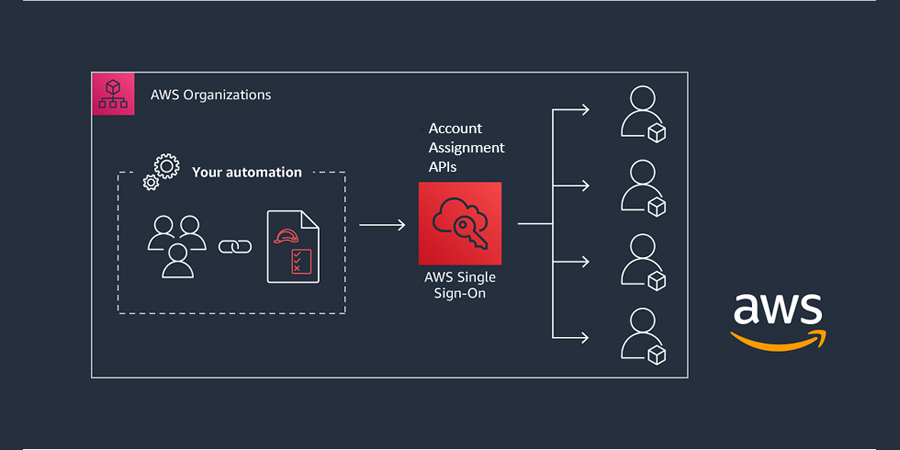
Creation of the sso with aws account

* Firstly we have to create a iam user and give them credentials as from the main account
* [**Identity and Access Management (IAM)**](https://us-east-1.console.aws.amazon.com/iam/home#/home)

Create an user >> Add user >> user name and give the credentials( **Access key - Programmatic access ; Password - AWS Management Console access)** >> Add the policies to the user >> Download the credentials file .

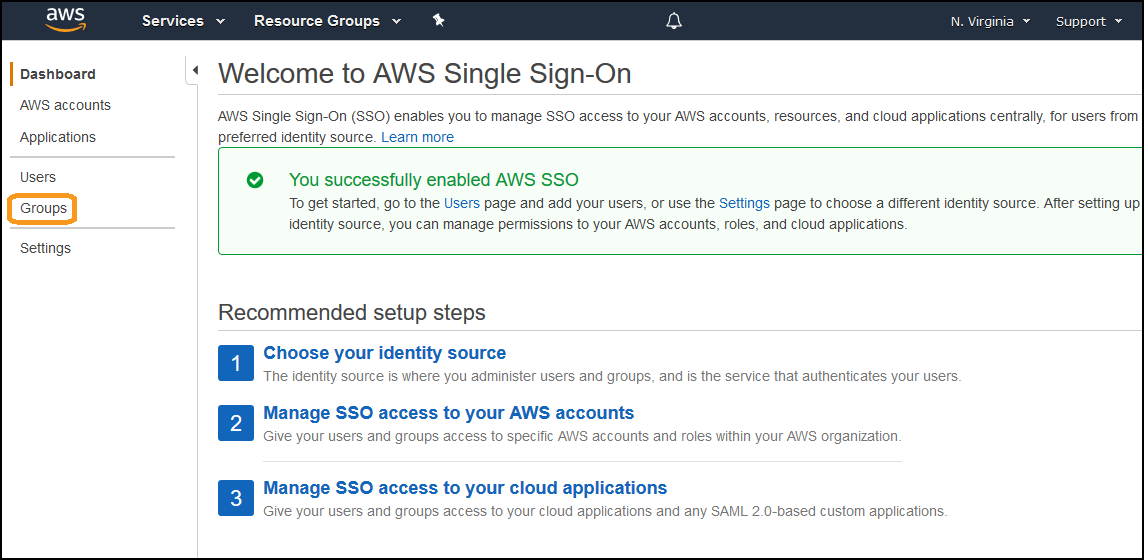
* After the creation of the Iam we can connect with those credentials and work with the services
* SSO :

AWS Single Sign-On (AWS SSO) is where you create, or connect, your workforce identities in AWS once and manage access centrally across your AWS organization. You can choose to manage access just to your AWS accounts or cloud applications



* Login to the AWS console as an IAM user with the required permissions, start typing **SSO** into the **Find Services** box and click on **AWS Single Sign-On**:

We can see the console as below



* Create a group first and create also create the users >> users can be given details are below

Enter the following details for user

* **Username**
* **Password**
* **Email address**
* **First name**
* **Last name**
* **Display name**
* After filling the user will receive an email, with a link to **Accept invitation**, the **Portal URL** and their **Username**
* When the user goes to the portal, they will enter in a **Password** and click **Update user**
* After that click on  **Permission sets**, and click **Create permission set**
* **In the permission set there are predefined and customized permissions**
* **After creation of permissions select the aws account in aws organization.**
* Assign the user to the account and give separate for each user or give all same permissions to the user .
* After that the user has access to the aws console each user has some set off permissions.

