main content area, there is a button labeled "Add webhook". An orange arrow points from the text "Click on 'Add Webhook'" to this button.

Click on “Add Webhook”

A screenshot of a "Confirm access" modal window. It displays a user profile picture and the text "Signed in as @ebotsidneysmith". Below this is a password input field with the placeholder "Password" and a "Forgot password?" link. A large orange arrow points from the text "Enter the password and click on confirm" to the "Password" input field. At the bottom of the modal, there is a green "Confirm" button. A small tip at the bottom of the modal states: "Tip: You are entering sudo mode. After you've performed a sudo-protected action, you'll only be asked to re-authenticate again after a few hours of inactivity."

Enter the password and click on confirm

The screenshot shows the GitHub settings interface for a repository named 'Boardgame'. The left sidebar contains navigation links for General, Access, Collaborators, Code and automation, Security, Insights, and Settings. Under 'Code and automation', the 'Webhooks' link is selected, indicated by a green preview button. The main content area is titled 'Webhooks / Add webhook'. It includes instructions about sending POST requests with event details. The 'Payload URL' field is set to 'https://example.com/postreceive'. The 'Content type' dropdown is set to 'application/x-www-form-urlencoded'. There is a 'Secret' input field, which is currently empty. The 'SSL verification' section has the 'Enable SSL verification' radio button selected. The 'Which events would you like to trigger this webhook?' section has the 'Just the push event.' radio button selected. The 'Active' checkbox is checked. At the bottom is a green 'Add webhook' button.

**On Payload URL:** Enter the Jenkins URL/github-webhook/

That is `http://54.226.220.79:8080/github-webhook/`

**On Content Type:** Select **application/json**

This screenshot is identical to the one above, showing the GitHub settings page for 'Boardgame'. The 'Webhooks' section is active. The 'Payload URL' is now set to 'http://54.146.239.233:8080/github-webhook/'. The 'Content type' is set to 'application/json'. The 'Secret' field is empty. The 'SSL verification' section has the 'Enable SSL verification' radio button selected. The 'Which events would you like to trigger this webhook?' section has the 'Just the push event.' radio button selected. The 'Active' checkbox is checked. A red arrow points to the green 'Add webhook' button at the bottom of the form.

**Click on "Add Webhook"**

The screenshot shows the GitHub repository settings page for 'ebotsidneysmith/Boardgame'. The 'Webhooks' section is active, displaying a single webhook entry. The URL is listed as 'http://54.146.239.233:8080/github... (push)'. A note indicates that this hook has never been triggered. There are 'Edit' and 'Delete' buttons for the webhook.

