DevOps-Project-BankApp-CICD-BlueGreen-Monitoring-LogAggregartion Multi-Cloud (AWSandAzure)

By Ritesh Kumar Singh

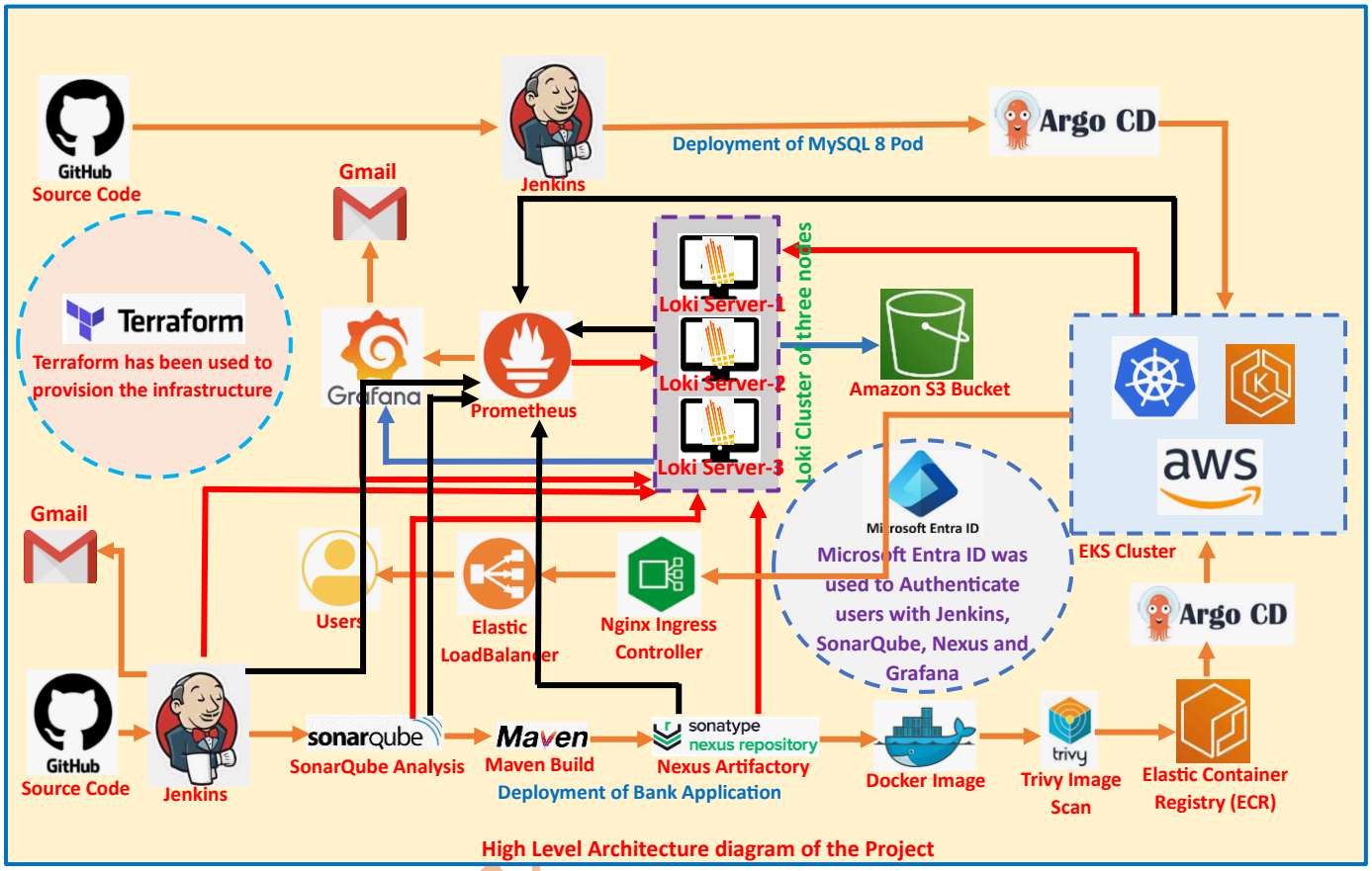
Email Address: - riteshkumarsingh9559@gmail.com

LinkedIn: - https://www.linkedin.com/in/ritesh-kumar-singh-41113128b/

GitHub: - https://github.com/singhritesh85

या कुन्देन्दुतुषारहारधवला या शुभ्रवस्त्रावृता
या वीणावरदण्डमण्डितकरा या श्वेतपद्मासना।
या ब्रह्माच्युत शंकरप्रभृतिभिर्देवैः सदा वन्दिता
सा मां पातु सरस्वती भगवती निःशेषजाड्यापहा ॥

DevOps-Project-BankApp-CICDBlueGreen-Monitoring-LogAggregartion

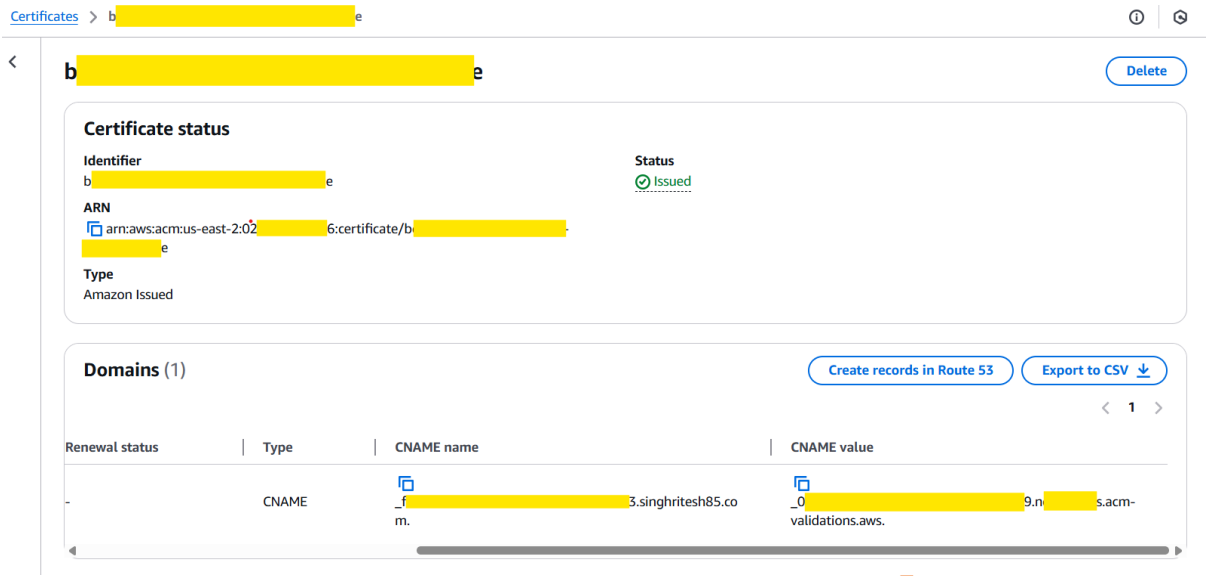


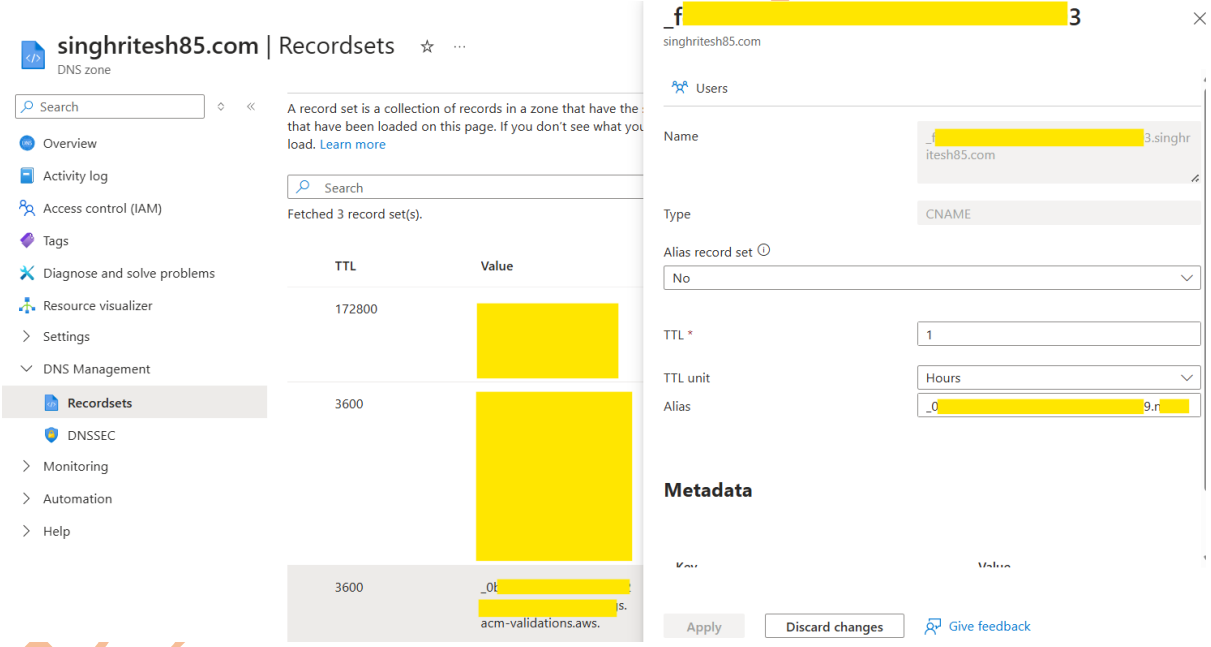
High Level Architecture diagram of the Project

This DevOps Project deals with creation of Infrastructure using Terraform and setup of CICD Pipeline using Jenkins, Monitoring using Prometheus and Grafana and Log Aggregation using Loki, Promtail and Grafana. SonarQube was used for Code-Analysis and Maven was used as the Build Tool. Nexus Artifactory was used to keep the Artifacts as shown in the Architecture diagram above. Trivy was used for Docker Image Scan. The Docker Image was kept in the Elastic Container Registry (ECR) and which was deployed to EKS Cluster using the ArgoCD as shown in the high-level architecture diagram above. User was able to access the Application through the Ingress and hence the Kubernetes Service. The source code was present in the GitHub Repository https://github.com/singhritesh85/Bank-App.git. For this project to deploy a new Release of code I had used Blue-Green deployment with the help of ArgoCD Rollout. Promtail and Node Exporter was installed on all the Loki Servers, Grafana, Prometheus, Jenkins (Master and Slave Nodes) using the boot-strapping script. For the EKS cluster the Promtail and Node Exporter was installed with the help of helm.

Promtail is Log Aggregation Agent for Loki and Node Exporter agent collects the metrics and forward to Loki and Prometheus respectively. The high-level architecture diagram of the project is as shown above. Installation of Jenkins-Master, Jenkins-Slave, Nexus, SonarQube, Loki Servers, Prometheus and Grafana had been done using the bootstrapping Script.

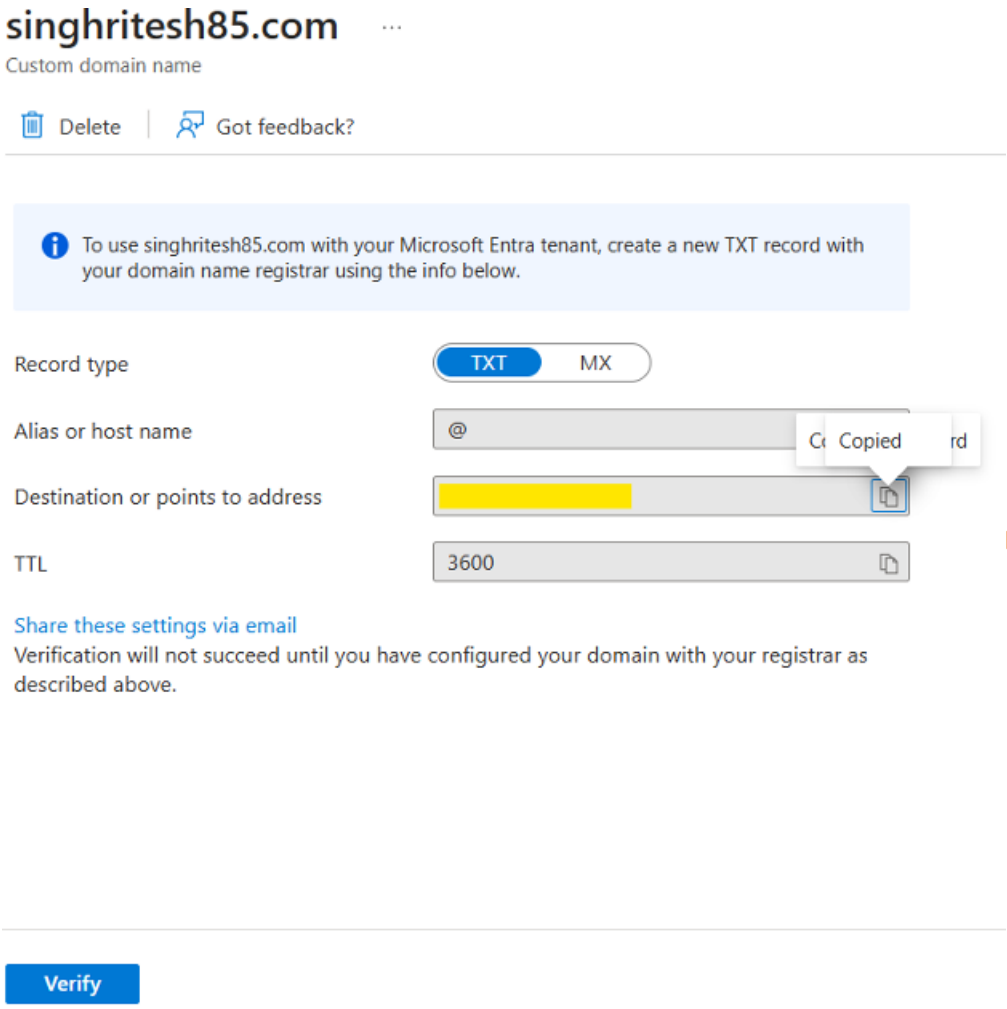
To validate the SSL Certificate generated from AWS certificate manager I used DNS validation and did the entry for Azure DNS Zone record set of type CNAME as shown in the screenshot attached below.



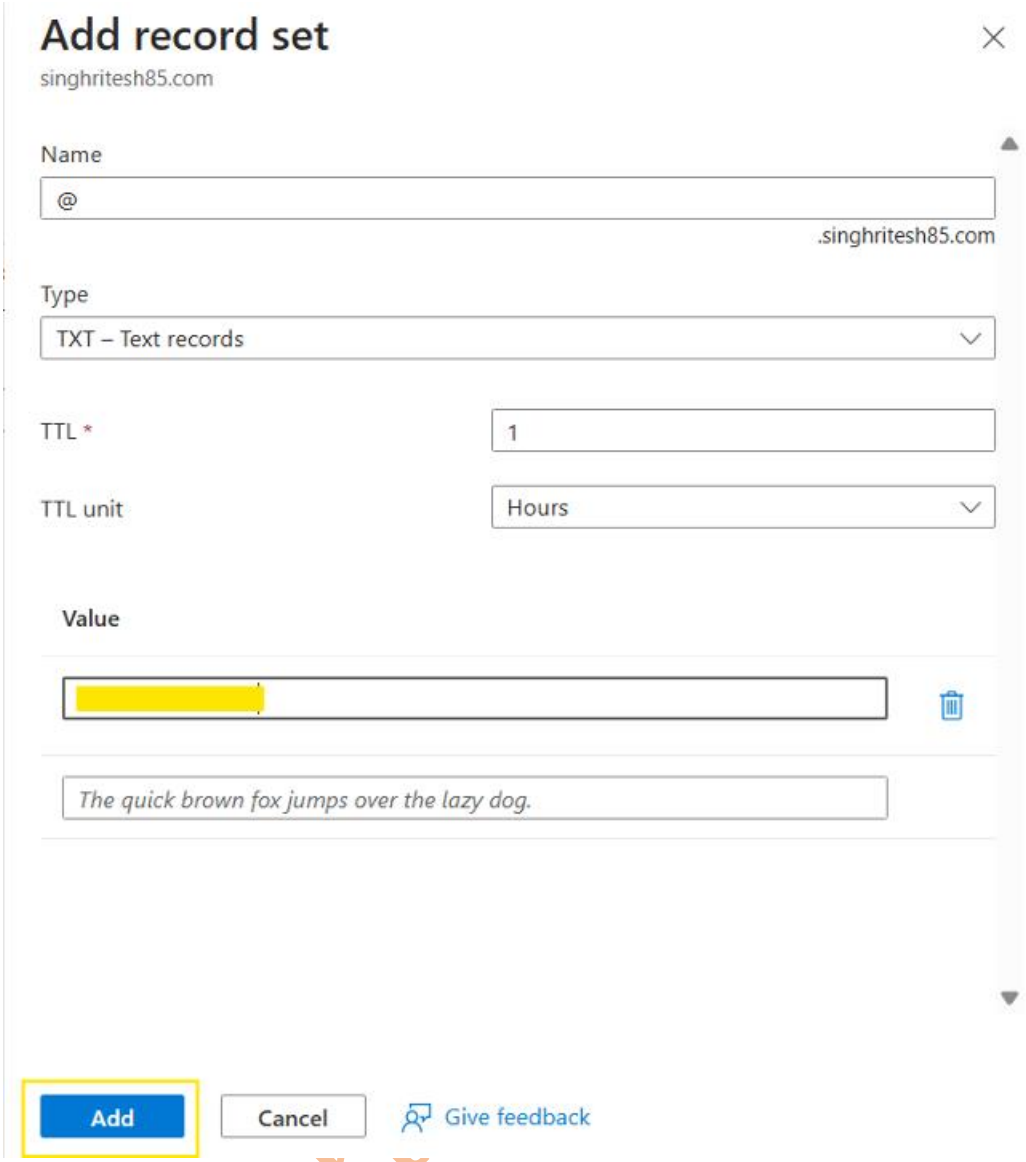


Then I wait for around 20 seconds and after that SSL Certificate was issued as shown in the first screenshot attached above.

