```
sudo apt update
```

// Download & Install Java 21.

```
sudo apt install fontconfig openjdk-21-jre -y
```

// Download & store GPG key for Jenkins Debian package.

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

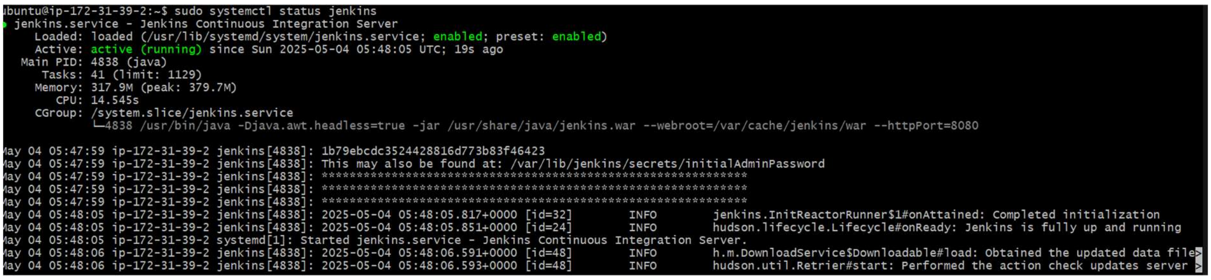
// To trust package repo and enable apt install.

```
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
```

// To install Jenkins.

```
sudo apt-get update
sudo apt-get install jenkins -y
```

// Verify status of Jenkins.

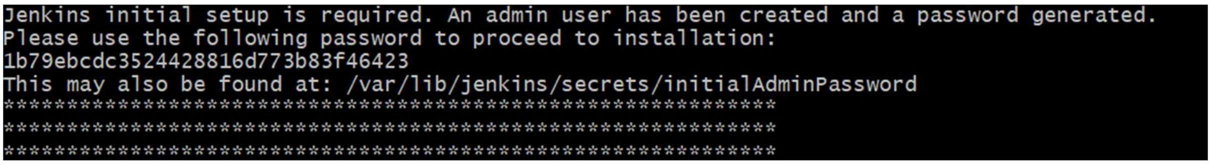


```
ubuntu@ip-172-31-39-2:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-05-04 05:48:05 UTC; 19s ago
   Main PID: 4838 (java)
      Tasks: 41 (limit: 1129)
     Memory: 317.9M (peak: 379.7M)
        CPU: 14.545s
     CGroup: /system.slice/jenkins.service
             └─4838 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

May 04 05:47:59 ip-172-31-39-2 jenkins[4838]: 1b79ebcdc3524428816d773b83f46423
May 04 05:47:59 ip-172-31-39-2 jenkins[4838]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
May 04 05:47:59 ip-172-31-39-2 jenkins[4838]: *************************************************************
May 04 05:47:59 ip-172-31-39-2 jenkins[4838]: *************************************************************
May 04 05:47:59 ip-172-31-39-2 jenkins[4838]: *************************************************************
May 04 05:48:05 ip-172-31-39-2 jenkins[4838]: 2025-05-04 05:48:05.817+0000 [id=32]       INFO       jenkins.InitReactorRunner$1#onAttained: Completed initialization
May 04 05:48:05 ip-172-31-39-2 jenkins[4838]: 2025-05-04 05:48:05.851+0000 [id=24]       INFO       hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
May 04 05:48:05 ip-172-31-39-2 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
May 04 05:48:06 ip-172-31-39-2 jenkins[4838]: 2025-05-04 05:48:06.591+0000 [id=48]       INFO       h.m.DownloadService$Downloadable#load: Obtained the updated data file>
May 04 05:48:06 ip-172-31-39-2 jenkins[4838]: 2025-05-04 05:48:06.593+0000 [id=48]       INFO       hudson.util.Retrier#start: Performed the action check updates server >
```

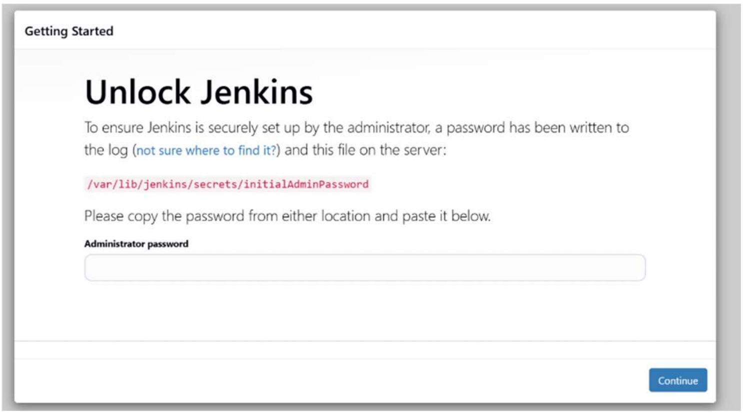
// Check Journal logs for Initial Password.

```
journalctl -u jenkins
```



```
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
1b79ebcdc3524428816d773b83f46423
This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
*************************************************************
*************************************************************
*************************************************************
```

// Go to http://<server-ip>:8080 & use the initial password.