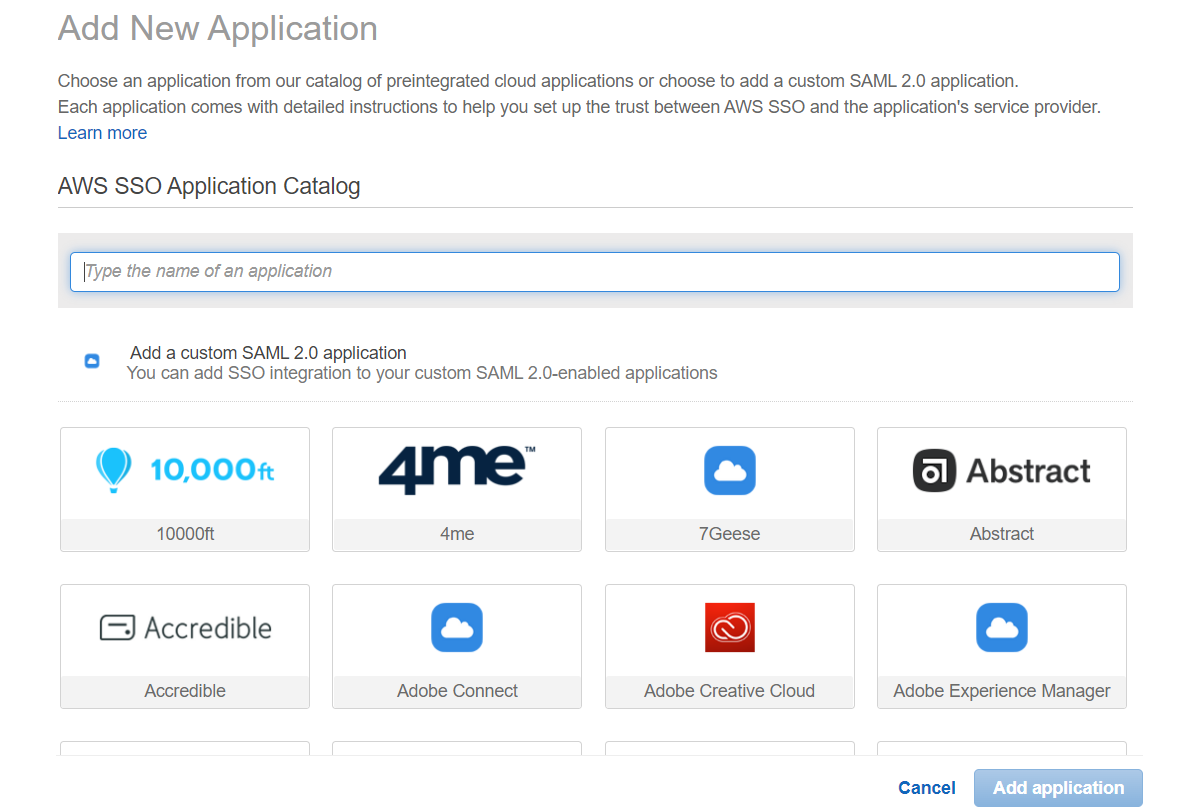
By using and modifying the data in the below doc

<https://www.wellarchitectedlabs.com/cost/100_labs/100_1_aws_account_setup/4_configure_sso/>

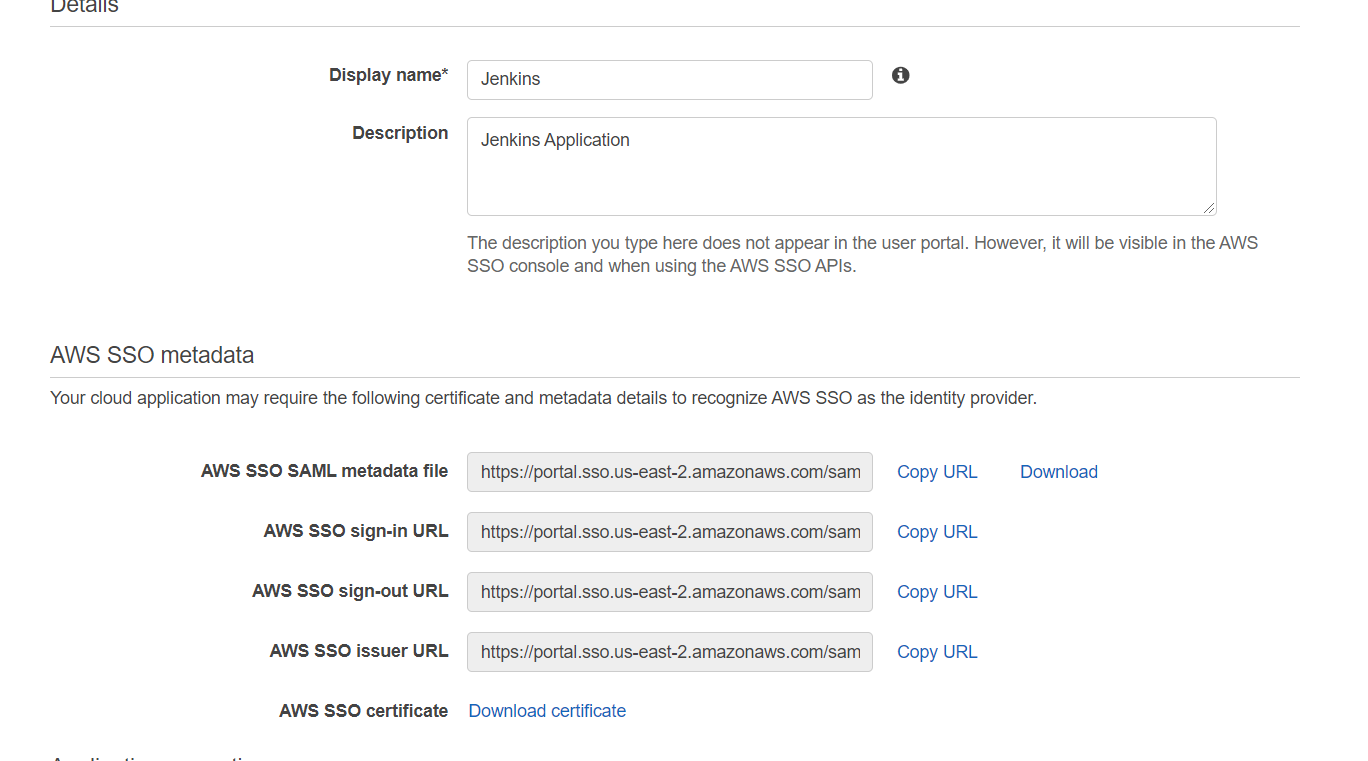
**Creation of the application in the sso**