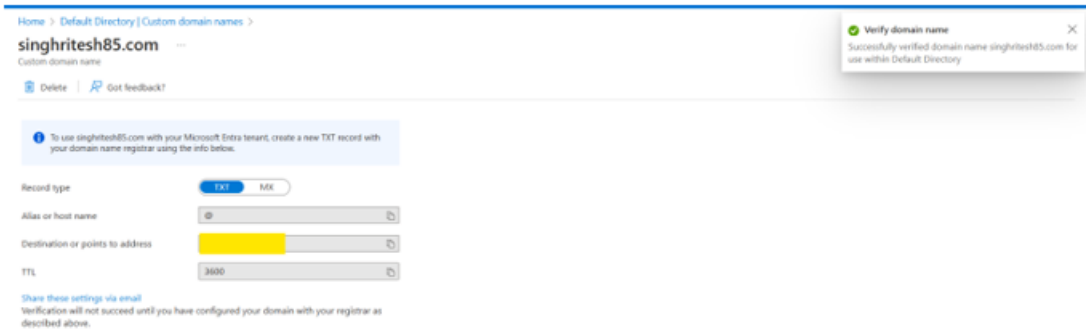
In my Azure Active Directory, I created a custom domain singhritesh85.com then made this custom domain as primary domain as shown in the screenshot attached below.



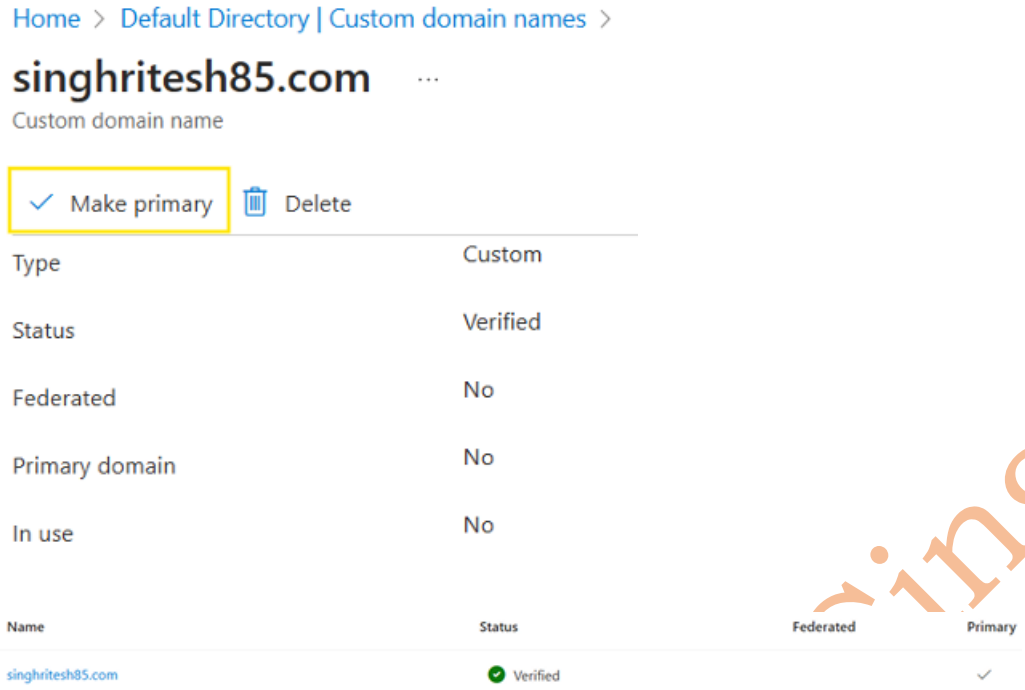
Do the entry in Azure DNS Zone to create the record set with Name and value and type TXT as shown in the screenshot attached below.



Then verified it and found it was verified successfully as shown in the screenshot attached below.



Then made this as a primary domain as shown in the screenshot attached below.



Verified from above attached screenshot, in my Azure Entra ID, the primary domain is **singhritesh85.com**.

For this project I am creating AWS Resources and the Azure Resources using the single terraform script present in the GitHub Repo https://github.com/singhritesh85/DevOps-Project-Bank-Application-Blue-Green-Deployment-Aws.git at the path **terraform-bankapp-multicloud**. In my case the terraform was installed on an EC2 Installed and a sufficient Role was attached (RBAC) to the EC2 and installed azure-cli and provided the access as shown in the screenshot attached below. The state file and sate lock in terraform was achieved using the Azure Storage Account's container.

```
[root@ main]# echo -e "[azure-cli]
> name=Azure CLI
> baseurl=https://packages.microsoft.com/yumrepos/azure-cli
> enabled=1
> gpgcheck=1
> gpgkey=https://packages.microsoft.com/keys/microsoft.asc" | sudo tee /etc/yum.repos.d/azure-cli.repo
[azure-cli]
name=Azure CLI
baseurl=https://packages.microsoft.com/yumrepos/azure-cli
enabled=1
gpgcheck=1
gpgkey=https://packages.microsoft.com/keys/microsoft.asc
[root@ main]# yum install azure-cli -y
```

```
[root@ main]# az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code  to authenticate
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "",
    "id": "",
    "isDefault": true,
    "managedByTenants": [],
    "name": "",
    "state": "Enabled",
    "tenantId": "",
    "user": {
      "name": "",
      "type": "user"
    }
  }
]
```

To create the Resources (AWS and Azure Resources) using the terraform script first switch the directory to **terraform-bankapp-multicloud/main** and then run the commands in below given orders.

terraform init ---------> **Initialize the working directory by installing plugins and required providers**

terraform validate ----------> **check the terraform script is correct or not**

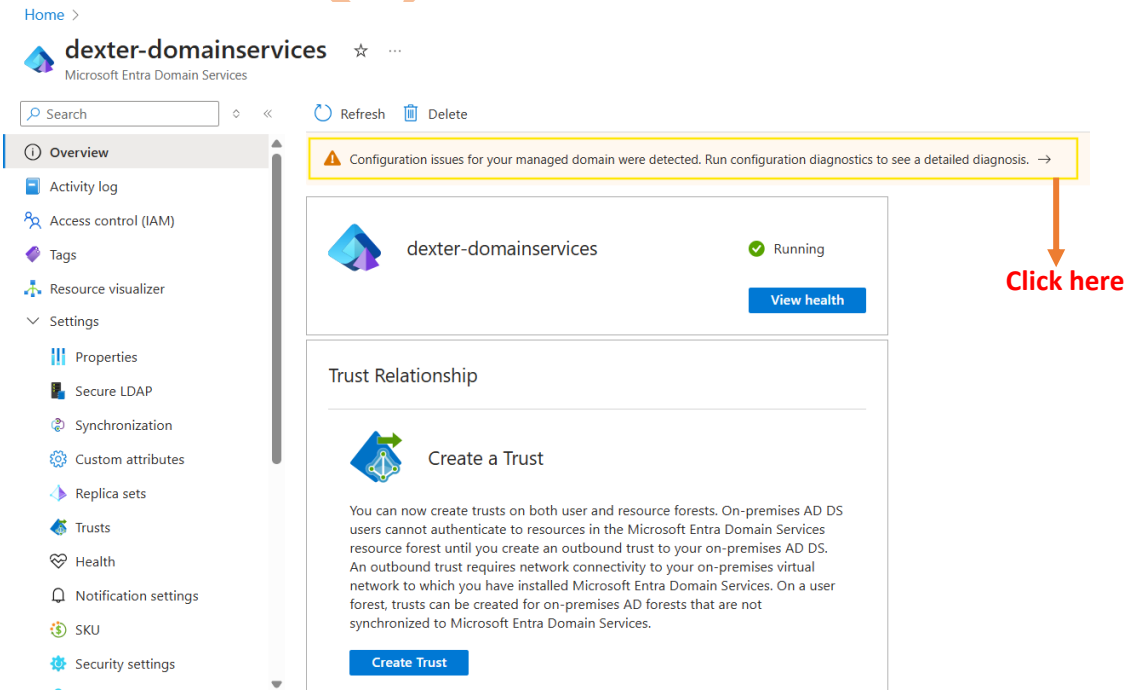
terraform plan ----------> **To verify what are the resources to be created**

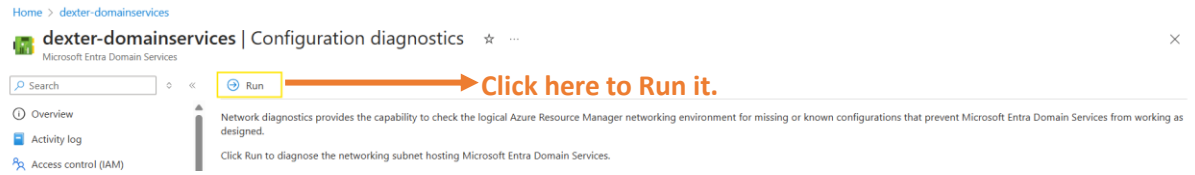
terraform apply -auto-approve ----------> **To create the resources**

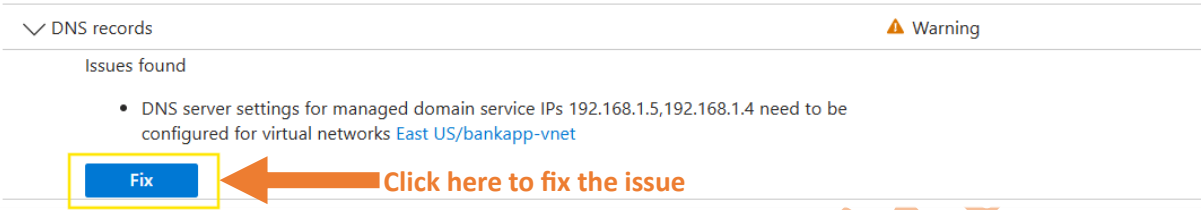
terraform destroy -auto-approve ----------> **Run this command only when you want to destroy all the resources.**

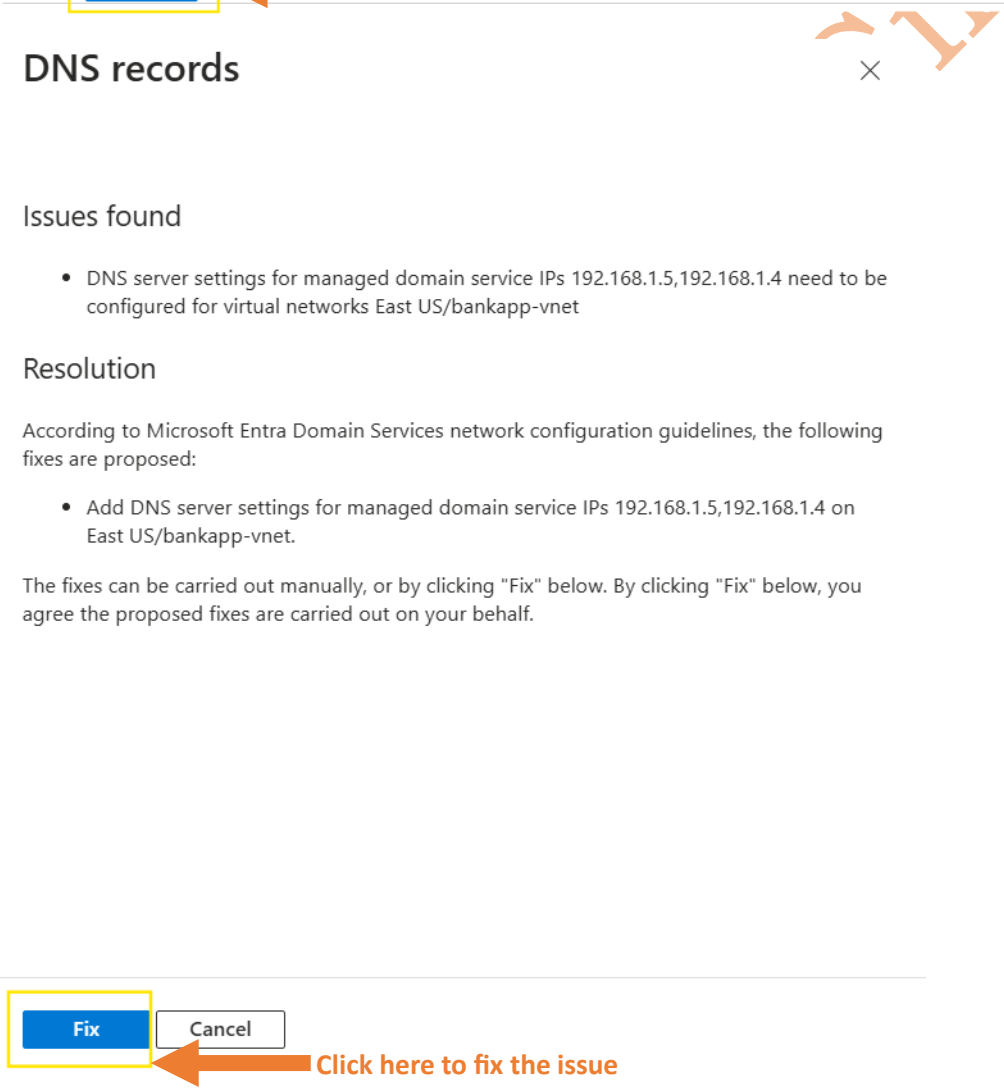
```
module.eks_cluster.azurerm_active_directory_domain_service.entra_ds: Still creating... [1h11m41s elapsed]
module.eks_cluster.azurerm_active_directory_domain_service.entra_ds: Still creating... [1h11m51s elapsed]
module.eks_cluster.azurerm_active_directory_domain_service.entra_ds: Still creating... [1h12m1s elapsed]
module.eks_cluster.azurerm_active_directory_domain_service.entra_ds: Creation complete after 1h12m4s [id=/subscriptions/[redacted]/r
esourceGroups/bankapp-rg/providers/Microsoft.AAD/domainServices/dexter-domainservices/initialReplicaSetId/[redacted]]
Releasing state lock. This may take a few moments...

Apply complete! Resources: 143 added, 0 changed, 0 destroyed.

Outputs:

acr_ec2_private_ip_alb_dns = {
  "EC2_Instance_Blaxbox_Exporter_Server_Private_IP_Address" = "[redacted]"
  "EC2_Instance_Grafana_Server_Private_IP_Address" = "[redacted]"
  "EC2_Instance_Jenkins_Master_Server_Private_IP_Address" = "[redacted]"
  "EC2_Instance_Jenkins_Slave_Server_Private_IP_Address" = "[redacted]"
  "EC2_Instance_Loki_Servers_Private_IP_Addresses" = [
    "[redacted]",
    "[redacted]",
    "[redacted]",
  ]
  "EC2_Instance_Prometheus_Server_Private_IP_Address" = "[redacted]"
  "EC2_Instance_SonarQube_Server_Private_IP_Address" = "[redacted]"
  "Grafana_ALB_DNS_Name" = "Grafana-[redacted].us-east-2.elb.amazonaws.com"
  "Jenkins_ALB_DNS_Name" = "jenkins-ms-[redacted].us-east-2.elb.amazonaws.com"
  "Loki_ALB_DNS_Name" = "Loki-[redacted].us-east-2.elb.amazonaws.com"
  "SonarQube_ALB_DNS_Name" = "SonarQube-[redacted].us-east-2.elb.amazonaws.com"
  "registry_id" = "[redacted]"
  "repository_url" = "02[redacted]6.dkr.ecr.us-east-2.amazonaws.com/bankapp"
}
```

This terraform script took total 1 hour 15 minutes in all to got executed successfully as shown in the screenshot attached above. After that All the resources in AWS and Azure will be created. The Microsoft Entra Domain Services gets created and I did below change in the Microsoft Entra Domain Services as shown in the screenshot attached below.