Getting Started

Unlock Jenkins

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

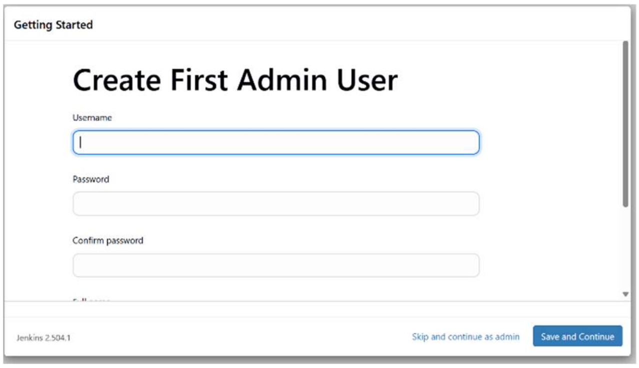
```
/var/lib/jenkins/secrets/initialAdminPassword
```

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

Administrator password

// Install suggested plugins or select as per your requirements.

// Create your admin user


Getting Started

Create First Admin User

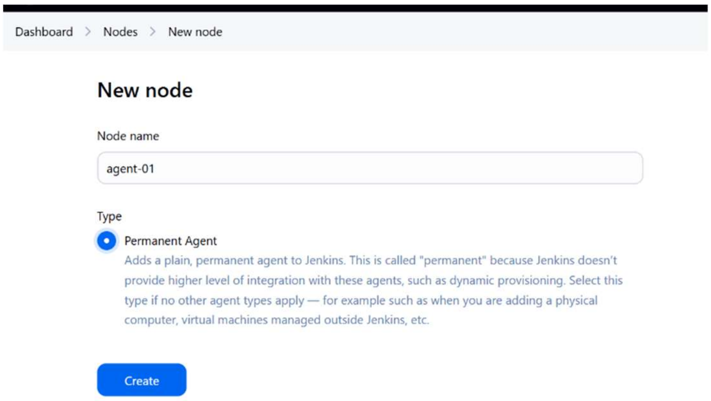
Username

Password

Confirm password

Jenkins 2.504.1

// Add agent-nodes to Jenkins


Dashboard > Nodes > New node

New node

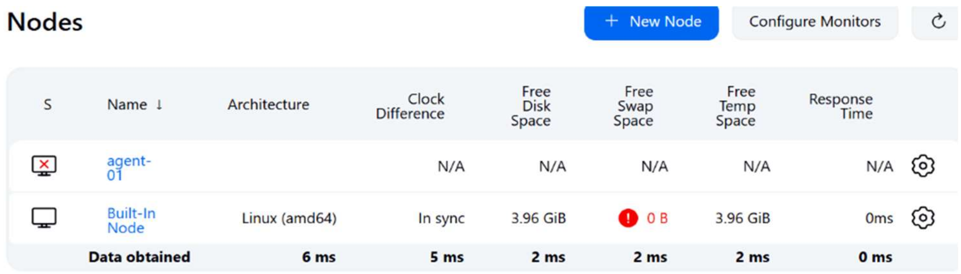
Node name: agent-01

Type

● Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

// Save without any changes & it looks like this.



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time	
☒	agent-01		N/A	N/A	N/A	N/A	N/A	⚙
🖥	Built-In Node	Linux (amd64)	In sync	3.96 GiB	❗ 0 B	3.96 GiB	0ms	⚙
	Data obtained	6 ms	5 ms	2 ms	2 ms	2 ms	0 ms	

// In the case of adding nodes on other servers, download and install java on that servers as well.

// Create a directory for agents & chmod permission 777 to directory.