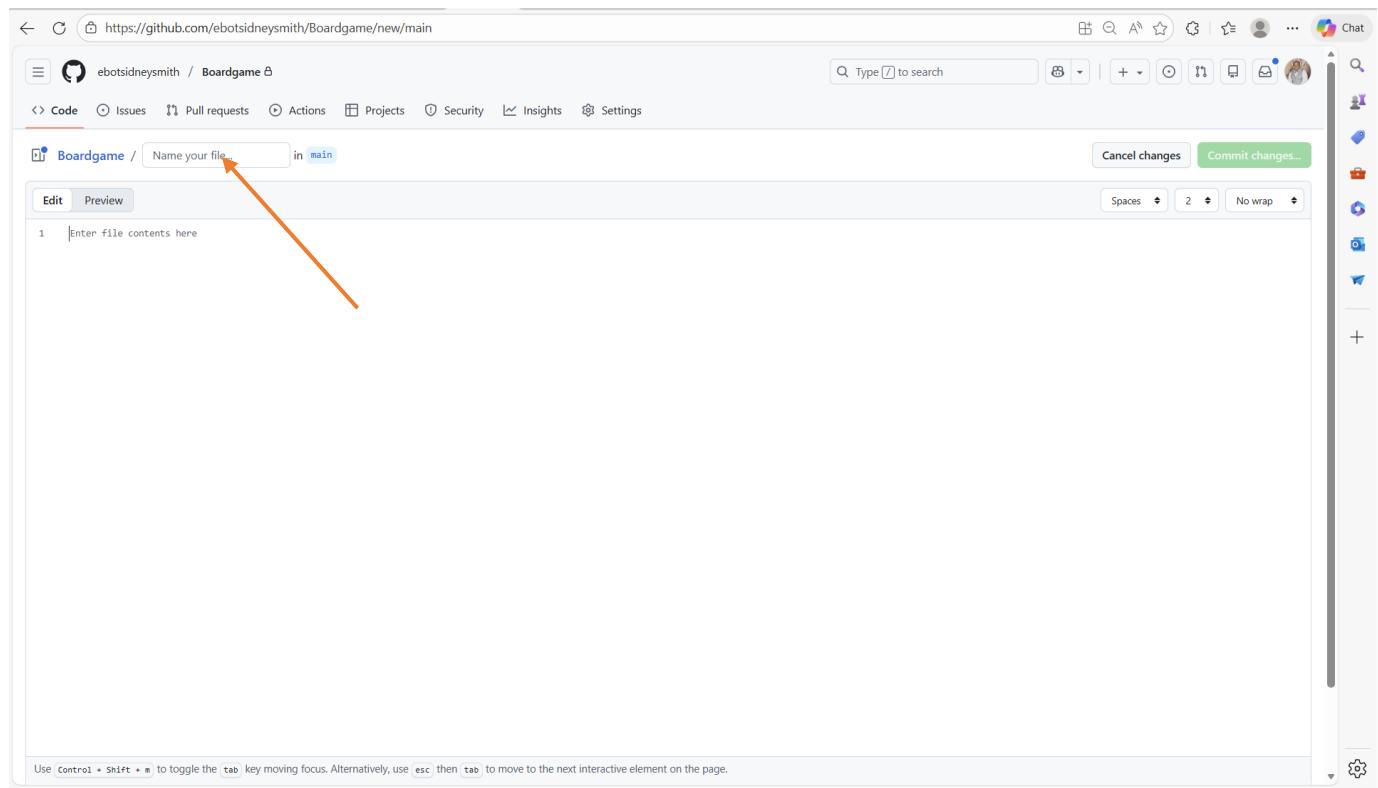
Now, let us test if the pipeline has been automated

#### 7.1.4 Verification

Let me add a new text file called “**test3.txt**” to our GitHub repository. Then commit to see if the pipeline will be triggered automatically.