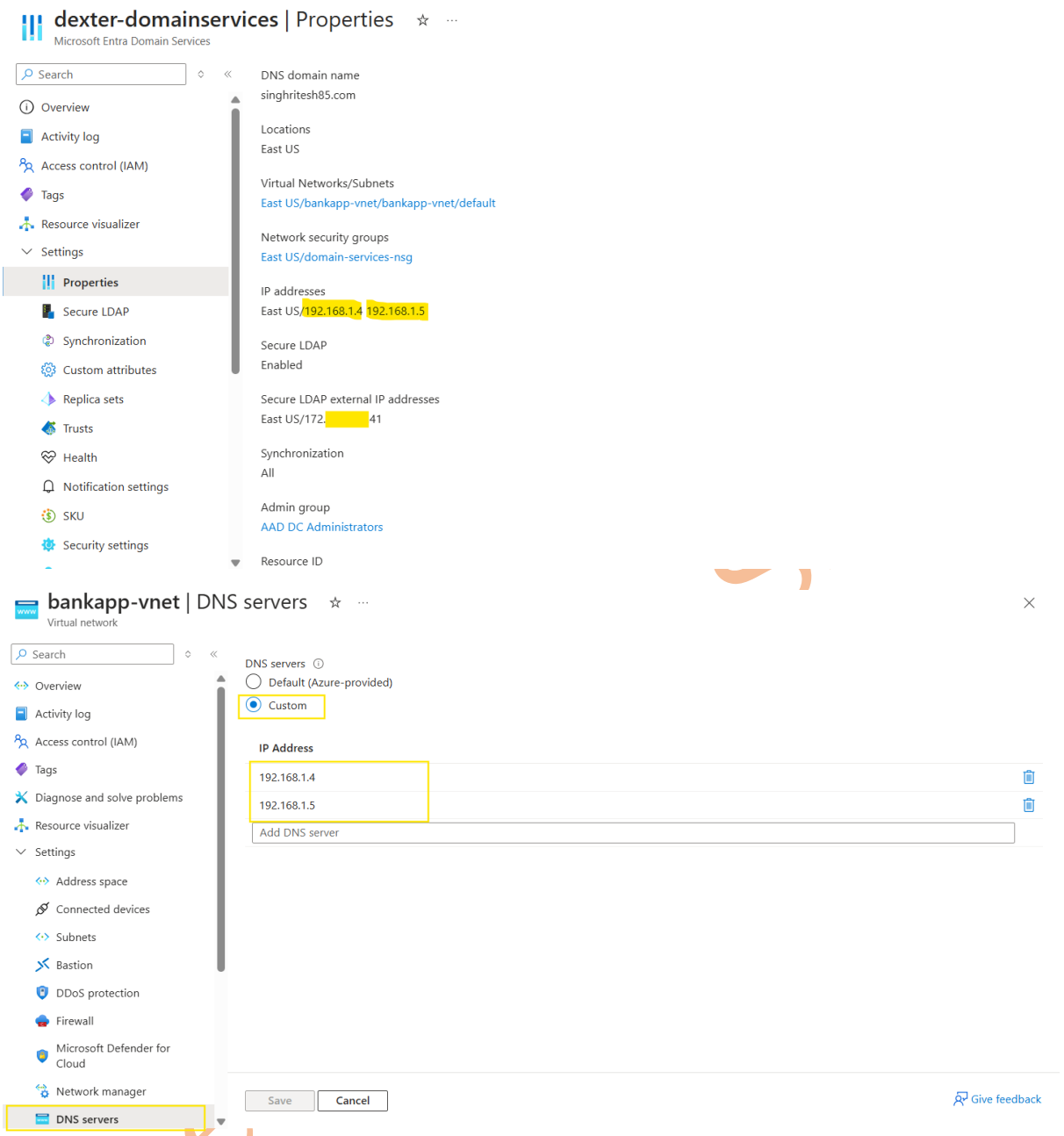




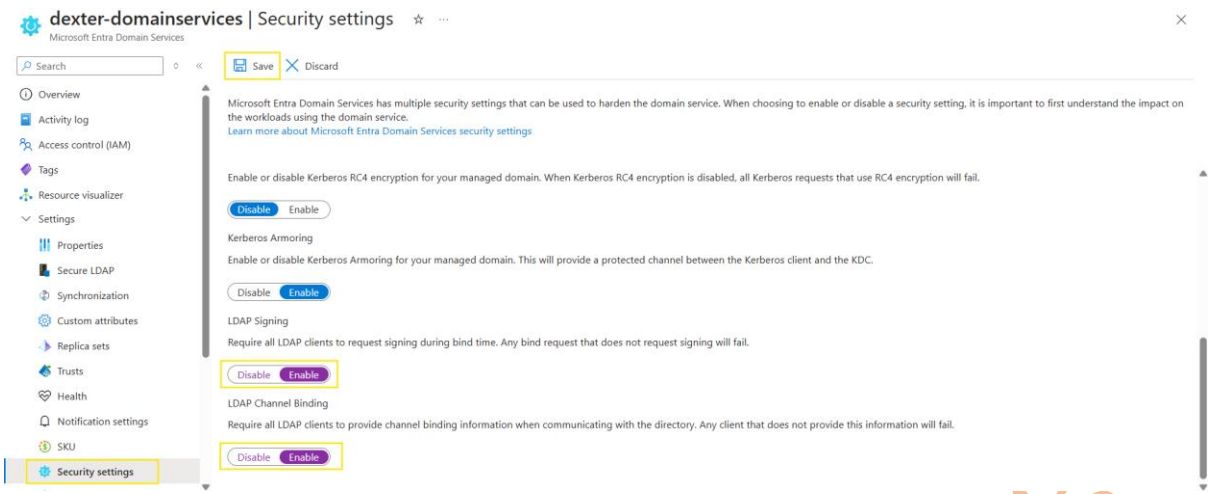


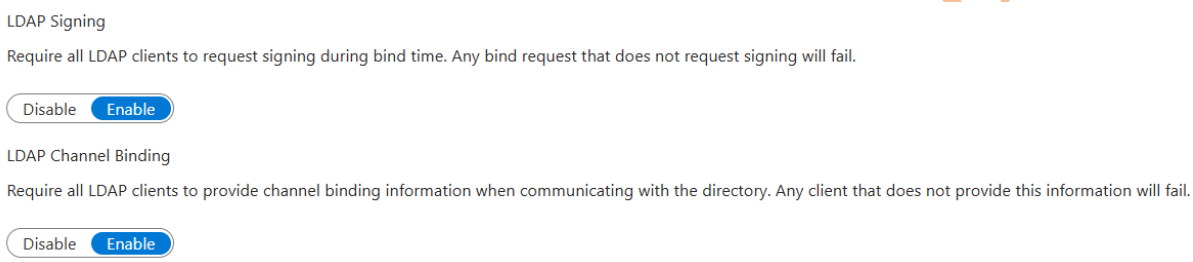


By fixing this issue through click this button will change the DNS Servers for VNet from Default (Azure-Provided) to Custom as shown in the screenshot attached below.

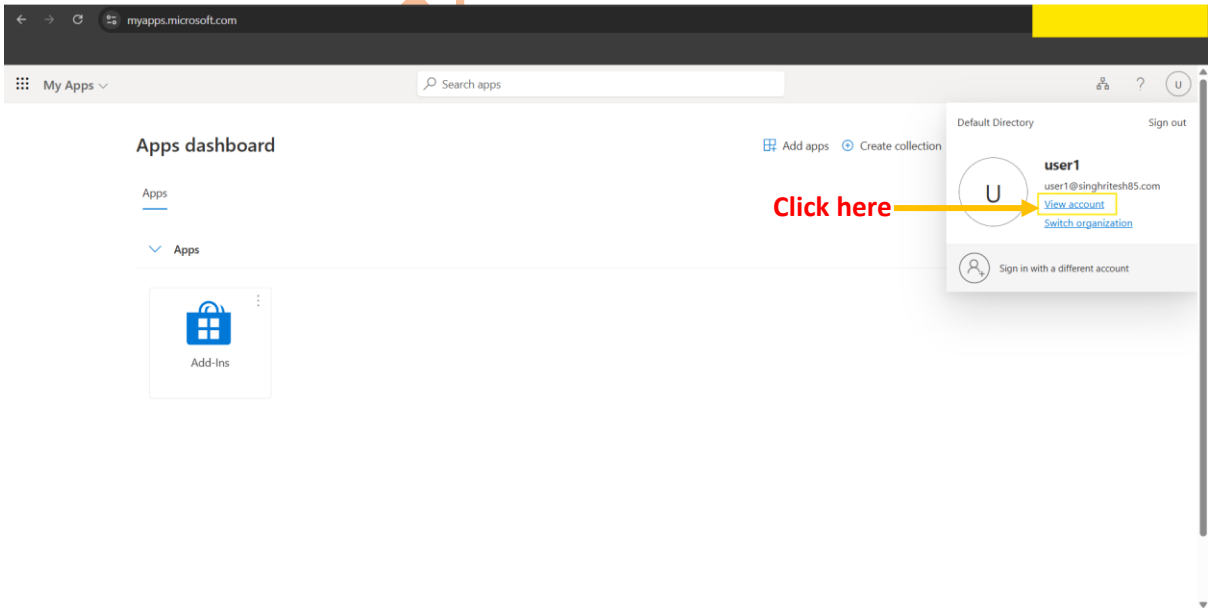


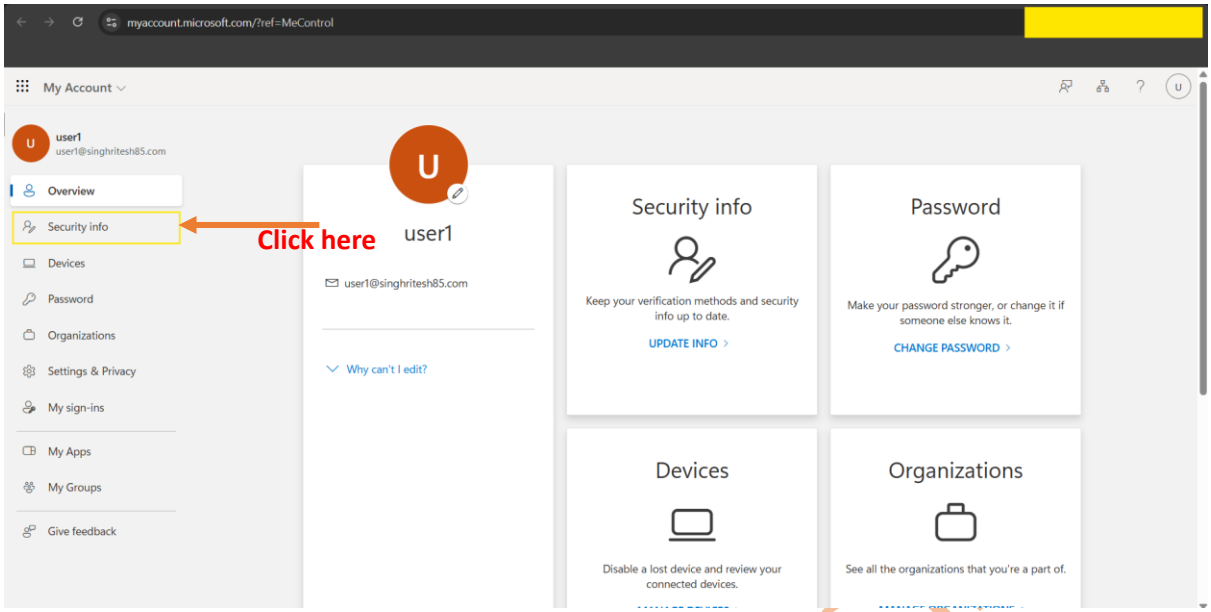
Then go to Security Setting of Microsoft Entra Domain Services and enable LDAP Signing and LDAP Channel Binding then save this setting as shown in the screenshot attached below.

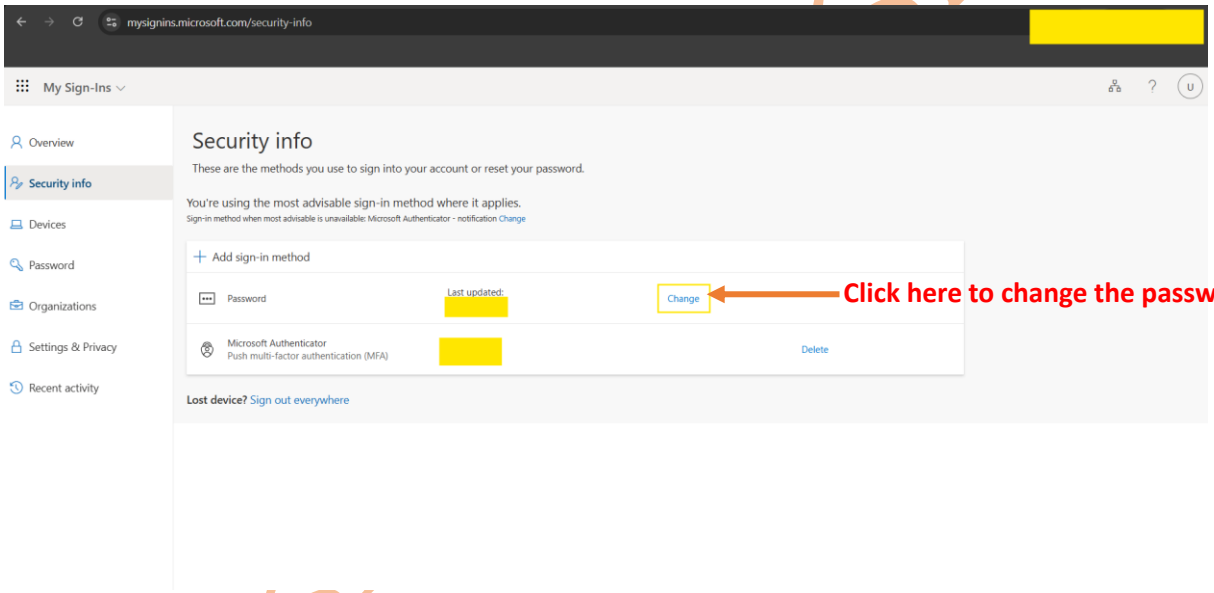


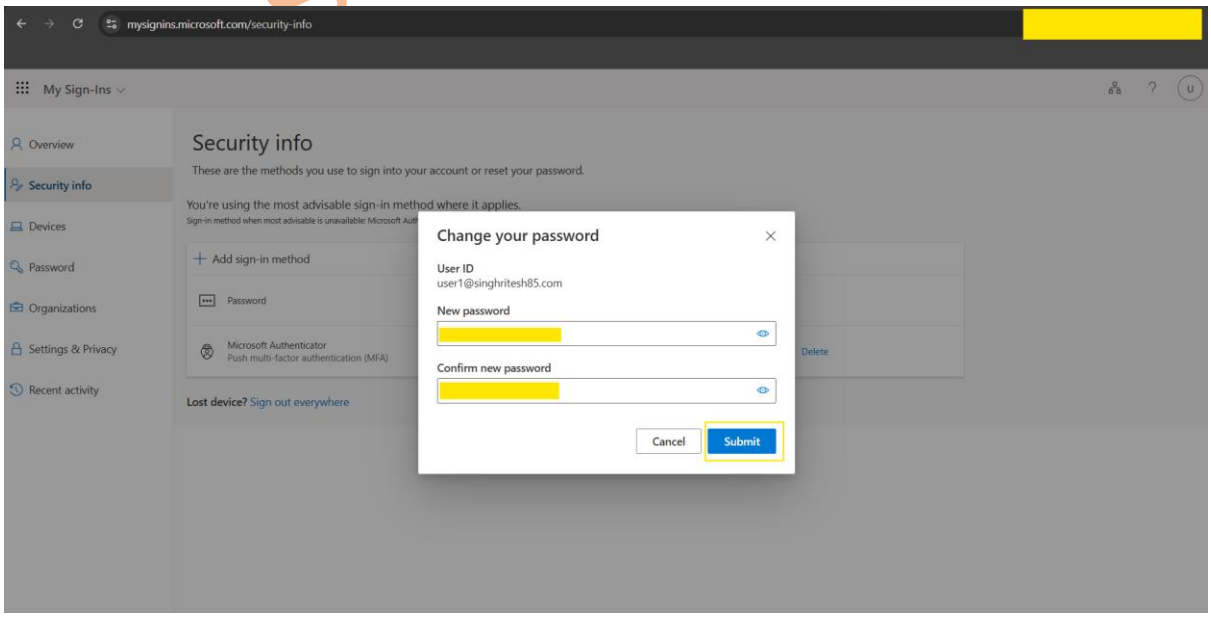


After successfully creating the Microsoft Entra Domain Services it is mandatory to change the password of user **user1** which was the member of Group **AAD DC Administrators**. To change the password of user **user1** I logged-in into the URL https://myapps.microsoft.com as shown in the screenshot attached below and changed the password.