// Run curl command in that directory.

```
curl -sO http://<server-ip>:<port>/jnlpJars/agent.jar
```

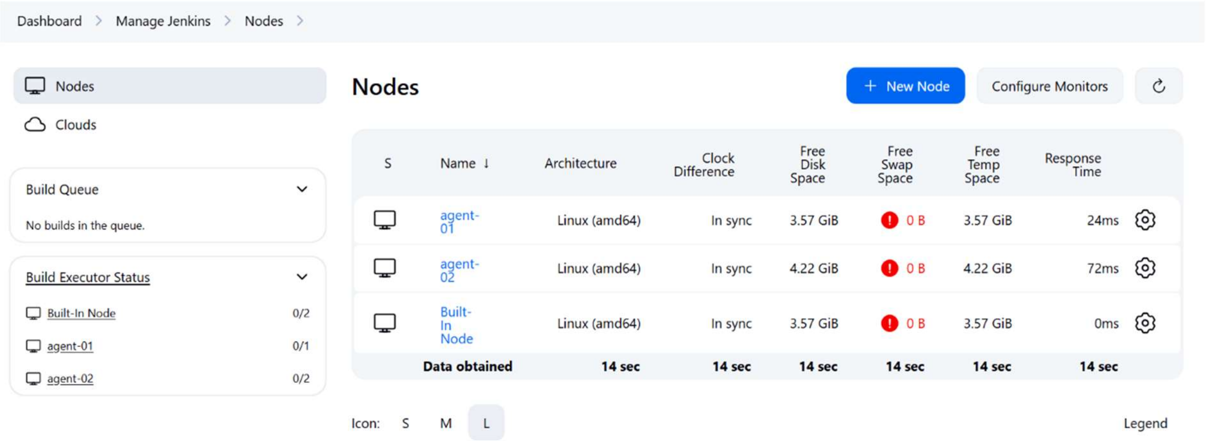
// Run agents with jar file in background with agent logs

```
nohup  java  -jar  agent.jar http://<server-ip>:<port>  -secret <Jenkins provided value>  -name “name of agent”   -webSocket   -workDir  <directory path>   > agent.log 2>&1 &
```

//If above command doesn’t work then use below one.

```
nohup java -jar agent.jar \
  -url <value> \
  -secret <value> \
  -name "agent-02" \
  -webSocket \
  -workDir <path> \
> agent.log 2>&1 &
```

// Now all agents should be synced.



Dashboard > Manage Jenkins > Nodes >

Nodes

Clouds

Build Queue

No builds in the queue.

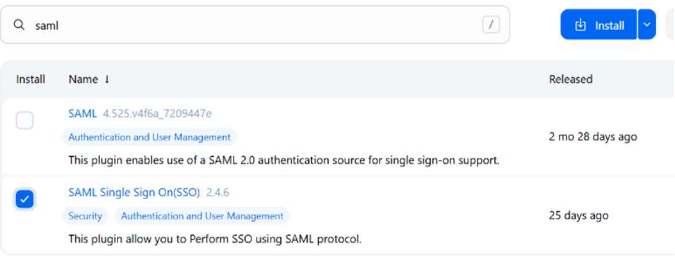
Build Executor Status

Built-In Node	0/2
agent-01	0/1
agent-02	0/2

Nodes

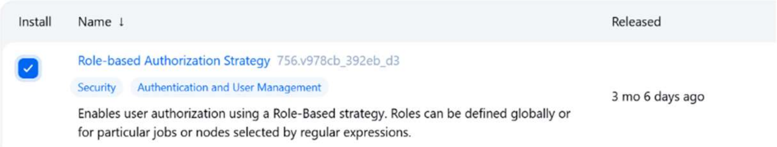
S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time	
	agent-01	Linux (amd64)	In sync	3.57 GiB	! 0 B	3.57 GiB	24ms	
	agent-02	Linux (amd64)	In sync	4.22 GiB	! 0 B	4.22 GiB	72ms	
	Built-In Node	Linux (amd64)	In sync	3.57 GiB	! 0 B	3.57 GiB	0ms	
	Data obtained	14 sec	14 sec	14 sec	14 sec	14 sec	14 sec	

// Install saml & role-based authorization plugin for SSO.



saml

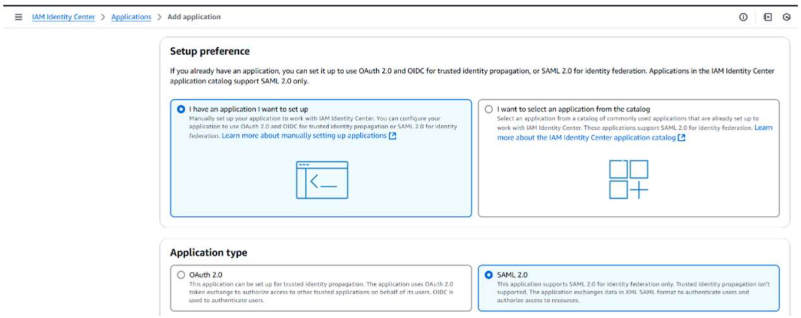
Install	Name ↓	Released
□	SAML 4.525.v4f6a_7209447e Authentication and User Management This plugin enables use of a SAML 2.0 authentication source for single sign-on support.	2 mo 28 days ago
☑	SAML Single Sign On(SSO) 2.4.6 Security Authentication and User Management This plugin allow you to Perform SSO using SAML protocol.	25 days ago



Install	Name ↓	Released
☑	Role-based Authorization Strategy 756.v978cb_392eb_d3 Security Authentication and User Management Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions.	3 mo 6 days ago

// For MiniOrange SSO setup.