* Jenkins application

Creation of the application of jenkins in the sso in a specific display



We can add the jenkins application and we connect to the instance where the jenkins is installed



1. Login to Jenkins with your administrator account.
2. On the Jenkins home page, choose **Manage Jenkins**. Then choose **Manage Plugins**.
3. Select the **Available** tab, then search for **SAML**.
4. Check the box next to **SAML**, then choose **Install without restart**.
5. Confirm SAML has a status of Success, then click on **Go back to dashboard.**
6. Click on **Manage Jenkins**, click on **Configure Global Security**.
7. Under **Security Realm**, choose **SAML 2.0**.
8. Download the **AWS SSO metadata** file and paste its content in **IdP Metadata**. **IdP Metadata** ( present in the below url it must be changes from user to user )
9. Next to **Email Attribute** enter **email**.
10. Next to **Logout URL** : ( enter the logout url)
11. Under **Authorization**, choose appropriate option. In this example, we will give choose **Project-based Matrix Authorization Strategy**, and check **Read** access overall for **Authenticated Users** group.
12. Next, click on **Apply** and then **Save**.
13. In your browser, paste **http://JENKINSURL:8080/securityRealm/metadata** in the address bar. Replace **JENKINSURL** with your server URL, hit Enter, and then save the metadata file on your computer.
14. Go back to the AWS SSO Console.
15. Configure **Application Metadata** by uploading file downloaded in step

Application ACS URL : **http://JENKINSURL/jenkins/securityRealm/finishLogin**

Application SAML audience : **http://JENKINSURL/jenkins/securityRealm/finishLogin**

1. Go to the **Attribute mappings** tab, here you need to update the Group attribute with the **Authenticated Users** value
2. Click **Save changes**. And assign the users to the application  in AWS SSO.

Verification sso from aws :

1. Access the AWS SSO end user portal using the credentials of a user assigned to the **Jenkins** application.
2. In the list of applications, choose **Jenkins** to initiate a login to Jenkins.
3. If login was successful you will be signed-in to the Jenkins application.

<http://JENKINSURL/jenkins/>

**In this jenkinsurl means : public ip with port number or create a load balancer attach the instance to it**

**We can encounter the error in the creation of the application**

**On daily we had start and stop the application the ip will changes so created the LOAD BALANCER for the application .**

**Create a application load balancer attach vpc ,subnet and target group and take the dns name and copy and paste in the JENKINSURL section .**

**By taking the help of the sample doc created the application but the aws give the doc for each application creation.**

<https://plugins.miniorange.com/saml-single-sign-on-sso-jenkins>

**By taking the process had created load balancer**

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/create-application-load-balancer.html>

<https://www.sumologic.com/blog/aws-application-load-balancer/>**.**

Creating users in Jenkins

===========================

1 Open the dashboard of jenkins

2 click on manage jenkins

3 click on manage users

4 click on create users

5 enter user credentials

Creating roles and assigning

==============================

1 Open the dashboard of jenkins

2 click on manage jenkins

3 click on manage plugins

4 click on role based authorization strategy plugin

5 install it

6 go to dashboard-->manage jenkins

7 click on configure global security

8 check enable security checkbox

9 go to authorization section-->click on role based strategy radio button

10 apply-->save

11 go to dashboard of jenkins

12 click on manage jenkins

13 click on manage and assign roles

14 click on mange roles

15 go to global roles and create a role "employee"

16 for this employee in overall give read access

and in view section give all access

I had assumed the user as Deveolper and Tester

17 go to project roles-->Give the role as developer

and patter as Dev.\* (ie developer role can access

only those jobs whose name start with Dev)

18 similarly create another role as tester and assign the pattern as "Test.\*"