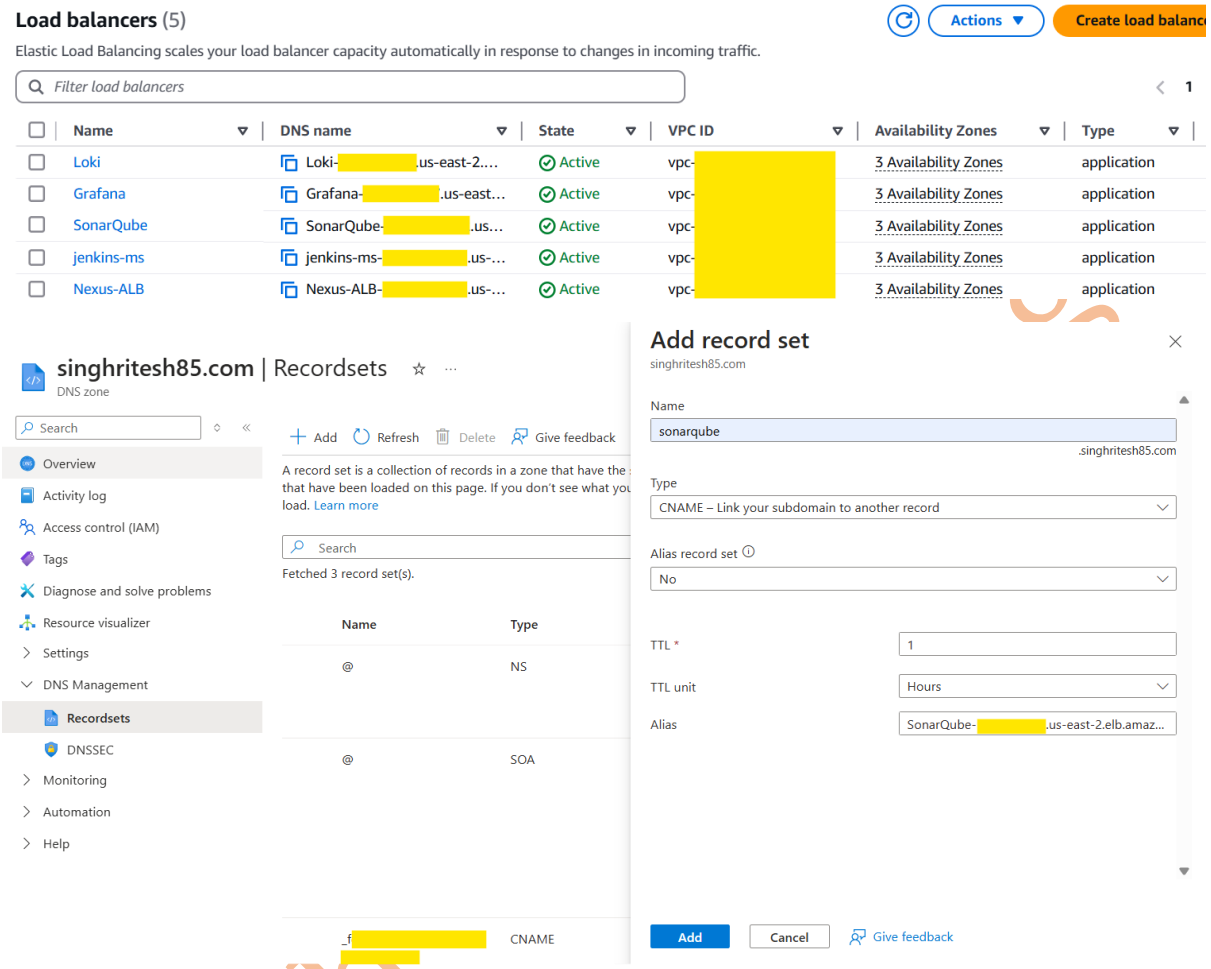



myaccount.microsoft.com/?ref=MeControl
My Account
user1
user1@singhritesh85.com
Overview
Security info
Devices
Password
Organizations
Settings & Privacy
My sign-ins
My Apps
My Groups
Give feedback
Click here
U
user1
user1@singhritesh85.com
Why can't I edit?
Security info
Keep your verification methods and security info up to date.
UPDATE INFO
Password
Make your password stronger, or change it if someone else knows it.
CHANGE PASSWORD
Devices
Disable a lost device and review your connected devices.
Organizations
See all the organizations that you're a part of.


mysignins.microsoft.com/security-info
My Sign-Ins
Overview
Security info
Devices
Password
Organizations
Settings & Privacy
Recent activity
Security info
These are the methods you use to sign into your account or reset your password.
You're using the most advisable sign-in method where it applies.
Add sign-in method
Password
Last updated:
Change
Click here to change the password
Microsoft Authenticator
Push multi-factor authentication (MFA)
Delete
Lost device? Sign out everywhere

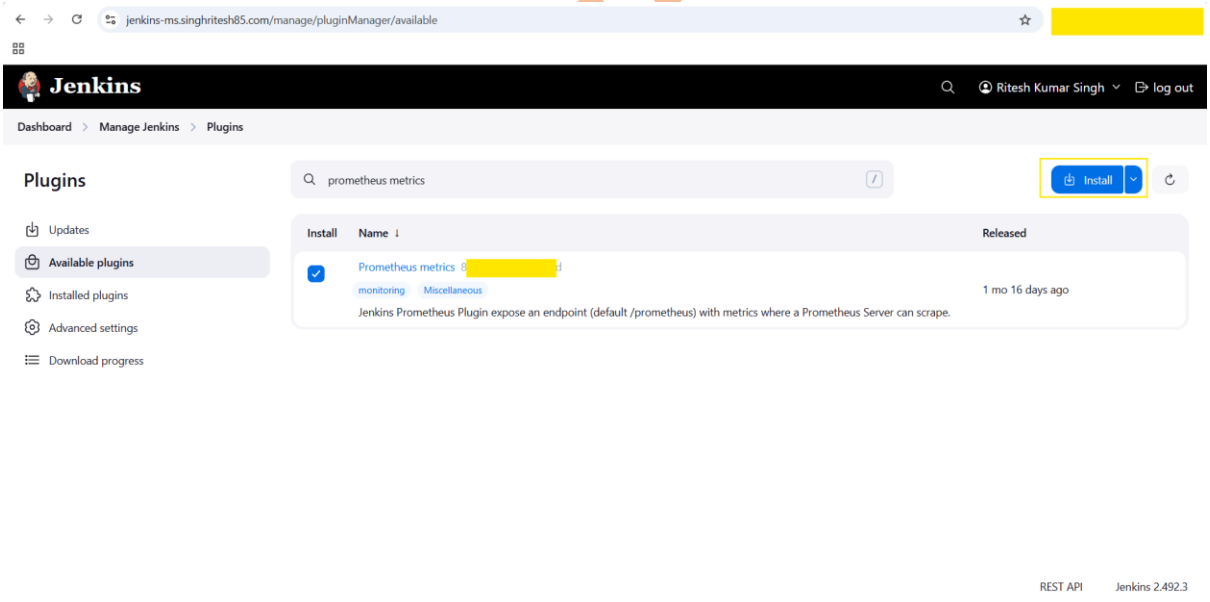

mysignins.microsoft.com/security-info
My Sign-Ins
Overview
Security info
Devices
Password
Organizations
Settings & Privacy
Recent activity
Security info
These are the methods you use to sign into your account or reset your password.
You're using the most advisable sign-in method where it applies.
Add sign-in method
Password
Microsoft Authenticator
Push multi-factor authentication (MFA)
Delete
Lost device? Sign out everywhere
Change your password
User ID
user1@singhritesh85.com
New password
Confirm new password
Cancel
Submit

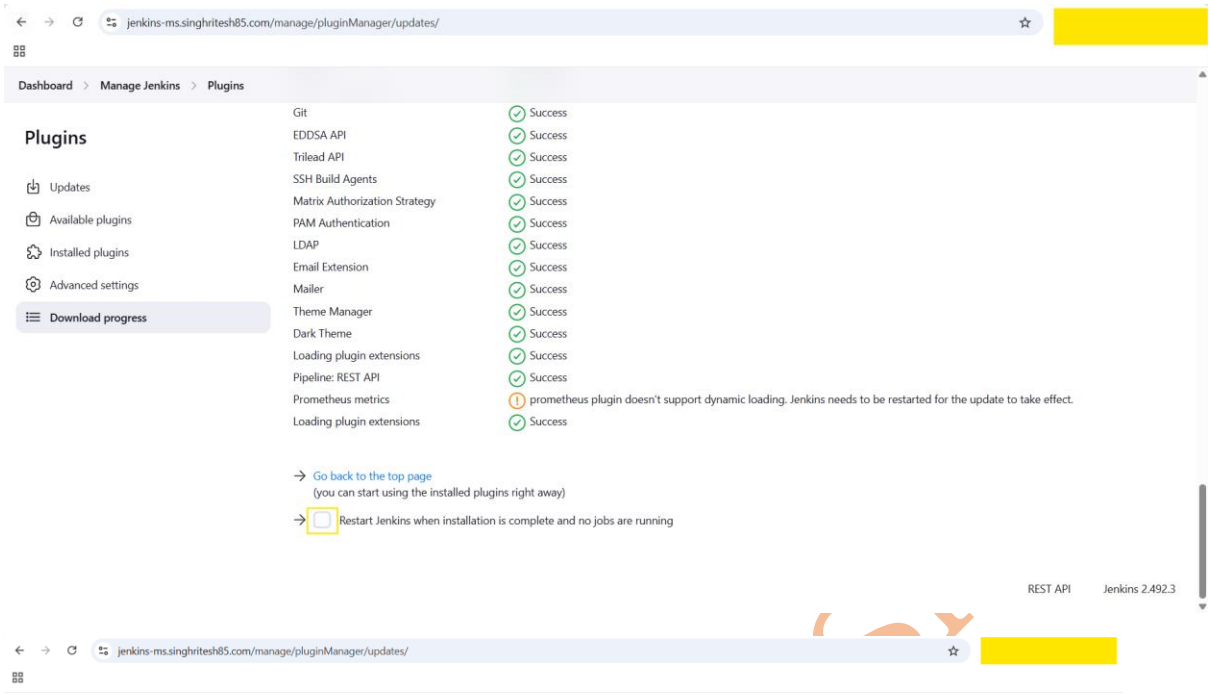
After running the terraform script LoadBalancers for SonarQube, Jenkins, Nexus, Loki and Grafana had been created as shown in the screenshot attached below. I created the record set of type CNAME for SonarQube, Jenkins, Nexus and Grafana in Azure DNS Zone using the DNS Name of the LoadBalancers for example I had shown below the creation of record set for SonarQube.

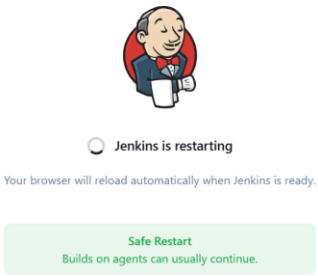


After doing the entry for all the DNS Names (of the LoadBalancers SonarQube, Jenkins, Nexus and Grafana) as shown in the screenshot attached below to create the record set. The final screenshot is as shown in the screenshot attached below.

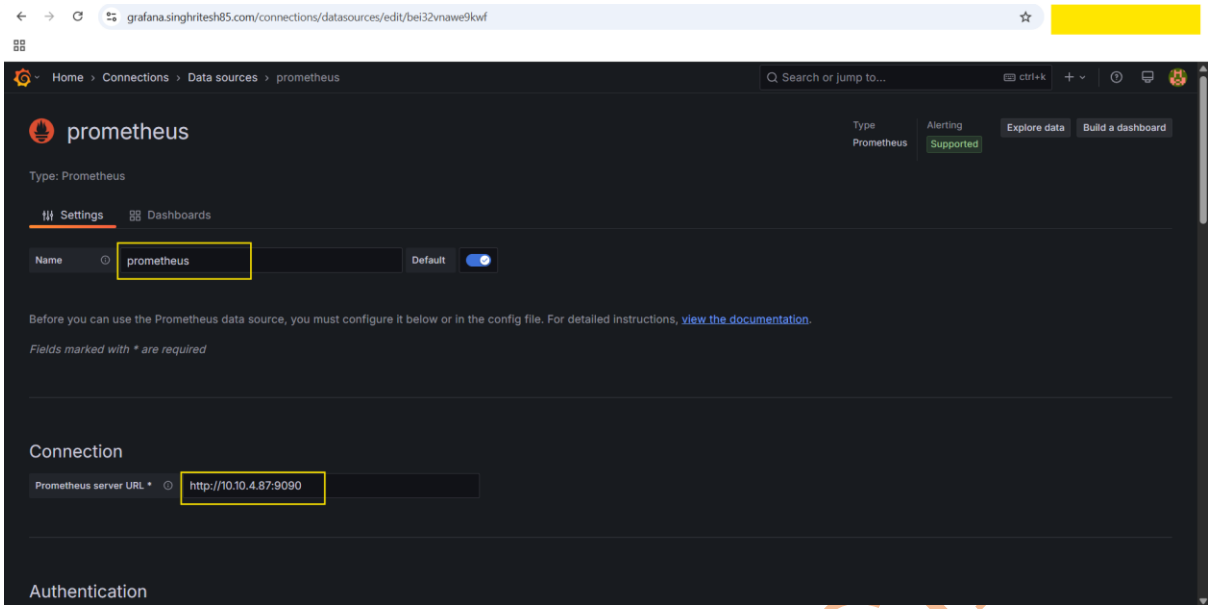
It is possible to monitor Jenkins Job using Prometheus and Grafana and for that I installed **Prometheus Metrics** plugin in Jenkins, after its installation I restarted the Jenkins as shown in the screenshot attached below.

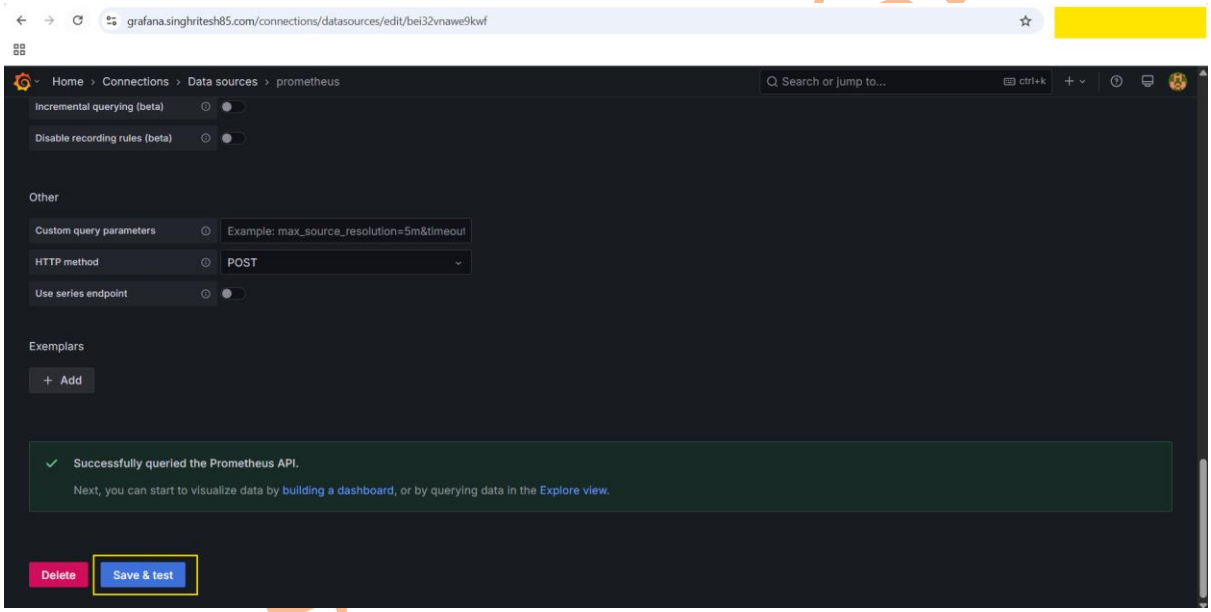


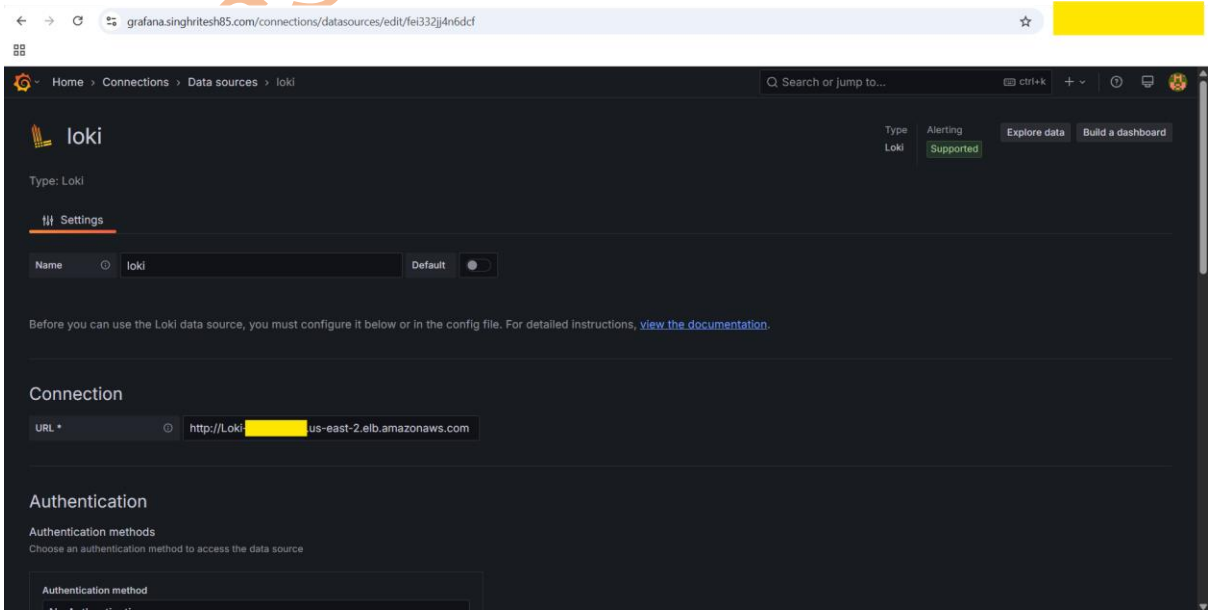


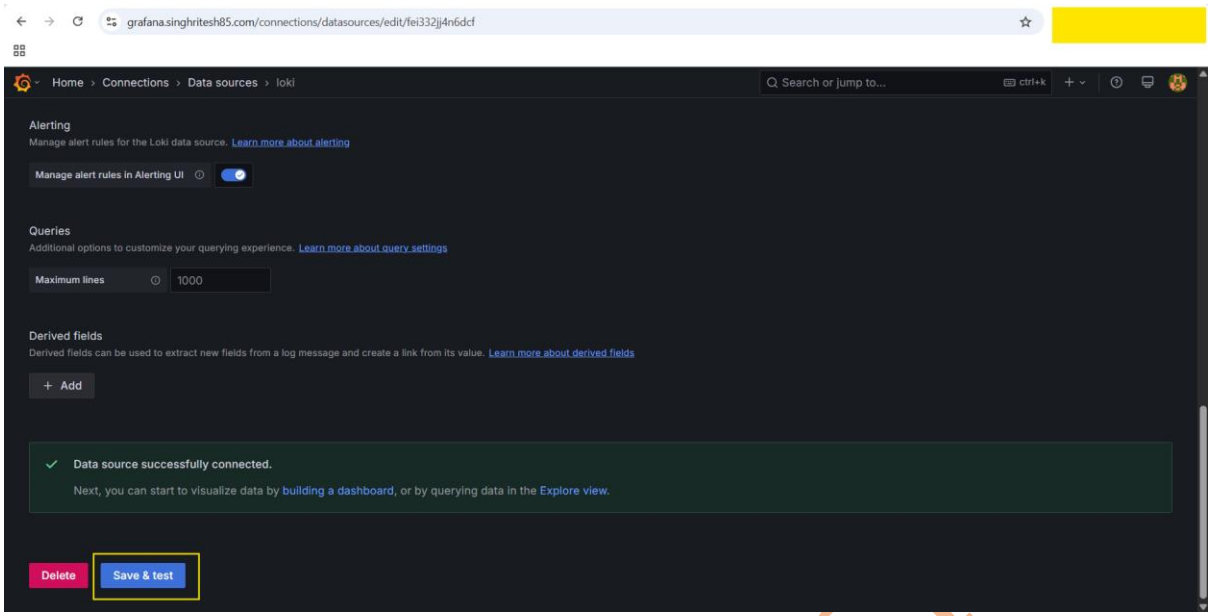


Now, login into the Grafana for the first time and update admin password then login into Grafana and created two data sources each for prometheus and loki as shown in the screenshot attached below.