Go to AWS Identity Center ---> Applications ---> Add Application
Select setup preference as I have an application I want to setup
Choose application type as SAML 2.0



IAM Identity Center > Applications > Add application

Setup preference

If you already have an application, you can set it up to use OAuth 2.0 and OIDC for trusted identity propagation, or SAML 2.0 for identity federation. Applications in the IAM Identity Center application catalog support SAML 2.0 only.

● I have an application I want to set up
Manually set up your application to work with IAM Identity Center. You can configure your application to use OAuth 2.0 and OIDC for trusted identity propagation or SAML 2.0 for identity federation. Learn more about manually setting up applications

○ I want to select an application from the catalog
Select an application from a catalog of commonly used applications that are already set up to work with IAM Identity Center. These applications support SAML 2.0 for identity federation. Learn more about the IAM Identity Center application catalog

Application type

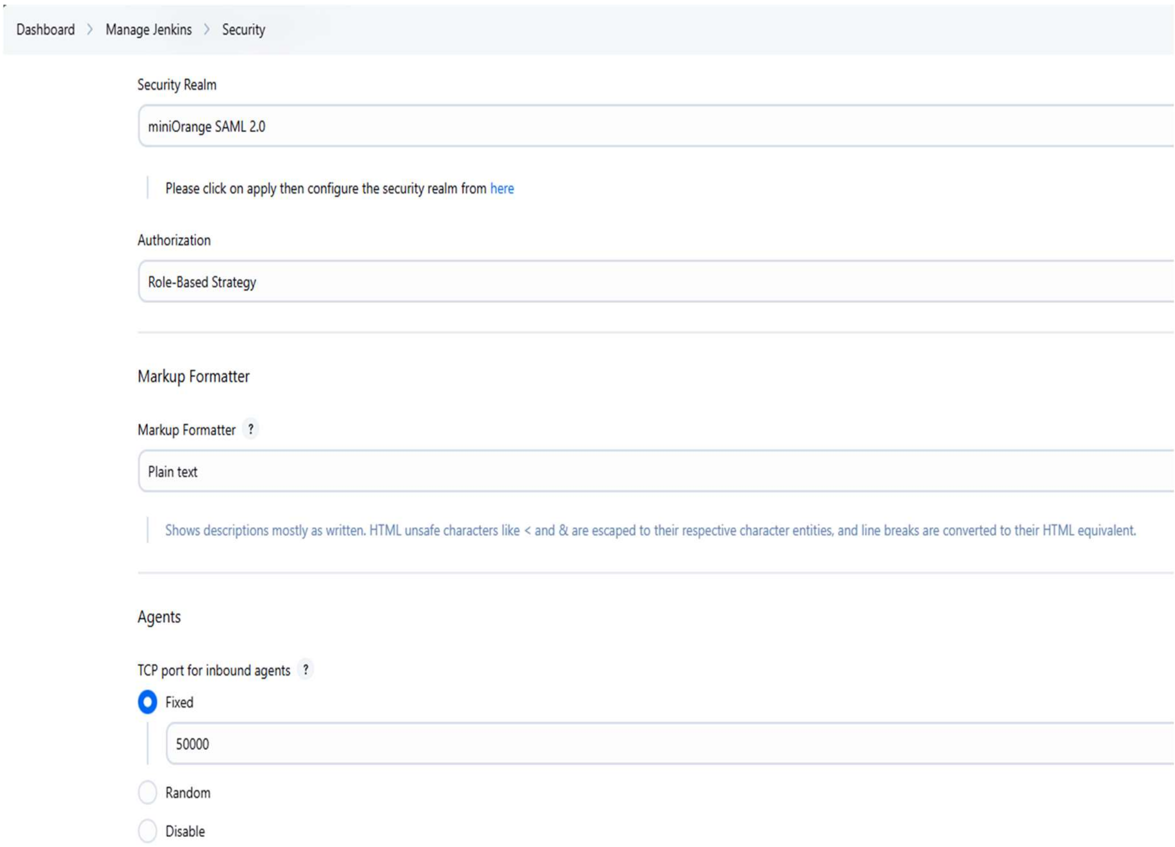
○ OAuth 2.0
This application can be set up for trusted identity propagation. The application uses OAuth 2.0 token exchange to authorize access to other trusted applications on behalf of its users. OIDC is used to authenticate users.