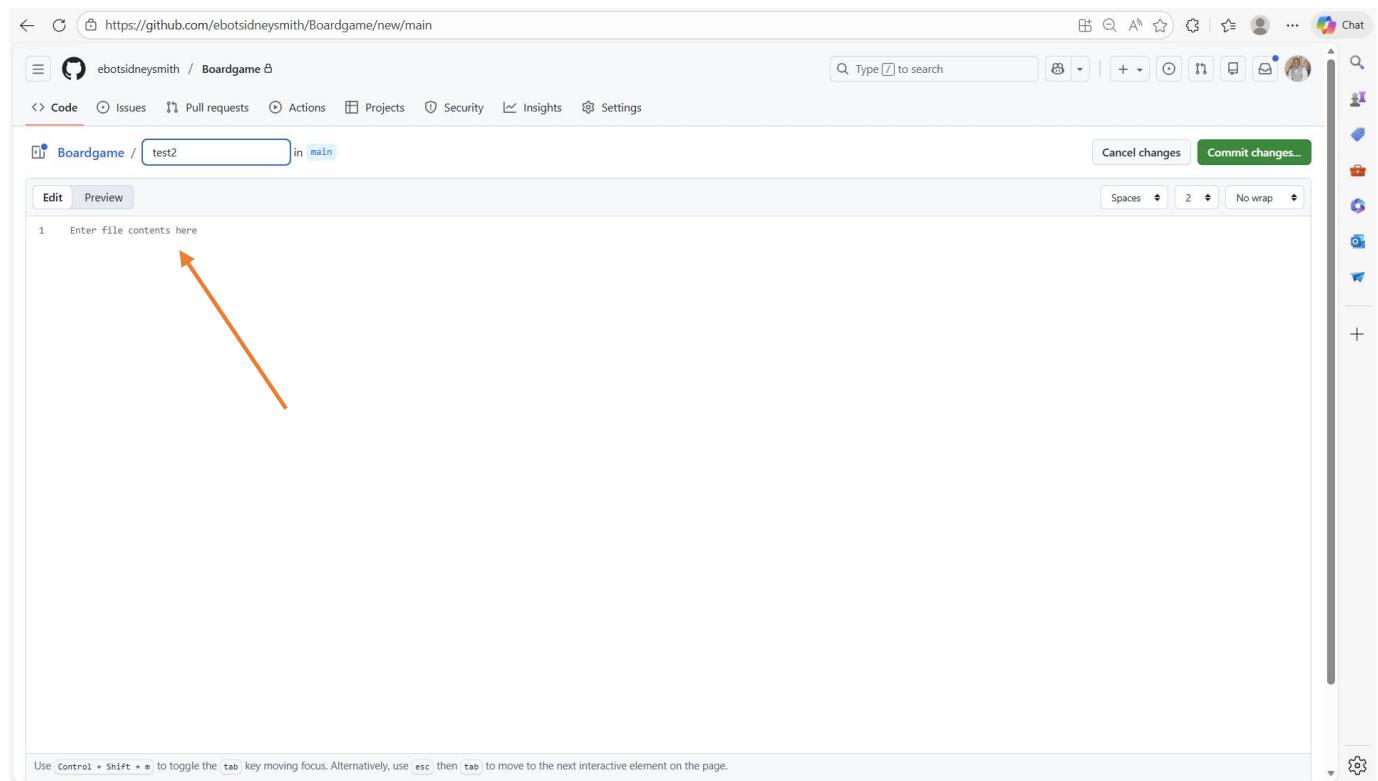
The screenshot shows the GitHub repository page for 'ebotsidneysmith/Boardgame'. A new commit has been pushed, containing files like '.github/workflows', '.mvnw', 'src', '.gitignore', 'Dockerfile', 'Jenkinsfile', 'README.md', 'deployment-service.yaml', 'mvnw', 'mvnw.cmd', 'pom.xml', and 'sonar-project.properties'. The commit was made by 'ebotsidneysmith' and is labeled 'Update container image for deployment'. The commit was made 19 hours ago. To the right, the 'Jenkins Project' sidebar is visible, showing '10 Commits' and a Jenkins logo. The 'Languages' section at the bottom indicates the codebase is primarily composed of HTML and Java.

Click on “Add file” and select “create a new file”



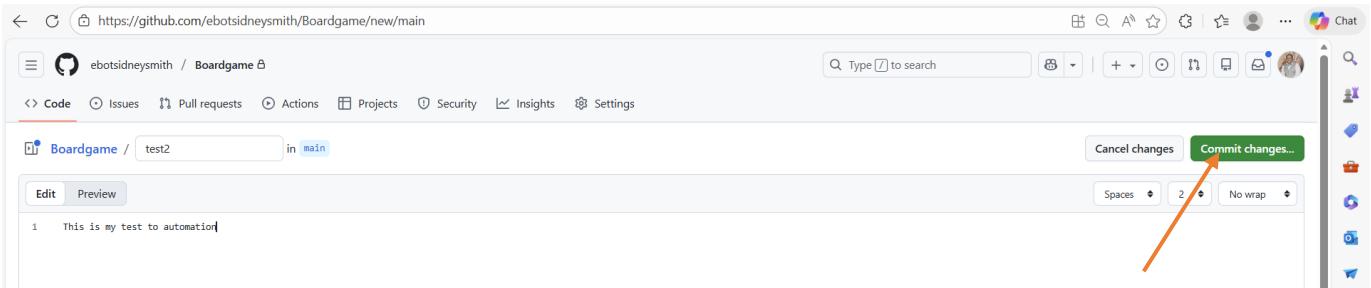
The screenshot shows the GitHub interface for creating a new file. The URL is https://github.com/ebotsidneysmith/Boardgame/new/main. The 'Code' tab is selected. In the top navigation bar, there is a search bar with placeholder text 'Type / to search' and several icons for issues, pull requests, actions, projects, security, insights, and settings. Below the navigation, the repository name 'Boardgame' and the path '/ in main' are shown. A red arrow points from the text 'Name your file...' to the input field where 'test2' is typed. The main area contains a text editor with the placeholder 'Enter file contents here' and a status bar at the bottom with keyboard shortcuts for focus and tab navigation.