19 give all permissions to developrs and tester

20 apply--save

21 click on assign roles

22 go to global roles and add user1 and user2

23 check user1 and user2 as employees

24 go to item roles

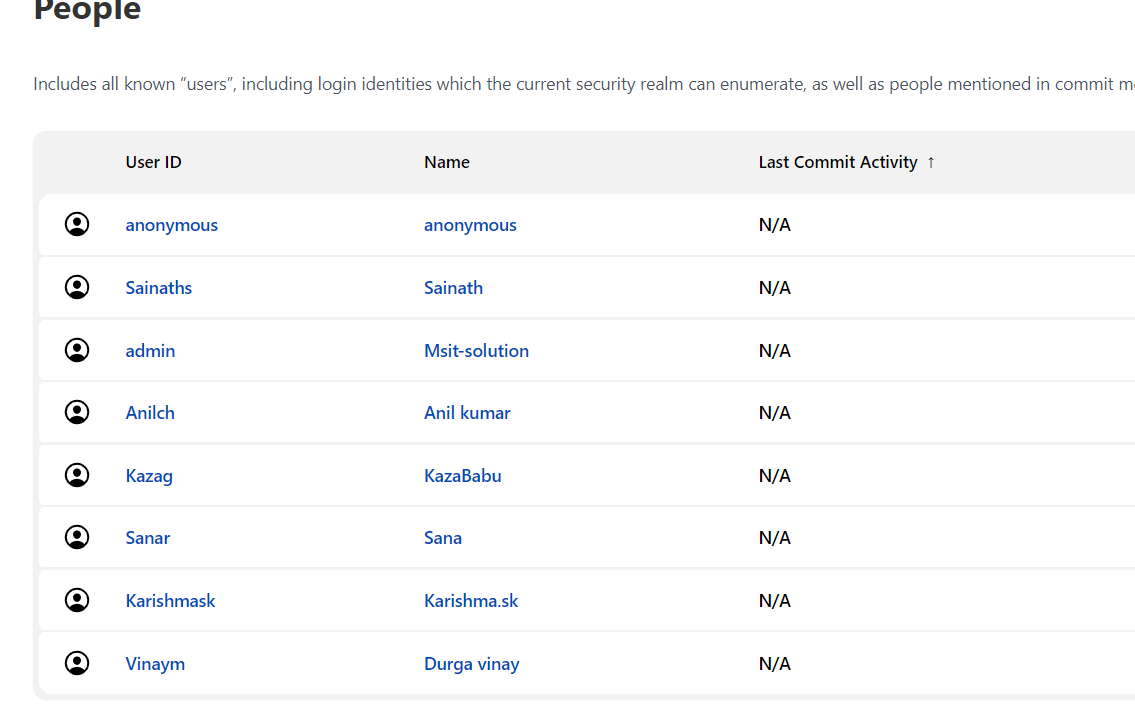
25 add user1 and user2

26 check user1 as developer and user2 as tester

27 apply-->save

If we login into jenkins as user1 we can access only the development

related jobs and user2 can access only the testing related jobs

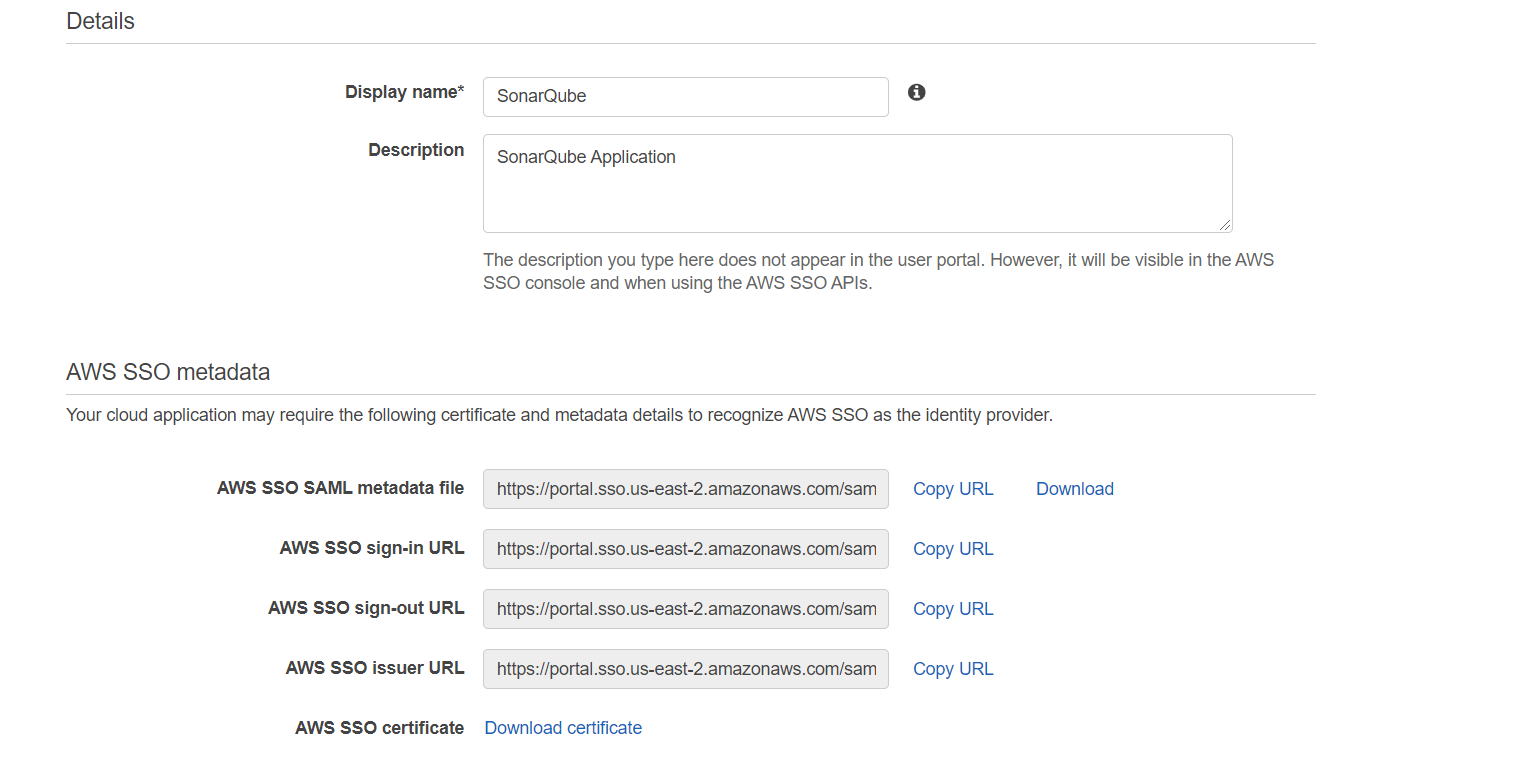


By taking the help of the doc created the user and give access to them

https://www.guru99.com/create-users-manage-permissions.html

* SonarQube Application
* SonarQube is a Code Quality Assurance tool that collects and analyzes source code, and provides reports for the code quality of your project. It combines static and dynamic analysis tools and enables quality to be measured continually over time.

On the Configure page in the AWS SSO Console, in the **Details** section, fill in the **Display name**, and the **Description**(optional) of the application



1. Login to your SonarQube account as an administrator.
2. On the home page, choose **Administration**, then choose **configuration click on general settings go to security** .
3. Enter the  **Identity Provider** and then choose **SAML enter saml login url and provider id .**
4. Download the AWS SSO certificate and paste it's content in **Provider certificate**. **Remove the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- from the certificate content copied .**
5. **Enter the values**

SAML user login attribute : login

SAML user name attribute : admin (or) user.name

SAML user email attribute : email

6.Scroll to the top of the SAML configuration page, and enable the toggle next to **Enabled**.

7.Go back to the AWS SSO console page where you are configuring the Application.

8. Under **Application properties**,

Application start url : <https://SONARQUBEURL/sessions/init/saml?return_to=%2F>

9. Under **Application metadata**, choose **If you don't have a metadata file, you can manually type your metadata values.** to display the application metadata settings

Application ACS URL : <https://SONARQUBEURL/oauth2/callback/saml>

Application SAML audience : Sonarqube

10.Click **Save changes**. And assign the users to the application  in AWS SSO.

## **Verifying SSO from AWS SSO**

1. Access the AWS SSO end user portal using the credentials of a user assigned to the **SonarQube** application.
2. In the list of applications, choose **SonarQube** to initiate a login to SonarQube.
3. If login was successful you will be signed-in to the SonarQube application.