● SAML 2.0
This application supports SAML 2.0 for identity federation only. Trusted identity propagation isn't supported. The application exchanges data in XML SAML format to authenticate users and authorize access to resources.

// Fill in details of application.

// Download IAM Identity Center Metadata files as it would be used further.

// In Jenkins select below configurations, click save and apply.



Dashboard > Manage Jenkins > Security

Security Realm

miniOrange SAML 2.0

Please click on apply then configure the security realm from here

Authorization

Role-Based Strategy

Markup Formatter

Markup Formatter ?

Plain text

Shows descriptions mostly as written. HTML unsafe characters like < and & are escaped to their respective character entities, and line breaks are converted to their HTML equivalent.

Agents

TCP port for inbound agents ?

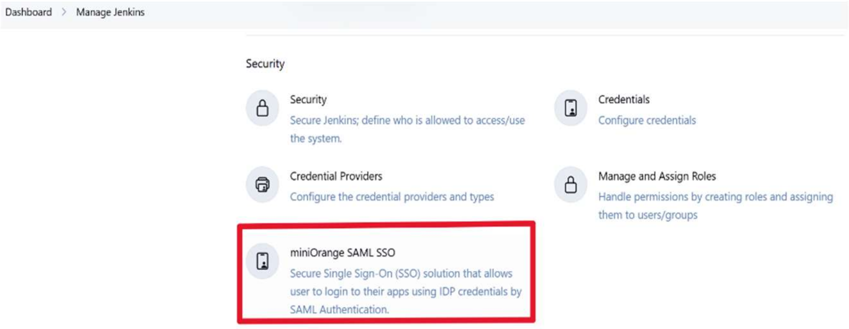
● Fixed

50000

○ Random

○ Disable

// Go to MiniOrange SSO in Jenkins.



Dashboard > Manage Jenkins

Security

Security
Secure Jenkins; define who is allowed to access/use the system.

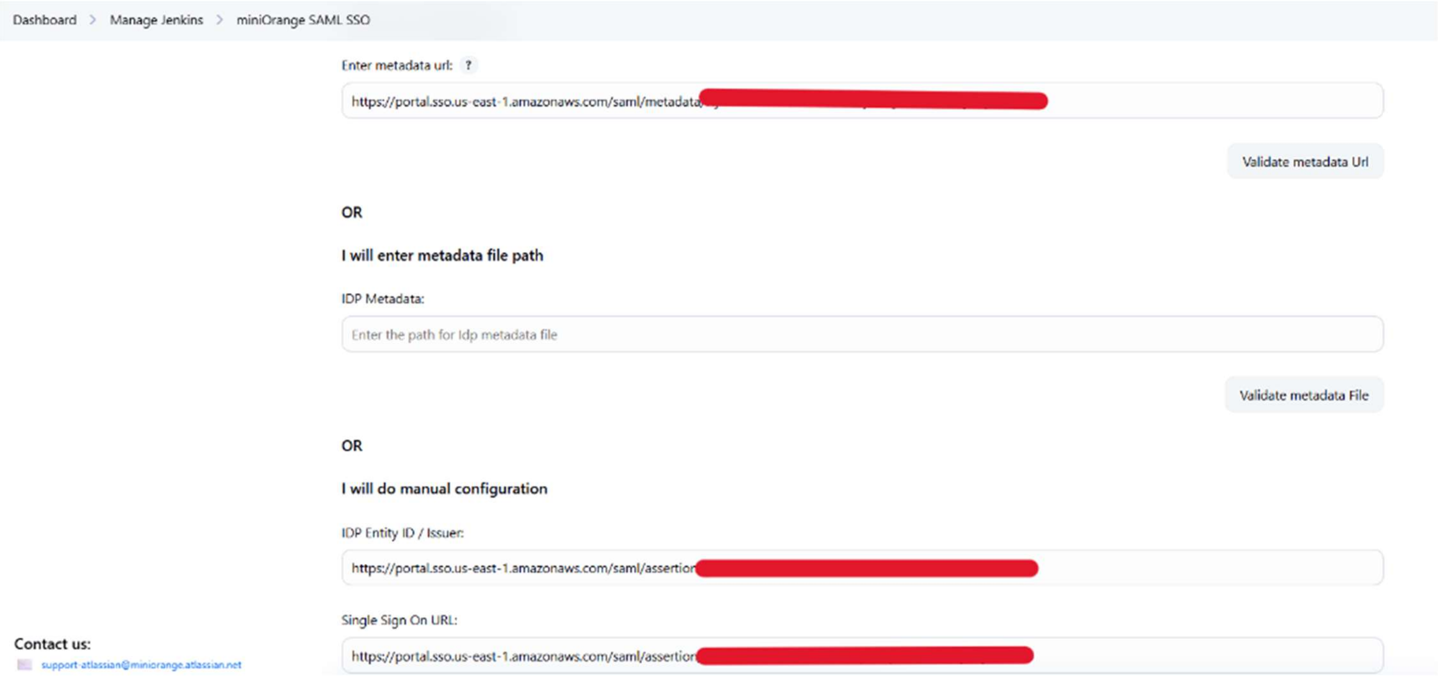
Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