grafana.singhritesh85.com/connections/datasources/edit/bei32vnawe9kwf
Home › Connections › Data sources › prometheus
Search or jump to...
prometheus
Type
Prometheus
Alerting
Supported
Explore data
Build a dashboard
Type: Prometheus
Settings
Dashboards
Name
prometheus
Default
Before you can use the Prometheus data source, you must configure it below or in the config file. For detailed instructions, view the documentation.
Fields marked with * are required
Connection
Prometheus server URL *
http://10.10.4.87:9090
Authentication


grafana.singhritesh85.com/connections/datasources/edit/bei32vnawe9kwf
Home › Connections › Data sources › prometheus
Search or jump to...
Incremental querying (beta)
Disable recording rules (beta)
Other
Custom query parameters
Example: max_source_resolution=5m&timeout
HTTP method
POST
Use series endpoint
Exemplars
Add
Successfully queried the Prometheus API.
Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.
Delete
Save & test

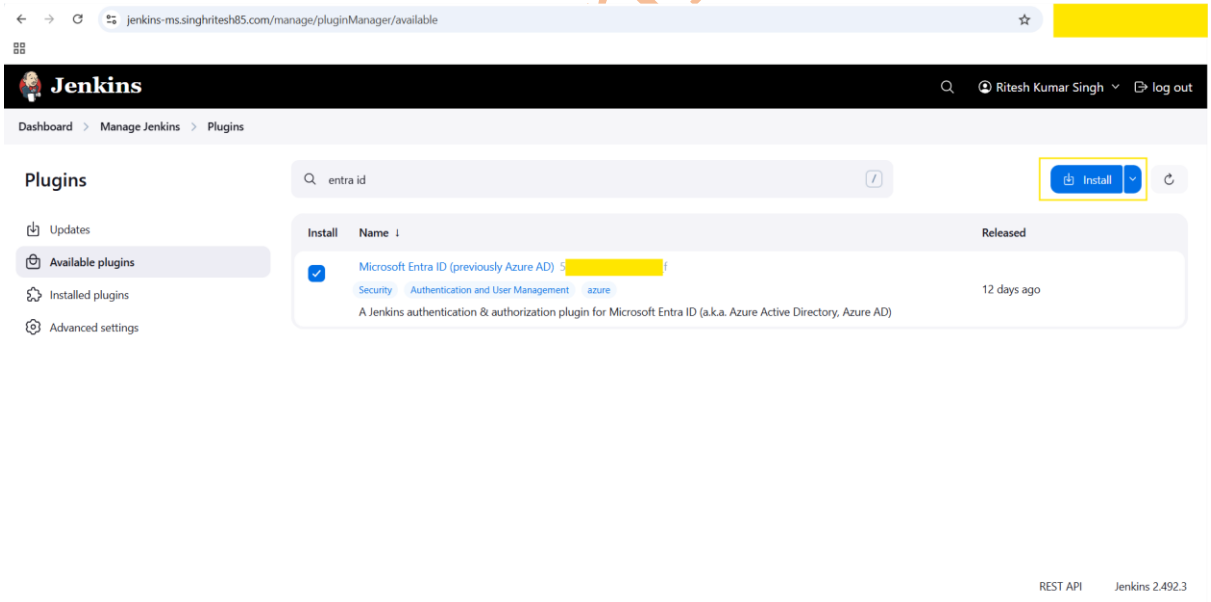

grafana.singhritesh85.com/connections/datasources/edit/fei332jj4n6dcf
Home › Connections › Data sources › loki
Search or jump to...
loki
Type
Loki
Alerting
Supported
Explore data
Build a dashboard
Type: Loki
Settings
Name
loki
Default
Before you can use the Loki data source, you must configure it below or in the config file. For detailed instructions, view the documentation.
Connection
URL *
http://Loki-[redacted]us-east-2.elb.amazonaws.com
Authentication
Authentication methods
Choose an authentication method to access the data source
Authentication method



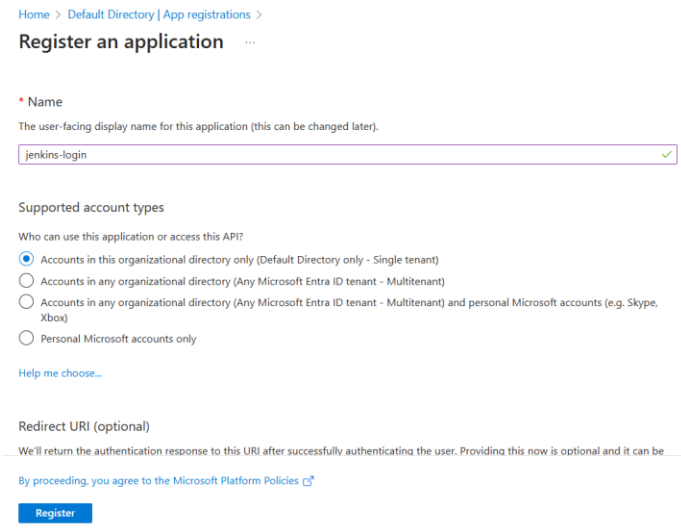
Now I will configure the Integration of Azure Entra ID with Jenkins, SonarQube and Grafana for Authentication

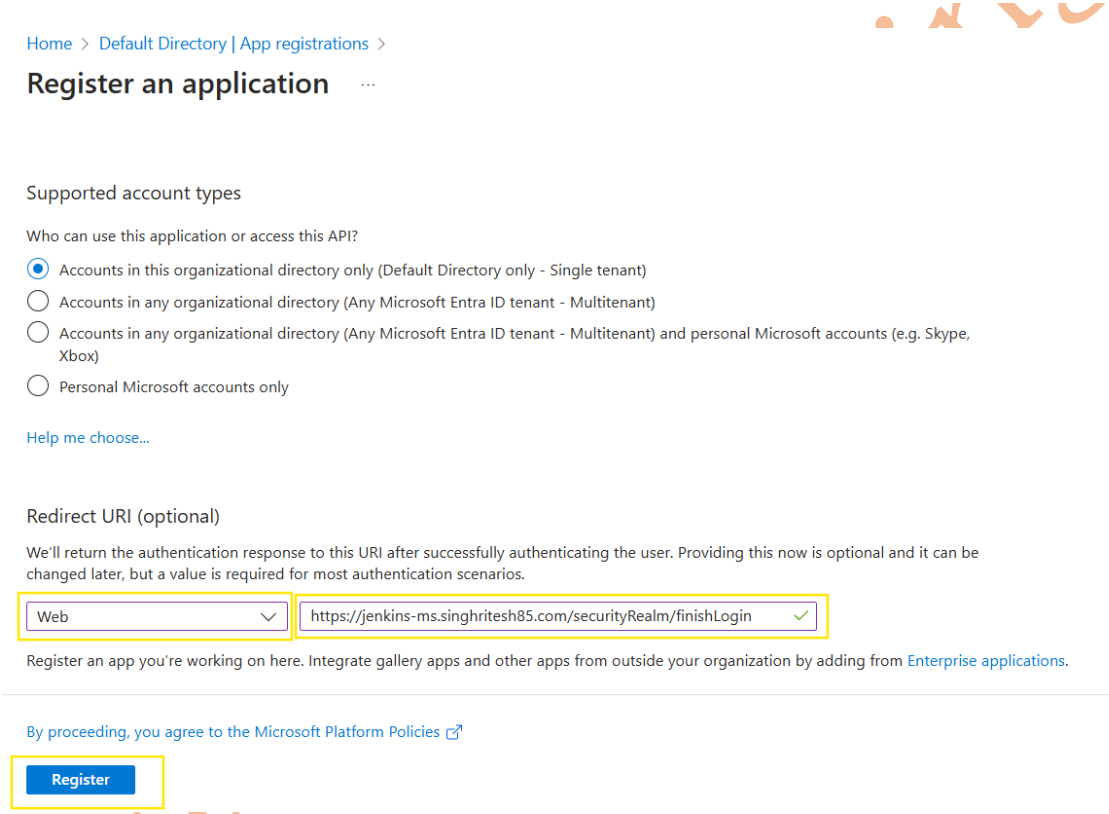
Configuration of Integration of Azure Entra ID with Jenkins

Search from the list of Available plugins for **Microsoft Entra ID** Plugin in Jenkins and install it as shown in the screenshot attached below.

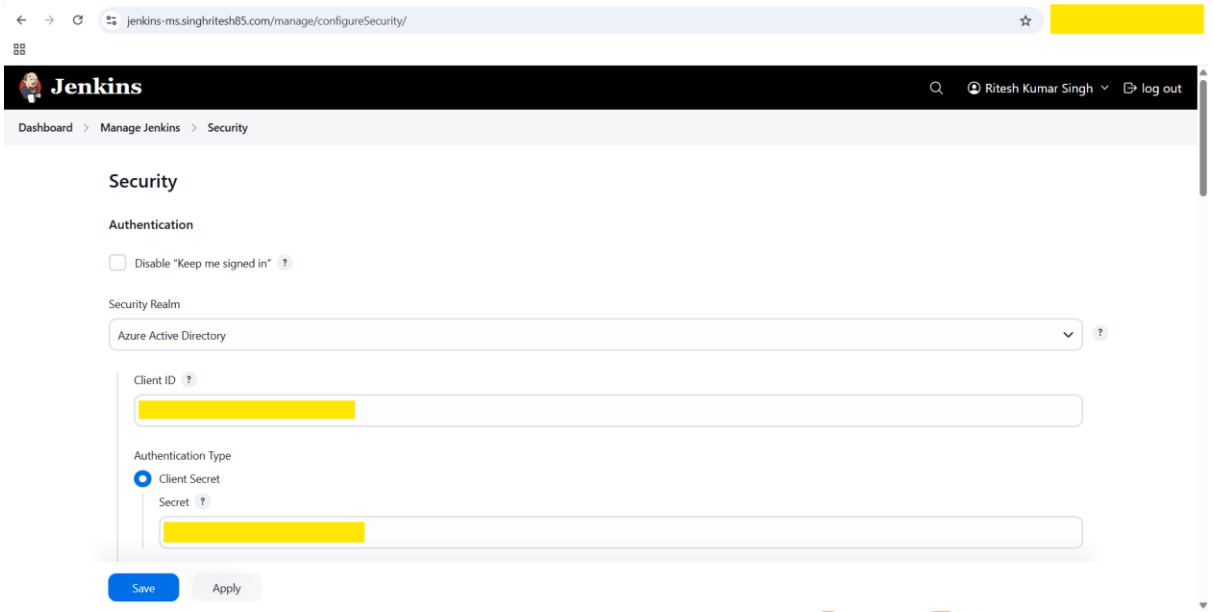


After installation of Azure Entra ID Plugin in Jenkins restart Jenkins and go to the Azure Portal and open Entra ID and create a new App Registration as shown in the screenshot attached below.

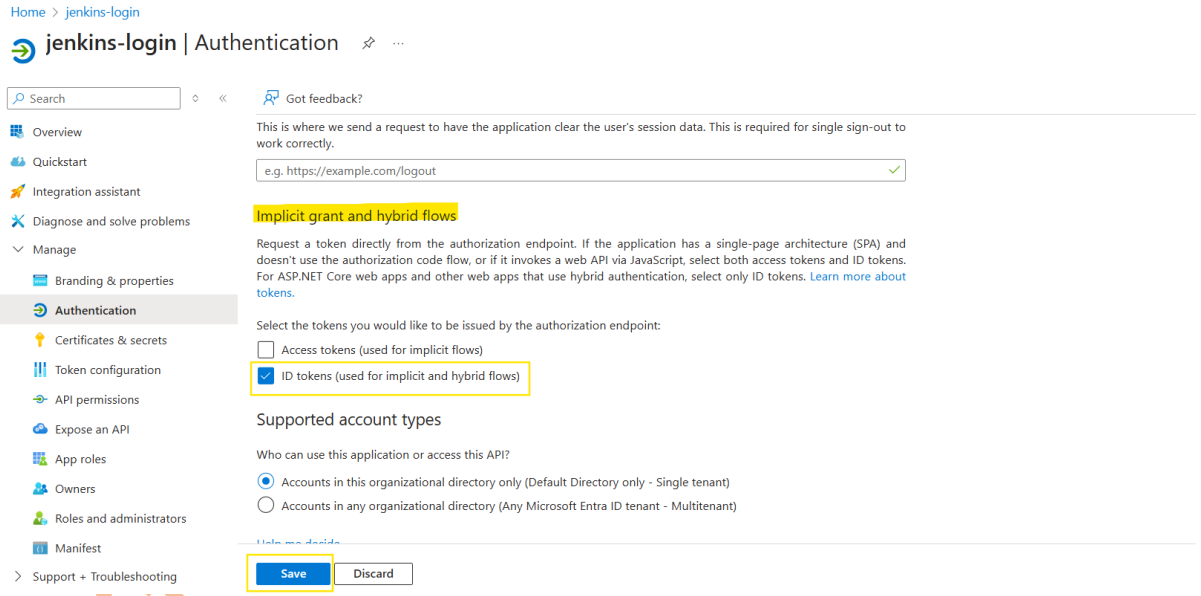




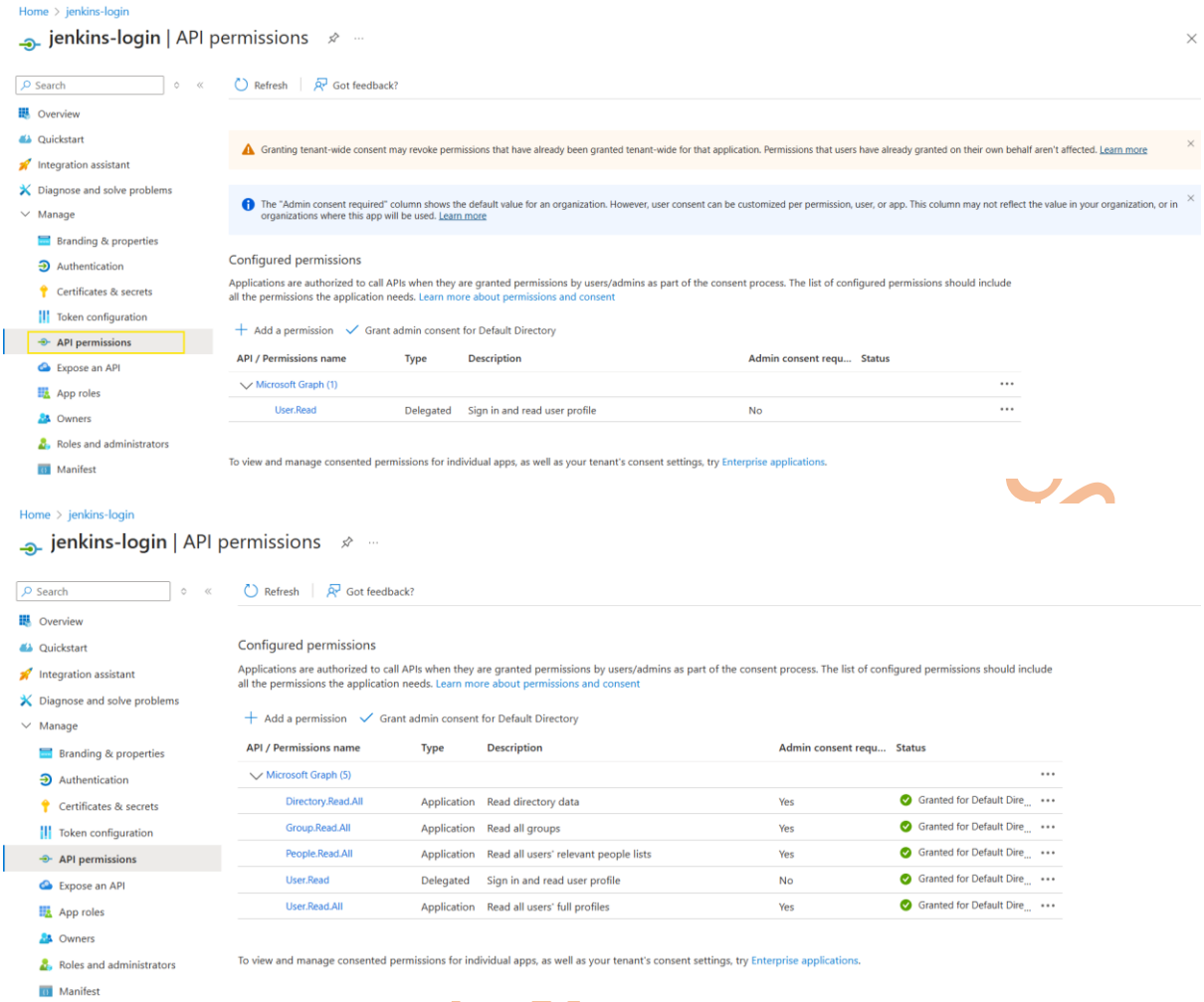
After App Registration in Azure go to Jenkins and **Manage Jenkins > Security > Security Realm and Select the Azure Active Directory** and provide the Client ID, Client Secret and Tenant ID then Save it as shown in screenshot attached below.