Give the file the name “**test2**”

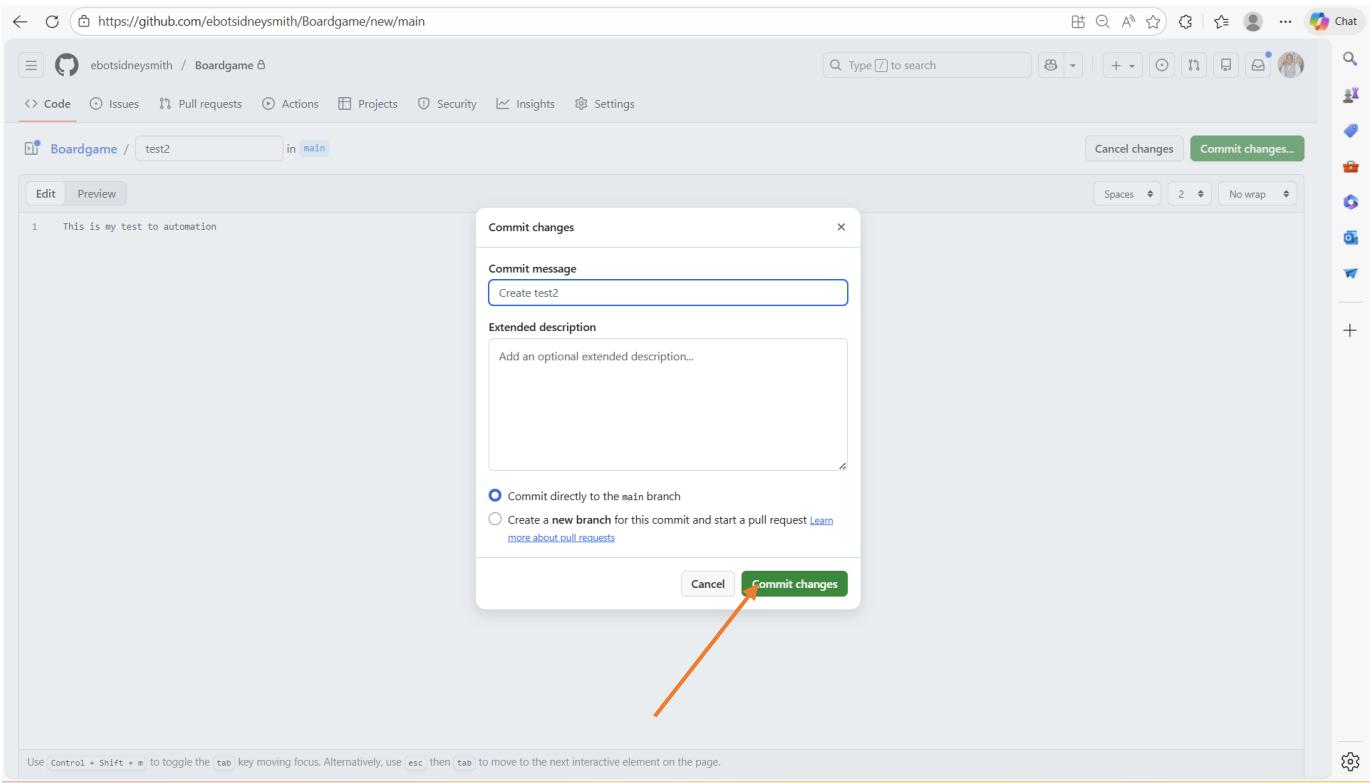


The screenshot shows the GitHub interface after naming the file 'test2'. The URL is https://github.com/ebotsidneysmith/Boardgame/new/main. The 'Code' tab is selected. The repository name 'Boardgame' and the path '/ test2 in main' are shown. A red arrow points from the text 'Enter file contents here' to the text area where the file content will be entered. The main area contains a text editor with a placeholder 'Enter file contents here' and a status bar at the bottom with keyboard shortcuts for focus and tab navigation.