Manage and Assign Roles
Handle permissions by creating roles and assigning them to users/groups

miniOrange SAML SSO
Secure Single Sign-On (SSO) solution that allows user to login to their apps using IDP credentials by SAML Authentication.

// Enter the IAM Identity Center’s metadata like this.



Dashboard > Manage Jenkins > miniOrange SAML SSO

Enter metadata url: ?

https://portal.sso.us-east-1.amazonaws.com/saml/metadata/[redacted]

Validate metadata Url

OR

I will enter metadata file path

IDP Metadata:

Enter the path for Idp metadata file

Validate metadata File

OR

I will do manual configuration

IDP Entity ID / Issuer:

https://portal.sso.us-east-1.amazonaws.com/saml/assertion[redacted]

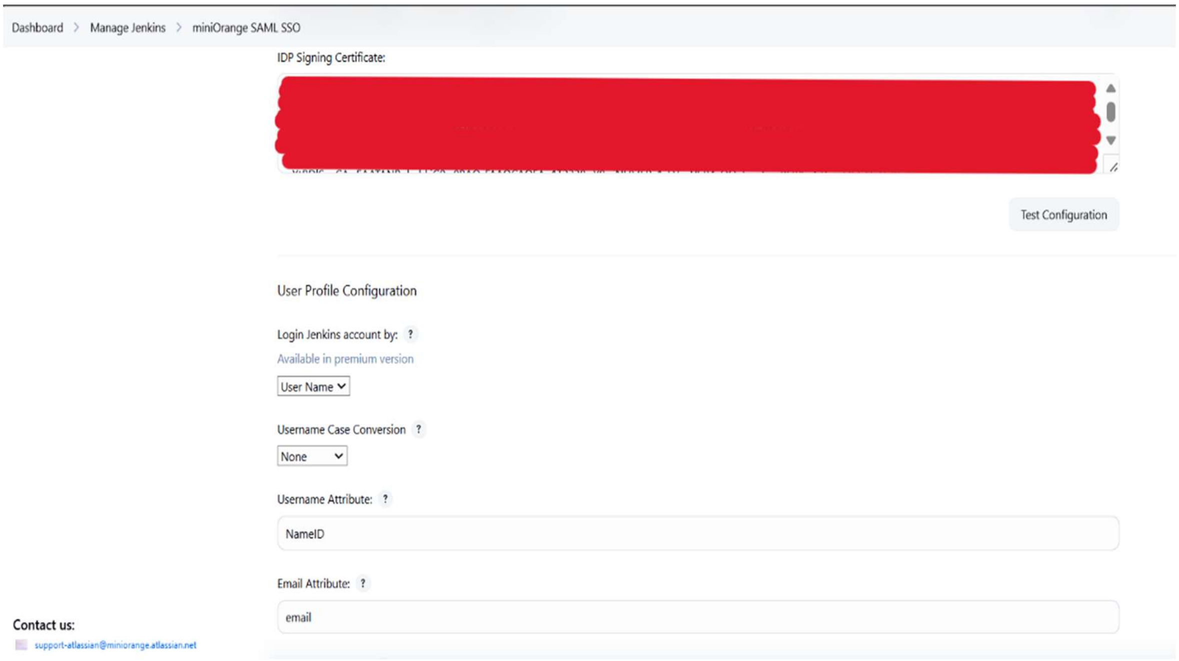
Single Sign On URL:

https://portal.sso.us-east-1.amazonaws.com/saml/assertion[redacted]

Contact us:

support-atlassian@miniorange.atlassian.net

// Upload the certificate, and add attributes in username & email id.



Dashboard > Manage Jenkins > miniOrange SAML SSO

IDP Signing Certificate:

[redacted]

Test Configuration

User Profile Configuration

Login Jenkins account by: ?

Available in premium version

User Name

Username Case Conversion ?

None

Username Attribute: ?