<http://SONARQUBEURL//>

**In this SonarQube URL means : public Ip with port number or create a load balancer attach the instance to it .**

**By taking the help of the sample doc created the application but the aws give the doc for each application creation.**

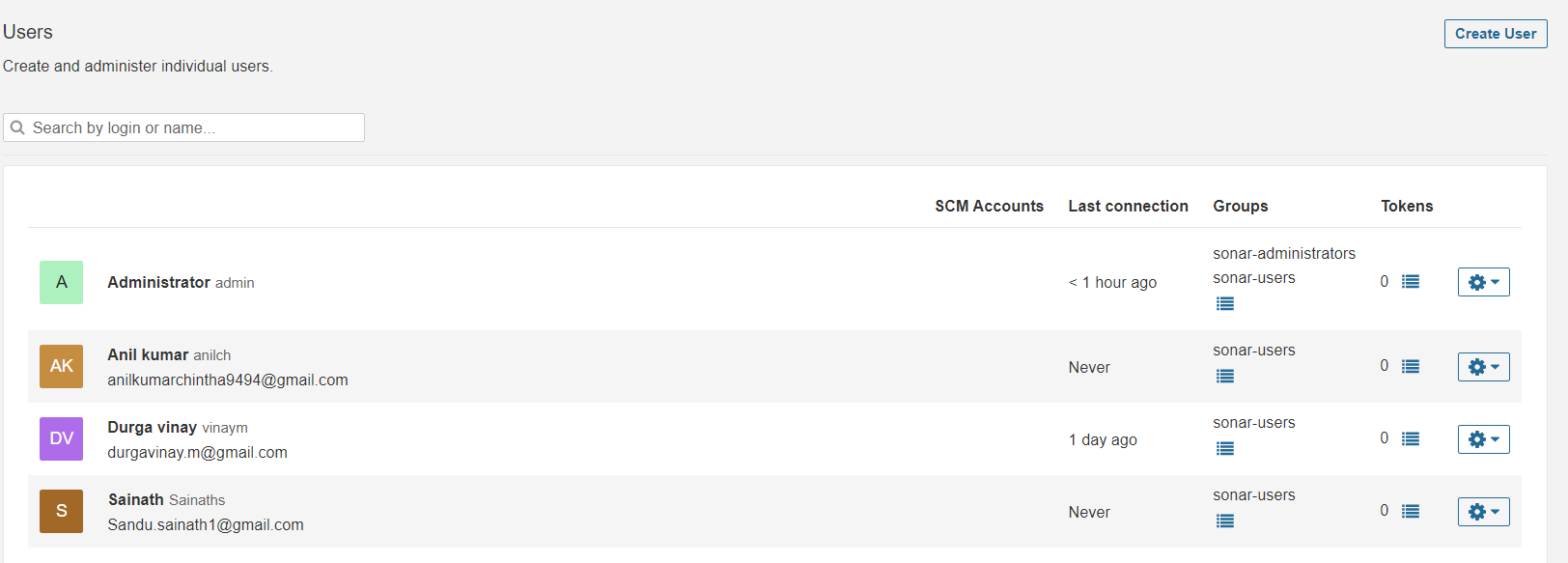
<https://www.miniorange.com/sonarqube-single-sign-on-(sso)#:~:text=SonarQube%20Single%20Sign%20On%20(SSO,user%20to%20the%20SonarQube%20application>.

Creation of Users

1.In Administration section go to security

2.Create the users and the user name email user name password

3.Go to global permissions and give the user the Quality profiles ,Quality gates ,Execut analaysis and create of project access

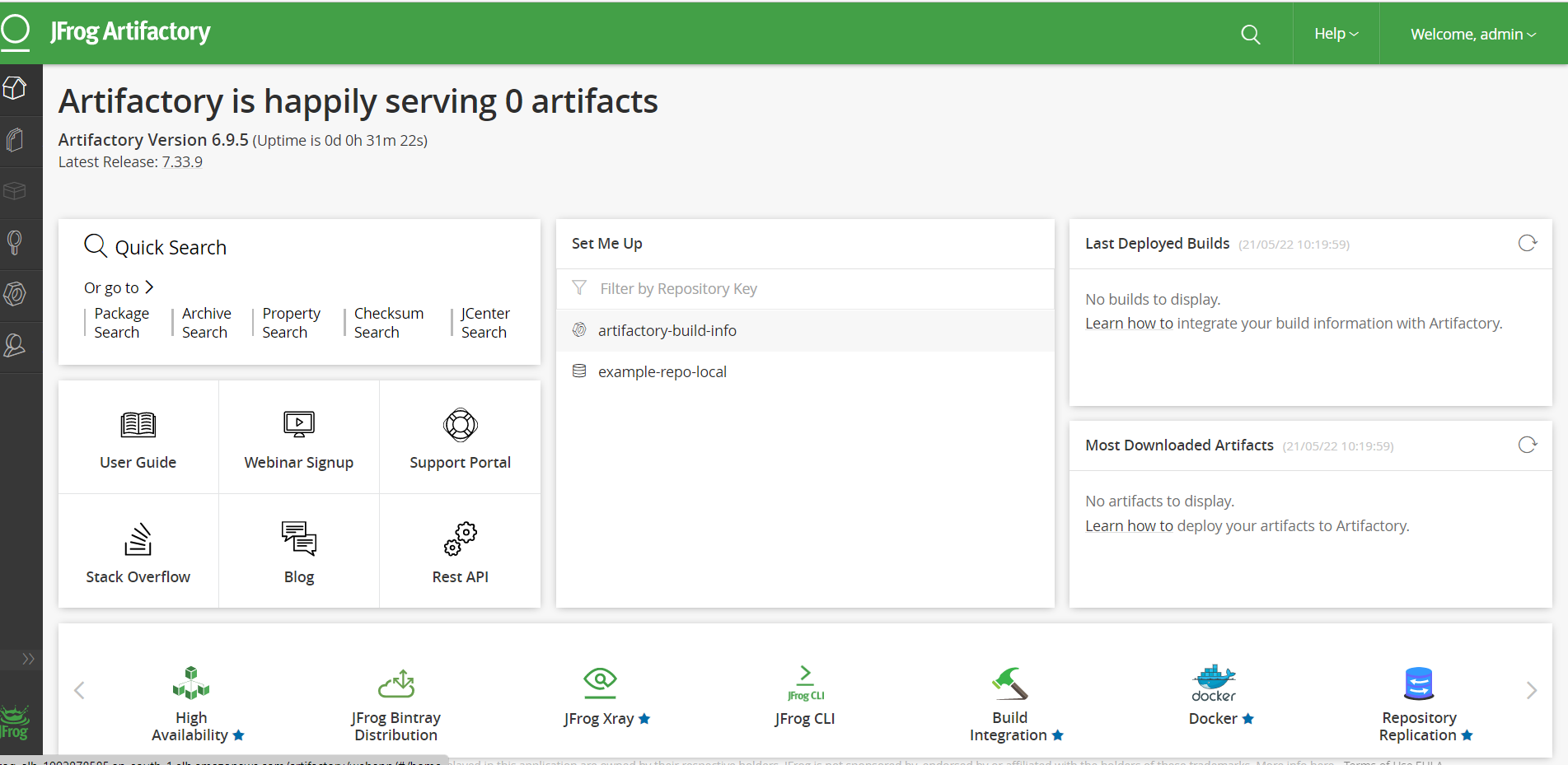


Creation of the users by using doc

<https://community.sonarsource.com/t/sonarqube-free-edition-access-to-multiple-users-or-admin-account/45308/4>