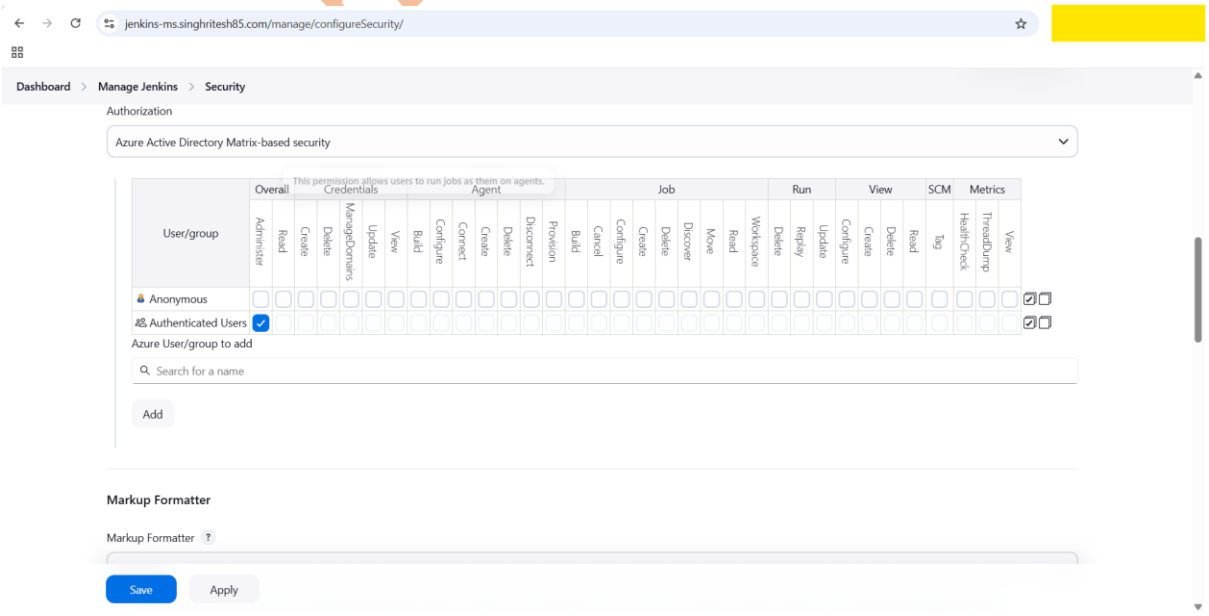
In the Registered Application go to Application and under **Implicit grant and hybrid flows** and select **ID Tokens** as shown in the screenshot attached below.



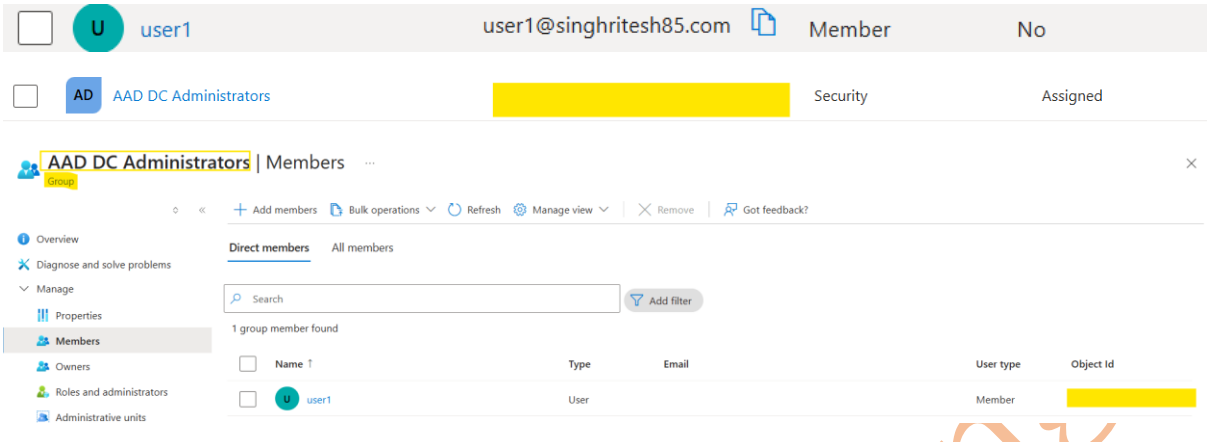
Now select API Permission > Add a permission > Microsoft Graph > Application Permissions People.Read.All, Group.Read.All, User.Read.All and Directory.Read.All in the created Registered App. Then Click on Grant admin consent as shown in the screenshot attached below



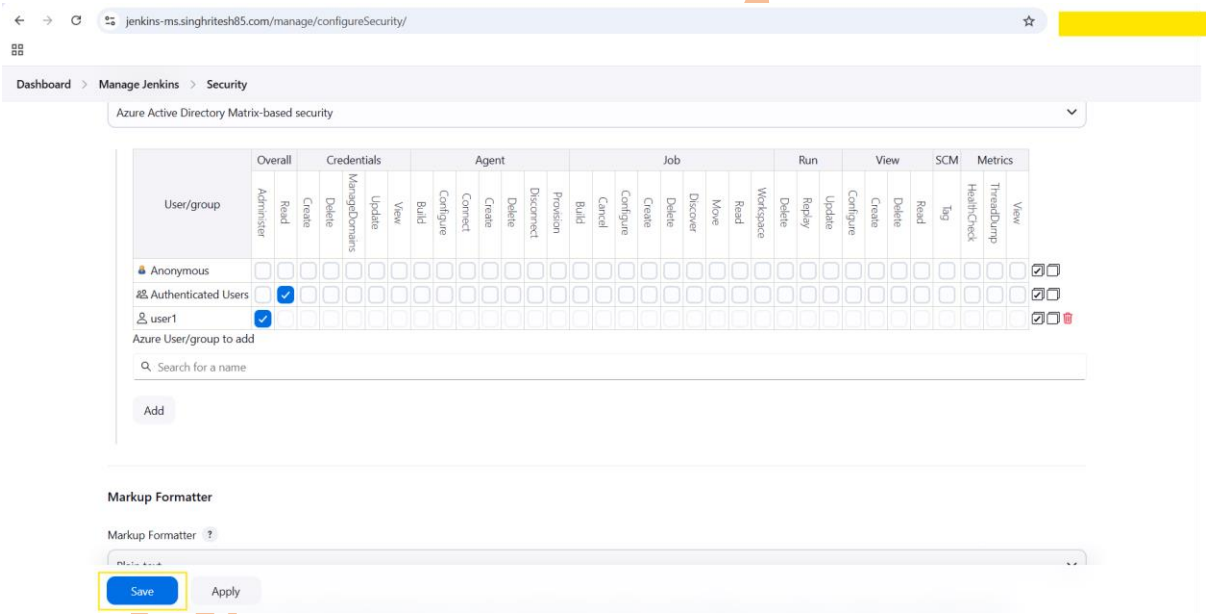
For Authorization select Azure Active Directory Matrix based Authorization as shown in the screenshot attached below, to login for the very first time provide Authenticated users as overall access and the login with a user and add other users if needed.



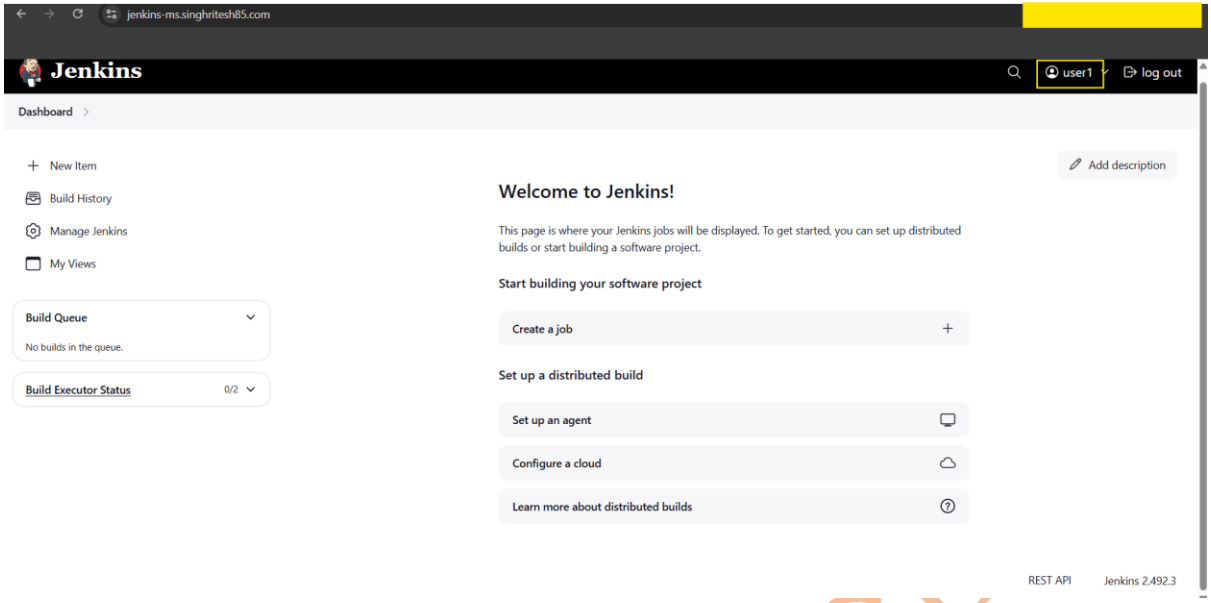
Below screenshot shows the user and group in Azure Entra ID Which I had created initially using the terraform.



For the very first time I logged-in with a user then I provided user1 as administrator privileges and removed the Authenticated user overall administrator access (and provided overall Read access) which I provided before first time logged-in as shown in the screenshot attached below.

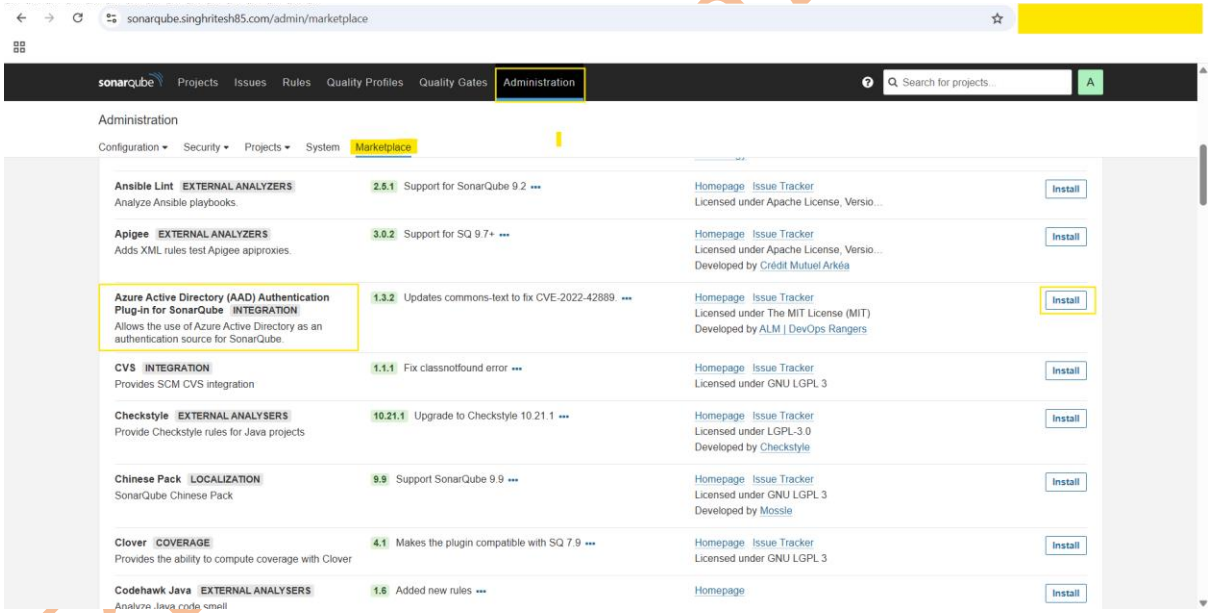


Below screenshot shows the access which I had when I logged-in with user1.

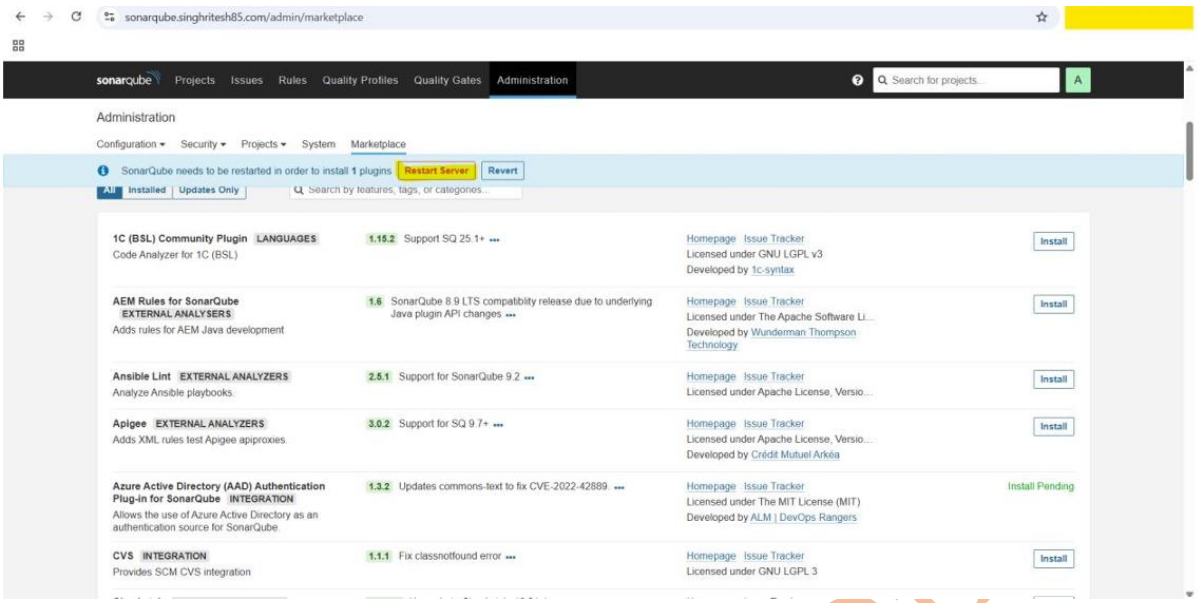


Integration of Azure Entra ID with SonarQube

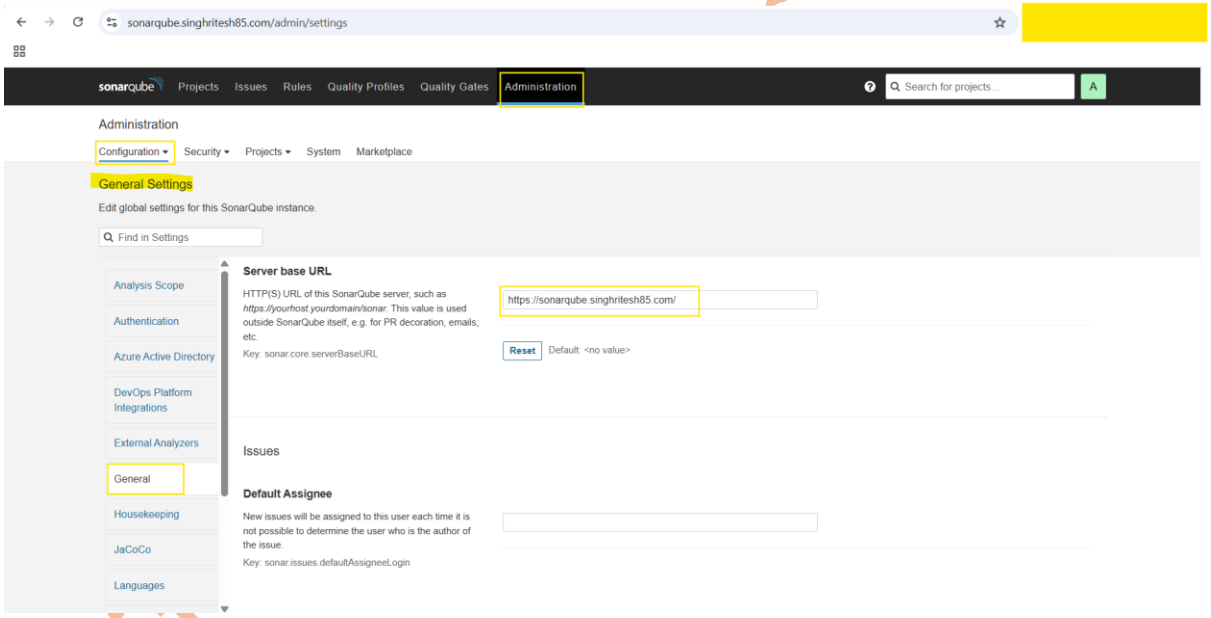
To Integrate Azure Active Directory with SonarQube I installed Azure Active Directory (AAD) Authentication Plug-in for SonarQube as shown in the screenshot attached below.