NameID

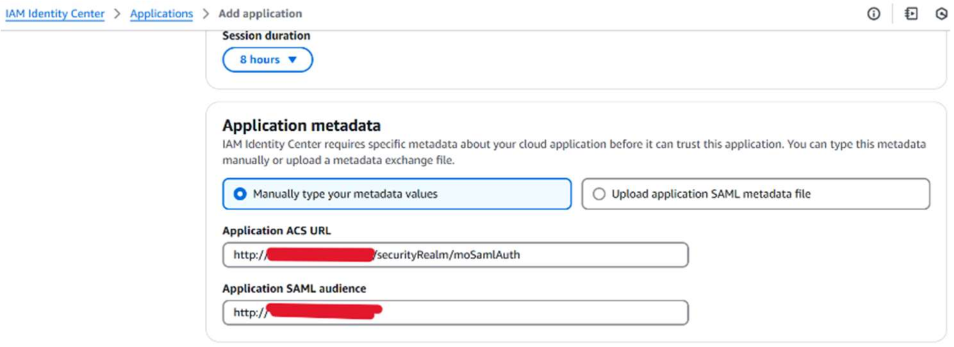
Email Attribute: ?

email

Contact us:

support-atlassian@miniorange.atlassian.net

// Go to IAM Identity Center, to make final mappings to application.



Session duration

8 hours ▼

Application metadata

IAM Identity Center requires specific metadata about your cloud application before it can trust this application. You can type this metadata manually or upload a metadata exchange file.

● Manually type your metadata values ○ Upload application SAML metadata file

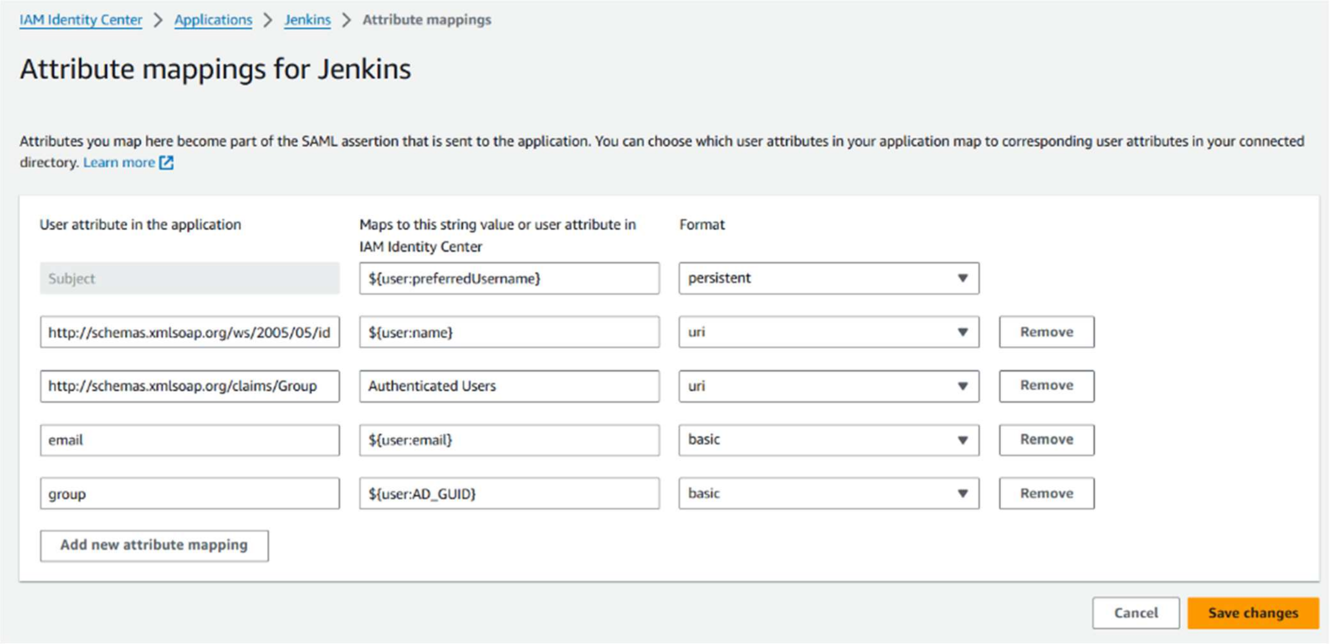
Application ACS URL

http://[redacted]/securityRealm/moSamlAuth

Application SAML audience

http://[redacted]

// Edit attribute mappings of application.