JFrog Artifactory :

JFrog Artifactory is a universal DevOps solution providing end-to-end automation and management of binaries and artifacts through the application delivery process that improves productivity across your development ecosystem



Installation of the infra service and installation can be made by using bash

We can add the some application for it but the Grafana has no sso linkage in aws-sso

**Monitoring Tools**

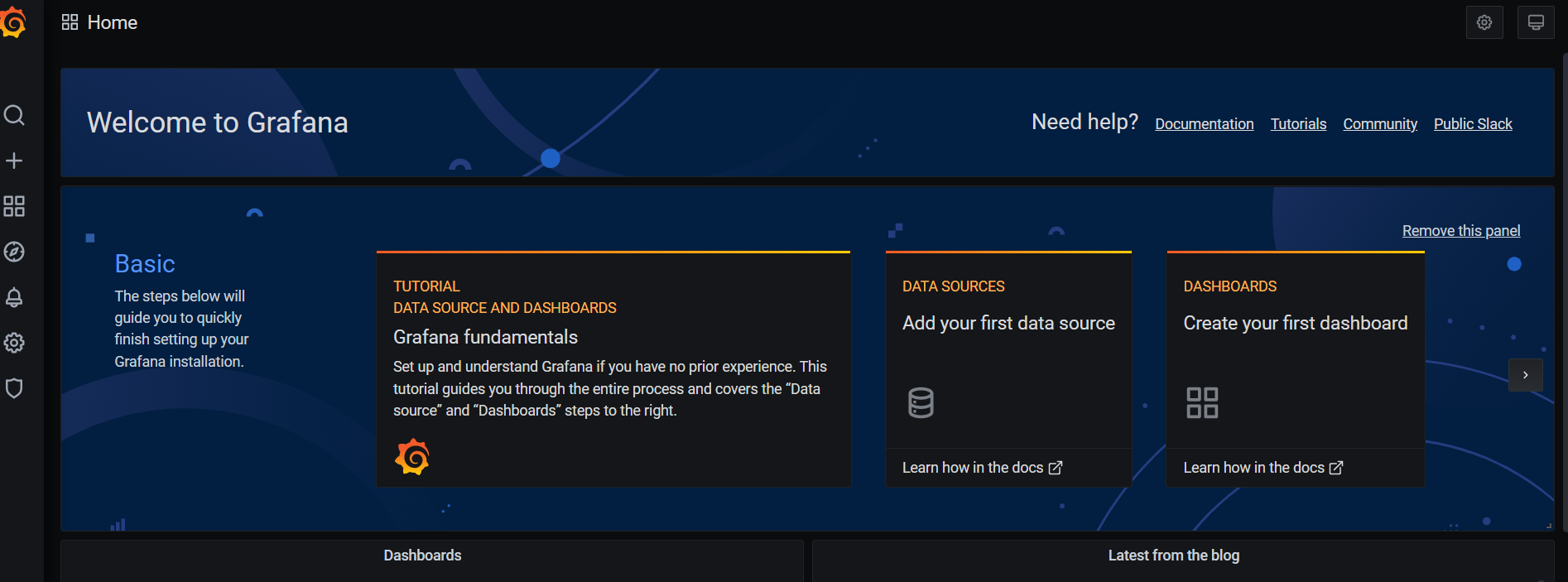
Grafana

[[](https://www.google.com/search?rlz=1C1FKPE_enIN969IN969&sxsrf=ALiCzsYowJKMTf6oOfG83Tya7tIwCOSQMw:1653709210132&q=What+is+Grafana+used+for?&tbm=isch&source=iu&ictx=1&vet=1&fir=l12hUZIIlVAwVM%252Cz82aWkRxXapNTM%252C_&usg=AI4_-kQghnWnbmOTqP6Y2huqSjszvGIQkQ&sa=X&ved=2ahUKEwi89p_wooH4AhXIR2wGHbdJCFQQ9QF6BAgZEAE#imgrc=l12hUZIIlVAwVM)](https://www.google.com/search?rlz=1C1FKPE_enIN969IN969&sxsrf=ALiCzsYowJKMTf6oOfG83Tya7tIwCOSQMw:1653709210132&q=What+is+Grafana+used+for?&tbm=isch&source=iu&ictx=1&vet=1&fir=l12hUZIIlVAwVM%252Cz82aWkRxXapNTM%252C_&usg=AI4_-kQghnWnbmOTqP6Y2huqSjszvGIQkQ&sa=X&ved=2ahUKEwi89p_wooH4AhXIR2wGHbdJCFQQ9QF6BAgZEAE" \l "imgrc=l12hUZIIlVAwVM)

Grafana is a multi-platform open source analytics and interactive visualization web application. It allows you to query, visualize, alert on, and understand your metrics no matter where they are stored.