Then, enter some text. I will enter “**This is my test to automation**”



Click on “Commit Changes” and we head back to our Jenkins GUI



Click on “Commit Changes” again

ebotsidneysmith / Boardgame

Name Last commit message Last commit date

- .github/workflows Initial commit last week
- .mvn/wrapper Initial commit last week
- src Initial commit last week
- .gitignore Initial commit last week
- Dockerfile Initial commit last week
- Jenkinsfile Initial commit last week
- README.md Initial commit last week
- deployment-service.yaml Update container image for deployment 19 hours ago
- mvnw Initial commit last week
- mvnw.cmd Initial commit last week
- pom.xml Update pom.xml yesterday
- sonar-project.properties Initial commit last week
- test2 Create test2 now

README.md

**Boardgame listing WebApp**

Go now to Jenkins and check if the Pipeline has started building

**BoardGame**

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

GitHub

SonarQube

Stages

Rename

Maven

Pipeline Syntax

GitHub Hook Log

Last Successful Artifacts

database\_service\_project-0.0.5-SNAPSHOT.jar 45.89 MiB view

database\_service\_project-0.0.5-SNAPSHOT.pom 3.83 KiB view

**Test Result Trend**

Passed Skipped Failed

#21 #22

**Stage View**

Average stage times: (full run time: ~1min 20s)

Declarative: Tool Install	Git Checkout	Compile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Published To Nexus	Build & Tag Docker Image
203ms	552ms	3s	10s	1s	13s	458ms	16s	19s	1s

Builds

Builds

Filter

Today

#23 1:59 AM

#22 1:28 AM

#21 1:12 AM

You can see that the Jenkins pipeline has started running automatically.

Not secure 54.226.220.79:8080/job/BoardGame/

**Jenkins / BoardGame**

Full Stage View GitHub SonarQube Stages Rename Maven Pipeline Syntax GitHub Hook Log

Builds Filter Today #23 1:59 AM #22 1:28 AM #21 1:12 AM

Build Pipeline

npile	Test	File System Scan	SonarQube Analysis	Quality Gate	Build	Publish To Nexus	Build & Tag Docker Image	Docker Image Scan	Push Docker Image	Deploy To Kubernetes	Verify the Deployment	Declarative: Post Actions
3s	15s	1s	13s	453ms	16s	19s	1s	596ms	3s	888ms	1s	1s
3s	15s	1s	13s	443ms (paused for 1s)	16s	19s	1s	592ms	3s	980ms	1s	1s
3s	15s	1s	13s	456ms (paused for 2s)	16s	19s	1s	597ms	2s	970ms	992ms	1s
3s	15s	973ms	13s	461ms (paused for 2s)	15s	19s	1s	600ms	2s	716ms	1s	1s

**SonarQube Quality Gate**

BoardGame Passed server-side processing: Success

The build is successful. Click on “Passed” to see the SonarQube Analysis result.

Not secure 34.203.223.19:9000/dashboard?id=BoardGame

You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately. Learn More

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects... A

BoardGame main Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

QUALITY GATE STATUS **Passed** All conditions passed.

MEASURES

New Code	Overall Code
Since January 11, 2026 Started 5 hours ago	
0 New Bugs	Reliability A
0 New Vulnerabilities	Security A
0 New Security Hotspots	Reviewed Security Review A
0 Added Debt	0 New Code Smells Maintainability A
— Coverage on 0 New Lines to cover	
— Duplications on 0 New Lines	

Last analysis of this Branch had 3 warnings January 11, 2026 at 9:00 PM Version not provided

Not secure 34.203.223.19:9000/dashboard?id=BoardGame

You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately. [Learn More](#)

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

BoardGame main Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Last analysis of this Branch had 3 warnings January 11, 2026 at 9:00 PM Version not provided

Coverage on 0 New Lines to cover Duplications on 0 New Lines

ACTIVITY

Choose graph type: Issues Bugs Code Smells Vulnerabilities

new code

06:45 07 PM 07:15 07:30 07:45 08 PM 08:15 08:30 08:45

Activity

January 11, 2026 at 9:00 PM not provided

January 11, 2026 at 8:13 PM Quality Gate: Passed

January 11, 2026 at 6:39 PM Quality Gate: Failed

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 - v9.9.8 (build 100196) NO LONGER ACTIVE - LGPL v3 - Community - Documentation - Plugins - Web API