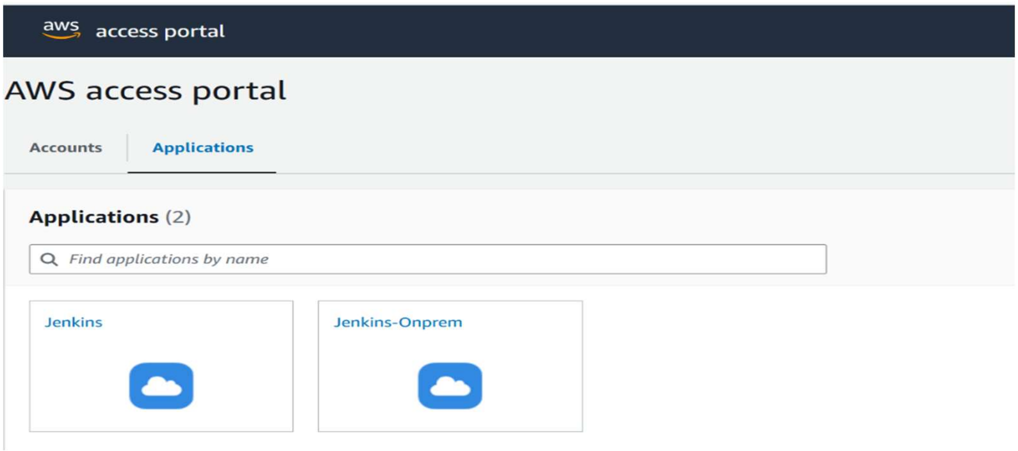
IAM Identity Center > Applications > Jenkins > Attribute mappings

Attribute mappings for Jenkins

Attributes you map here become part of the SAML assertion that is sent to the application. You can choose which user attributes in your application map to corresponding user attributes in your connected directory. Learn more

User attribute in the application	Maps to this string value or user attribute in IAM Identity Center	Format	
Subject	${user:preferredUsername}	persistent ▼	
http://schemas.xmlsoap.org/ws/2005/05/id	${user:name}	uri ▼	Remove
http://schemas.xmlsoap.org/claims/Group	Authenticated Users	uri ▼	Remove
email	${user:email}	basic ▼	Remove
group	${user:AD_GUID}	basic ▼	Remove

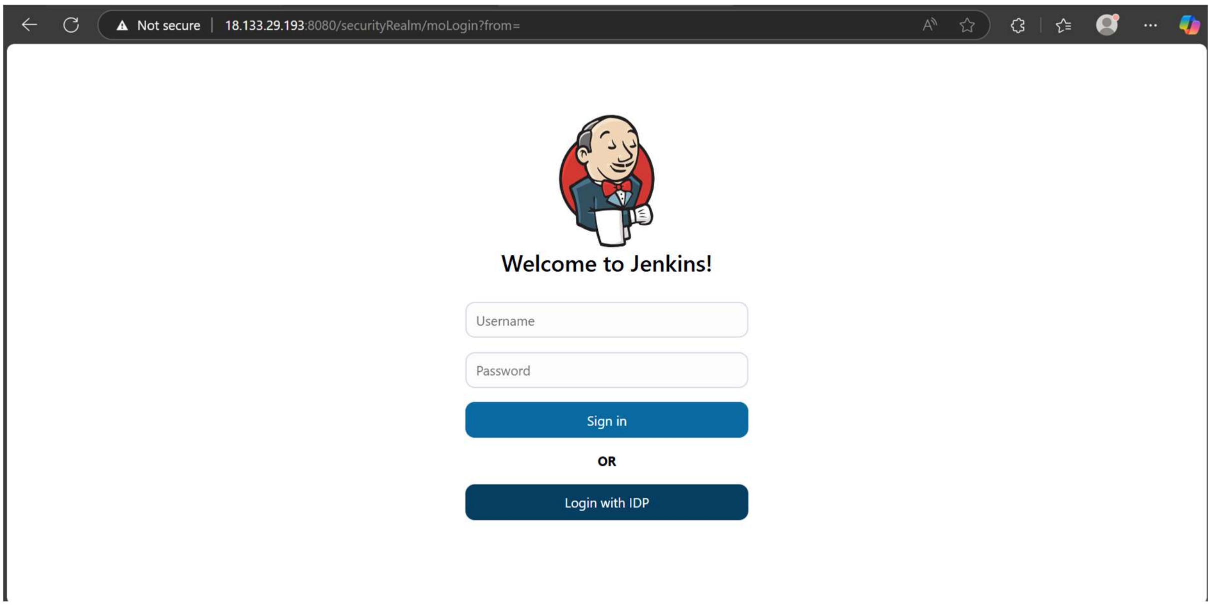
// Now save changes and re-login. It should reflect on login page.



AWS access portal

Applications (2)

// Ideal login page, after successful implementation of SSO.



Important points

* HTTPS is recommended for SSO.

* As for poc, HTTP is used.

* Assign the IAM Application created to user.