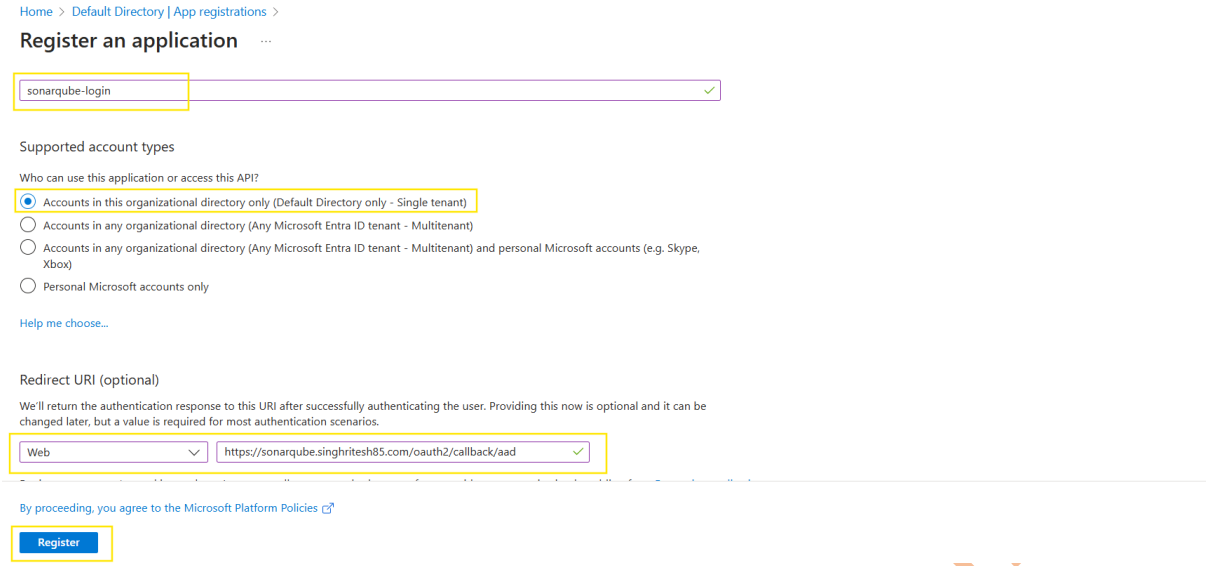
Then restarted the sonarqube server as shown in the screenshot attached below.



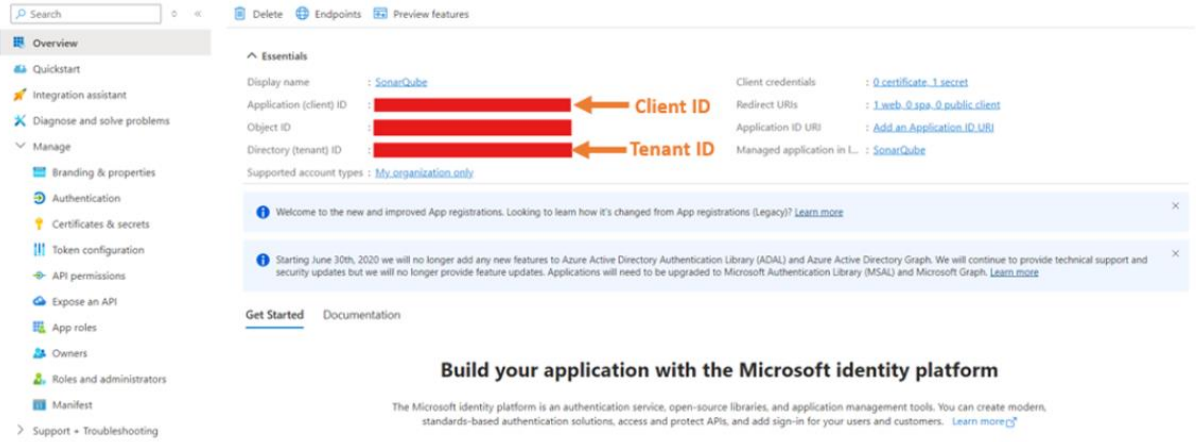
Go to SonarQube UI then to Administration > Configuration > General Settings > General and Add the Server Base URL as shown in the screenshot attached below.



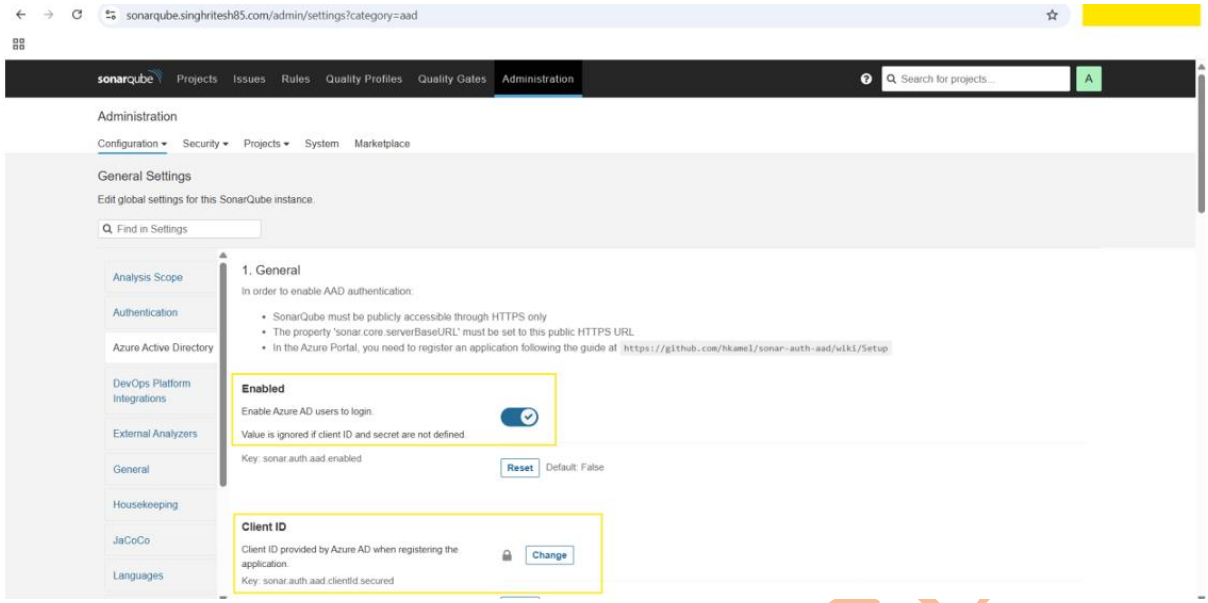
Create the App Registration in Azure Entra ID as shown in the screenshot attached below.

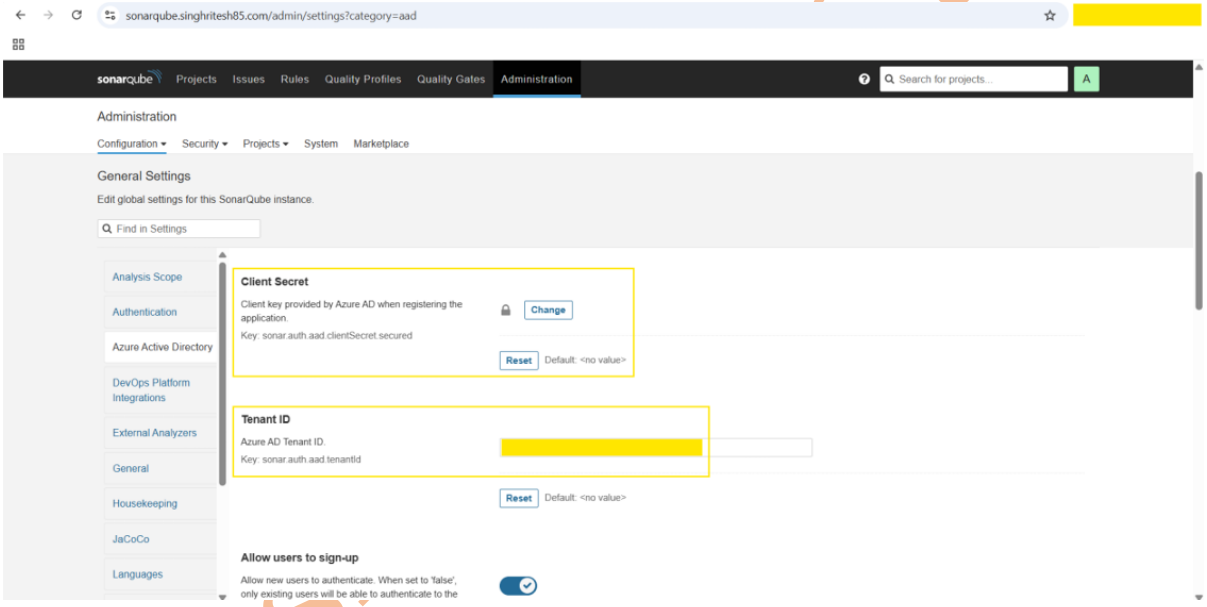


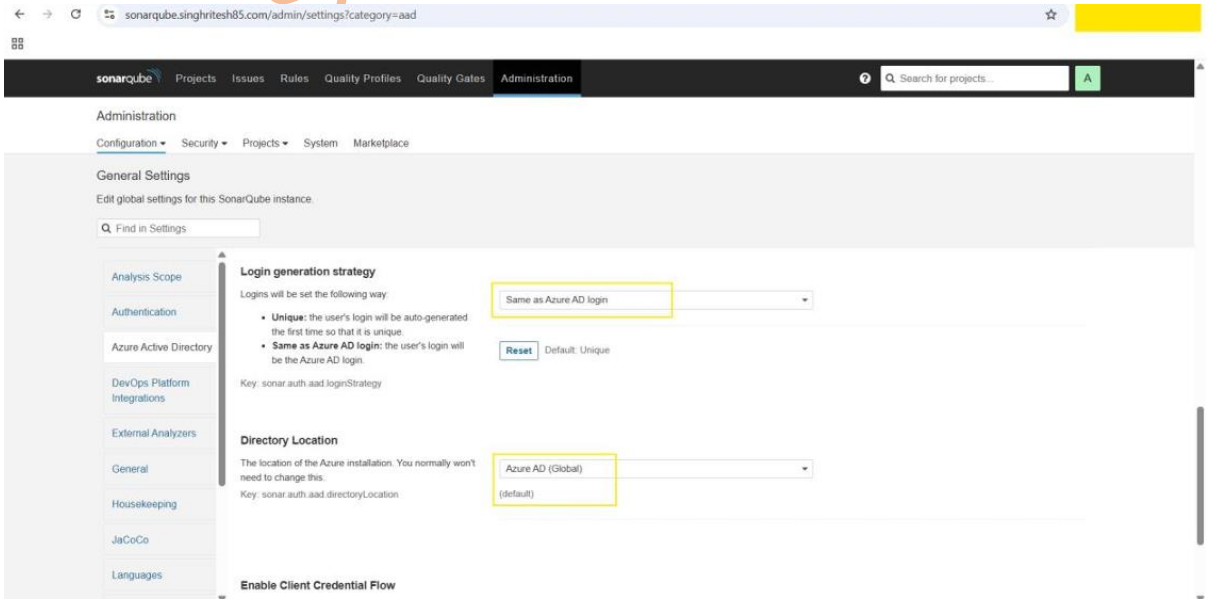
Create the client secret as shown in the screenshot attached below.



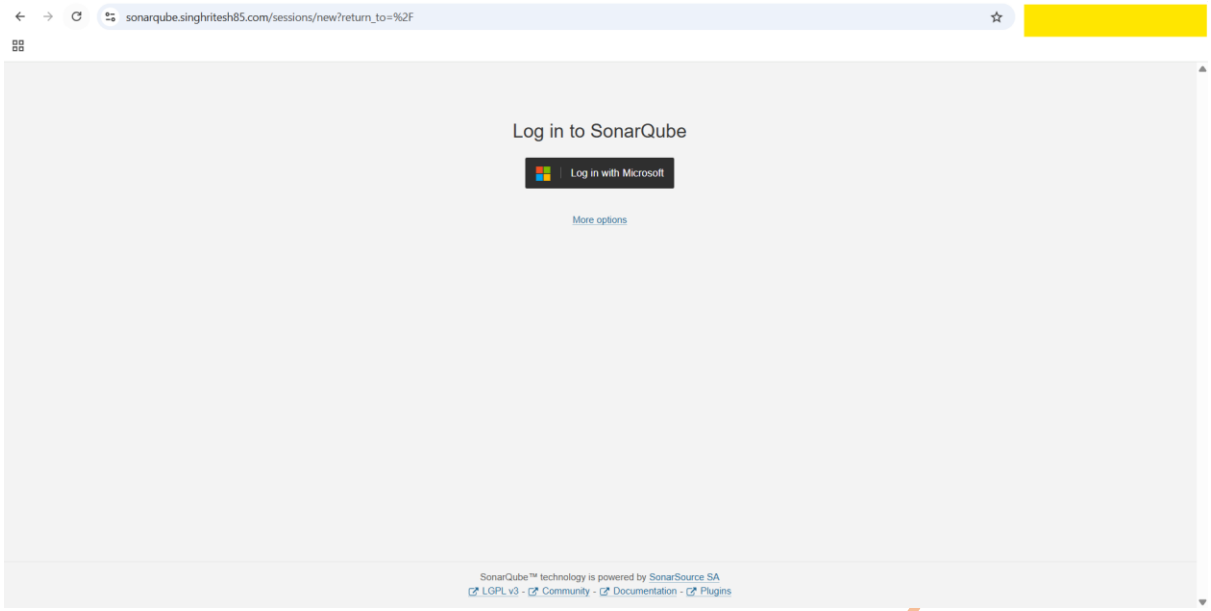
Now go to Sonarqube UI and then to **Administration > Configuration > General Settings > Azure Active Directory** as shown in the screenshot attached below and enable Azure AD users to login then provide the **client id**, **client secret** and **tenant id** as shown in the screenshot attached below.


sonarqube.singhritesh85.com/admin/settings?category=aad
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
Administration
Configuration
Security
Projects
System
Marketplace
General Settings
Edit global settings for this SonarQube instance.
Find in Settings
Analysis Scope
Authentication
Azure Active Directory
DevOps Platform Integrations
External Analyzers
General
Housekeeping
JaCoCo
Languages
1. General
In order to enable AAD authentication:
SonarQube must be publicly accessible through HTTPS only
The property 'sonar.core.serverBaseURL' must be set to this public HTTPS URL
In the Azure Portal, you need to register an application following the guide at https://github.com/hkamel/sonar-auth-aad/wiki/Setup
Enabled
Enable Azure AD users to login.
Value is ignored if client ID and secret are not defined.
Key: sonar.auth.aad.enabled
Reset
Default: False
Client ID
Client ID provided by Azure AD when registering the application.
Key: sonar.auth.aad.clientId.secured
Change

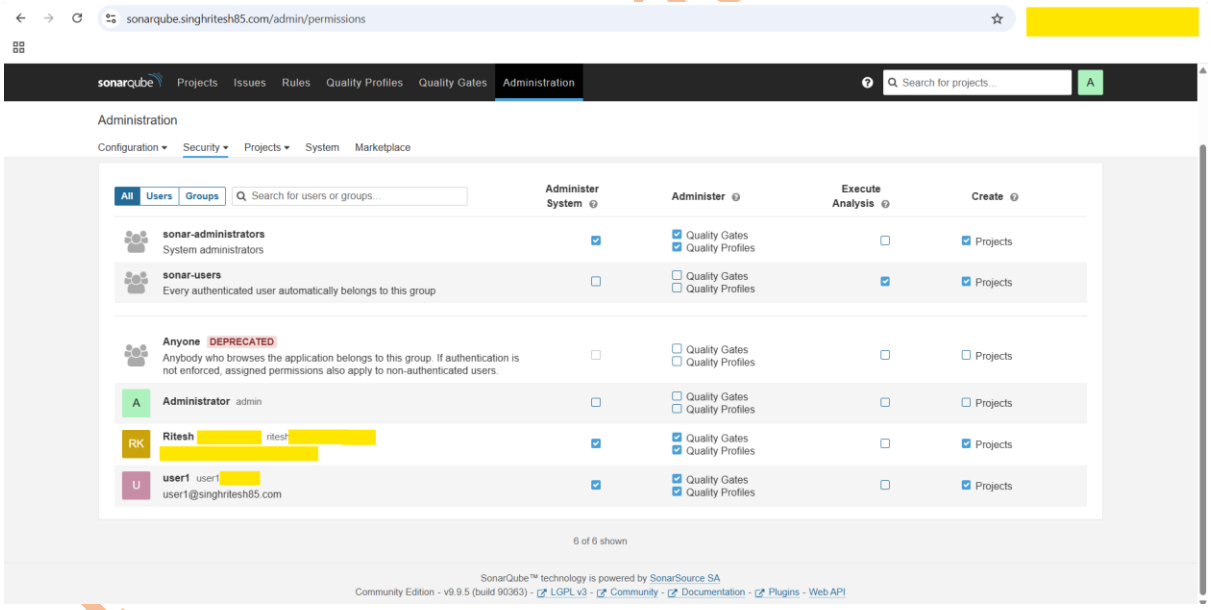

sonarqube.singhritesh85.com/admin/settings?category=aad
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
Administration
Configuration
Security
Projects
System
Marketplace
General Settings
Edit global settings for this SonarQube instance.
Find in Settings
Analysis Scope
Authentication
Azure Active Directory
DevOps Platform Integrations
External Analyzers
General
Housekeeping
JaCoCo
Languages
Client Secret
Client key provided by Azure AD when registering the application.
Key: sonar.auth.aad.clientSecret.secured
Change
Reset
Default: <no value>
Tenant ID
Azure AD Tenant ID.
Key: sonar.auth.aad.tenantId
Reset
Default: <no value>
Allow users to sign-up
Allow new users to authenticate. When set to 'false', only existing users will be able to authenticate to the


sonarqube.singhritesh85.com/admin/settings?category=aad
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
Administration
Configuration
Security
Projects
System
Marketplace
General Settings
Edit global settings for this SonarQube instance.
Find in Settings
Analysis Scope
Authentication
Azure Active Directory
DevOps Platform Integrations
External Analyzers
General
Housekeeping
JaCoCo
Languages
Login generation strategy
Logins will be set the following way:
Unique: the user's login will be auto-generated the first time so that it is unique.
Same as Azure AD login: the user's login will be the Azure AD login.
Same as Azure AD login
Reset
Default: Unique
Key: sonar.auth.aad.loginStrategy
Directory Location
The location of the Azure installation. You normally won't need to change this:
Key: sonar.auth.aad.directoryLocation
Azure AD (Global)
(default)
Enable Client Credential Flow

Then logout from SonarQube UI and login again as shown in the screenshot attached below.