Grafana has been installed by using the bash script and Infra has been created by using the terraform

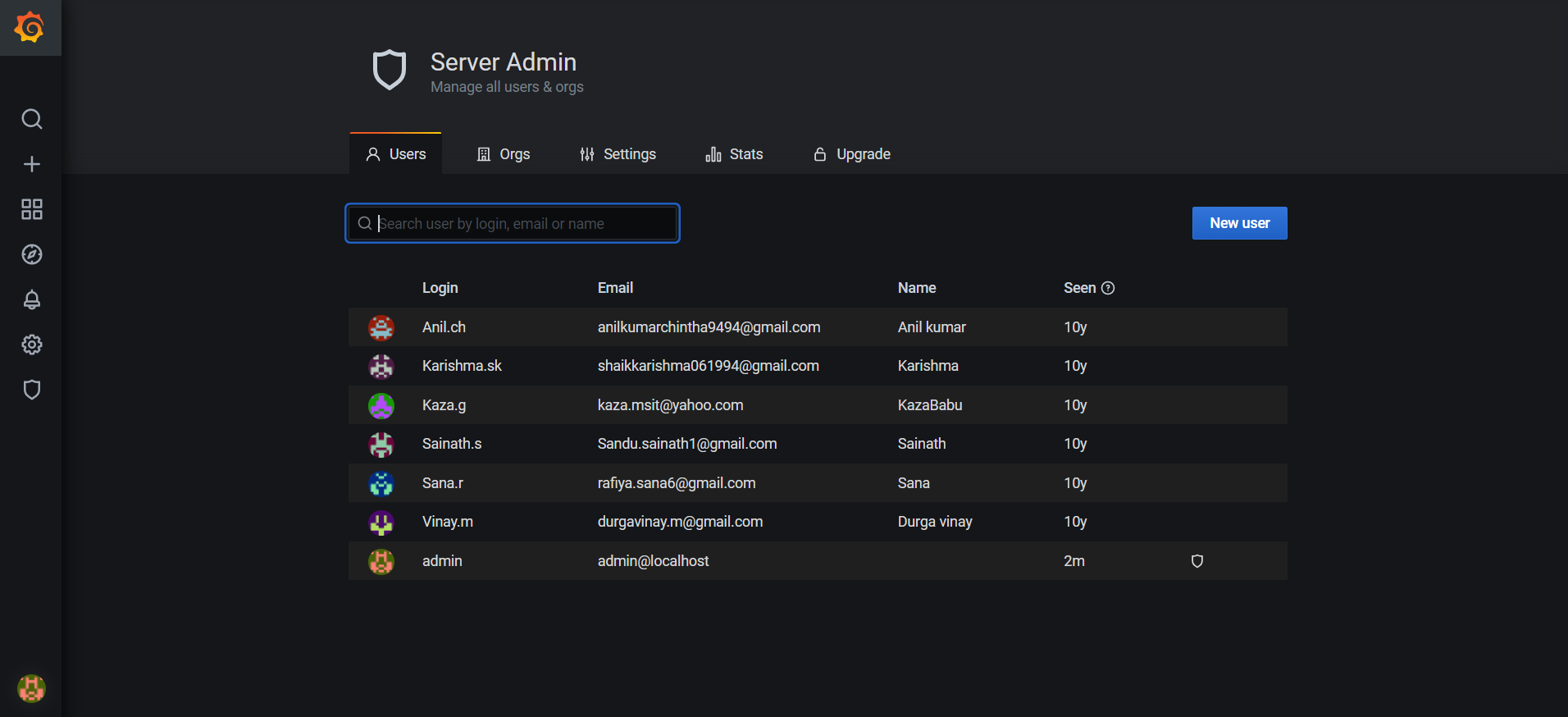
The user interface of the grafana has been like below image



In the Grafan I had created the some users and gave the permissions for them.

**CREATE USERS**

1. On the sidebar, click the **Server Admin** (shield) icon.
2. In the Users tab, click **New user**.
3. In **Name**, enter the name of the user.
4. In **E-mail**, enter the email of the user.
5. In **Username**, enter the username that the user will use to log in.
6. In **Password**, enter a password. The user can change their password once they log in.
7. Click **Create** to create the user account.



**ASSIGN THE USERS TO TEAM:**

Instead of granting permissions to individual users, teams let you grant permissions to a group of users.

Teams are useful when onboarding new colleagues. If you add a user to a team, they get access to all resources assigned to that team.

**Create a Team**

1. In the sidebar, hover your mouse over the **Configuration** (gear) icon and then click **Teams**.
2. Click **New team**.
3. In **Name**, enter the name of the team: either *Marketing* or *Engineering*. You do not need to enter an email.
4. Click **Create**.

**Add a User to the team**

1. Click **Add member**.
2. In the **Add team member** box, select the user you want to add to the team. Refer to the table above for team assignments.
3. Click **Add to teams**