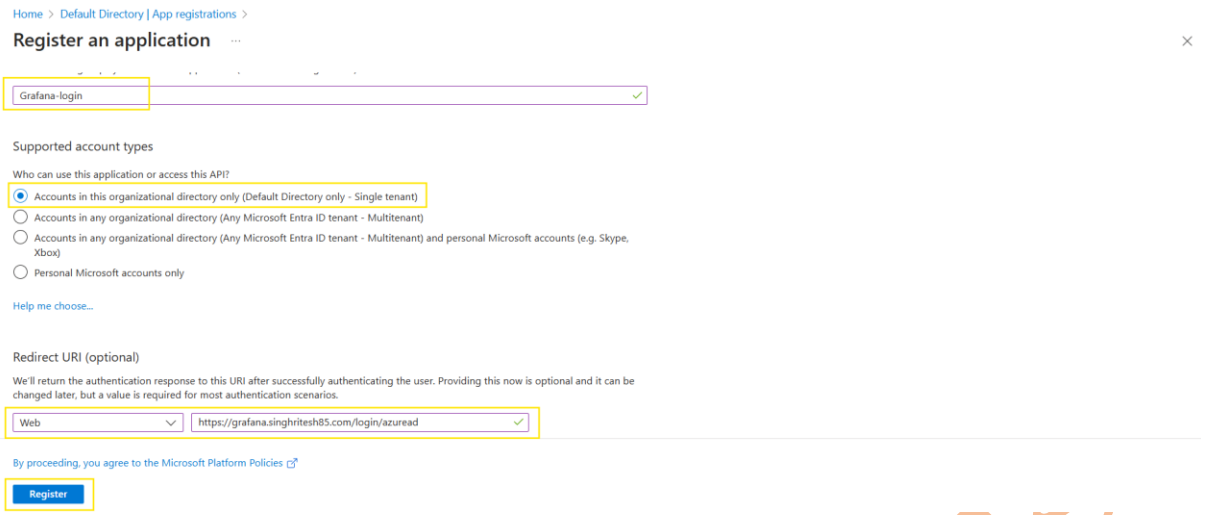


After the user will login once, you can change their permission as shown in screenshot attached below.

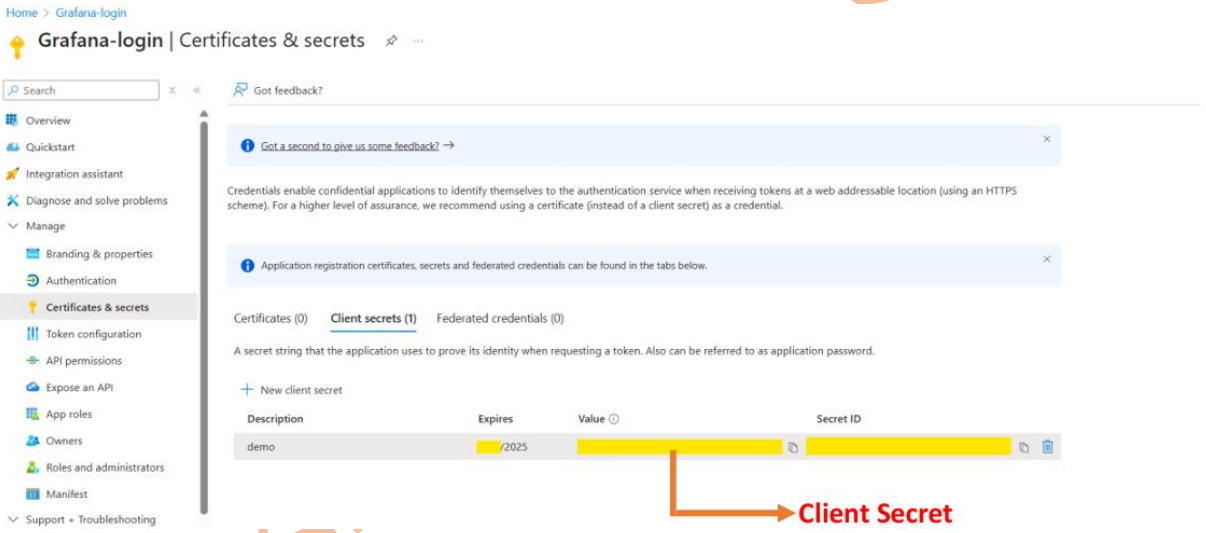


Integration of Azure Entra ID with Grafana

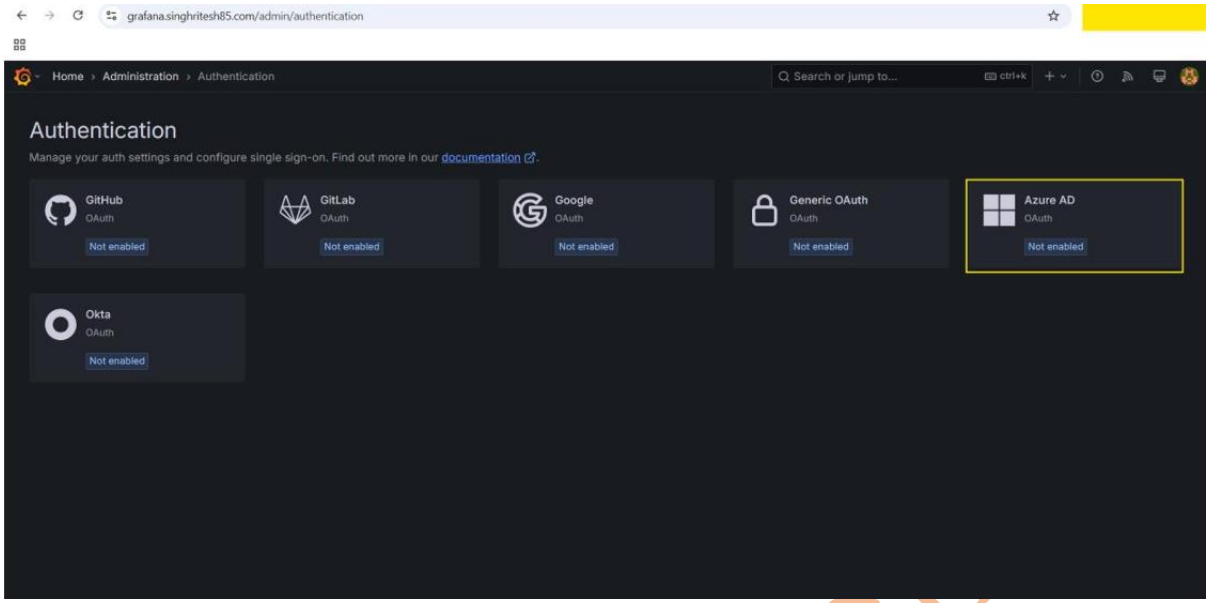
To integrate Azure Entra ID with Grafana I created an Azure Entra ID Service Principal (using APP Registration) as shown in the screenshot attached below. Go to **App Registration > New Registration** and configure a new Application as shown in the screenshot attached below.



Created the Secrets for Registered App Grafana-login in Azure Entra ID as shown in the screenshot attached below.

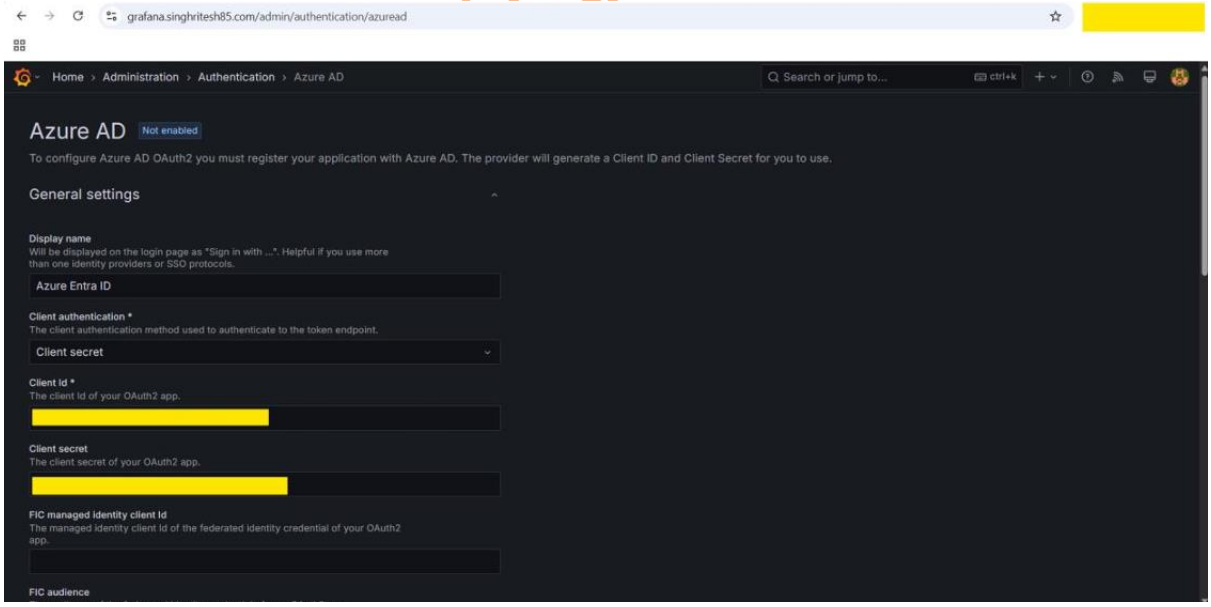


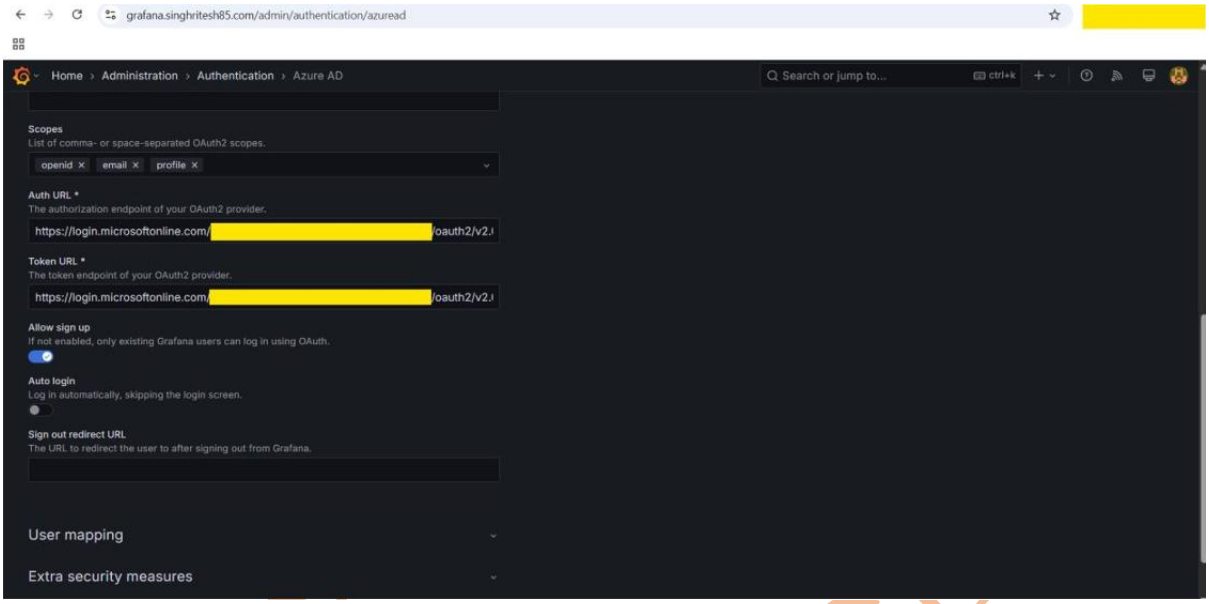
Now, login into the Grafana and update admin password then login into Grafana and Go to Grafana **Home > Administration > Authentication** and **enable the Azure AD OAuth** as shown in the screenshot attached below.

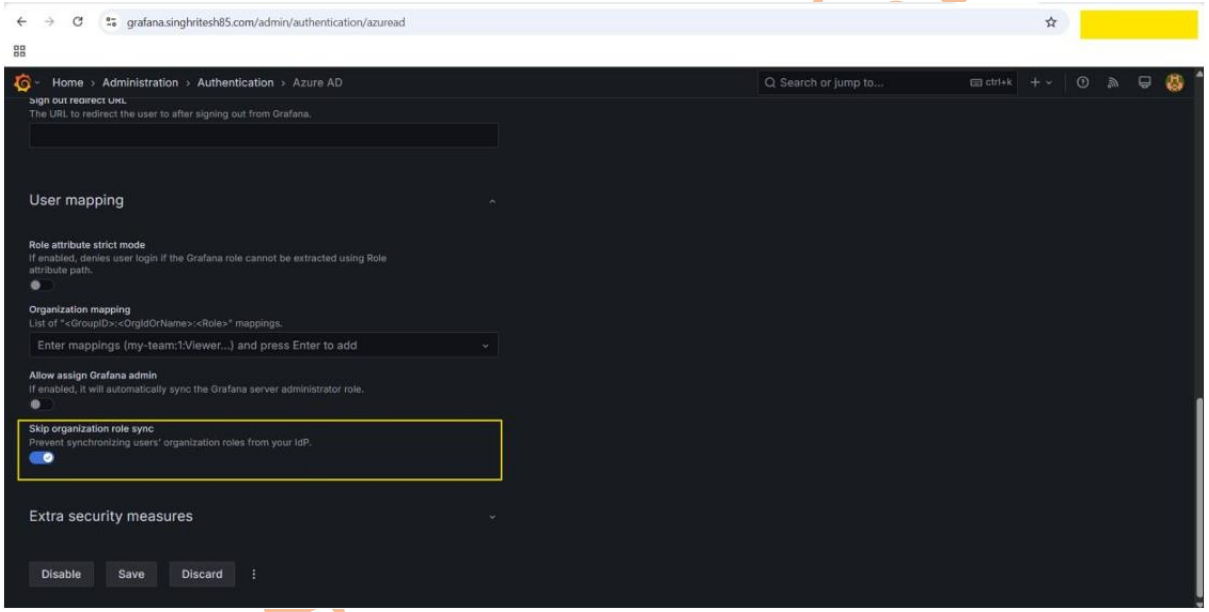


Provide the Client ID, Client Secret, Tenant ID and enable Allow Sign up and enable Skip organization role sync as shown in the screenshot attached below.

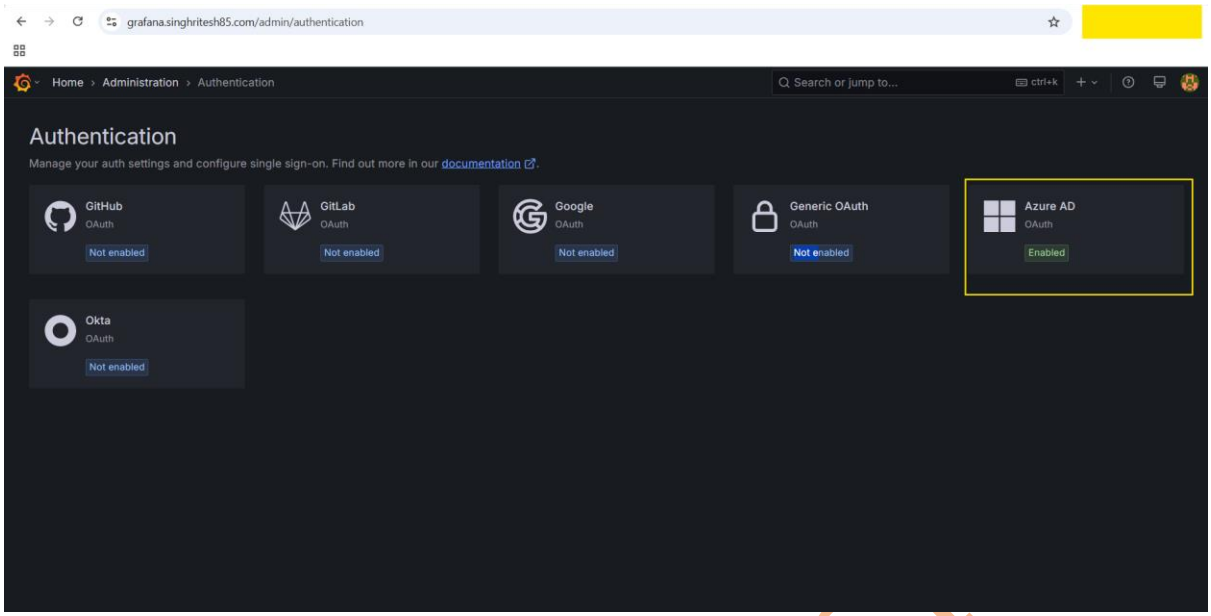
I enabled the **Skip organization role sync** otherwise Grafana will Sync the Azure Entra ID users with Main Org. Role as **Viewer** and Grafana Administrator cannot change it further but if I enabled **Skip organization role sync** then Grafana Administrator can change the viewer Role and can assign another Role as Editor or Admin. In this demonstration I created two users in Azure Entra ID user1 and user2. User1, I had provided as Administrator Role and user2 as default viewer access in Main Organisation (**Main Organisation always have Organisation ID 1**).







Now the Azure AD OAuth is enable as shown in the screenshot attached below.



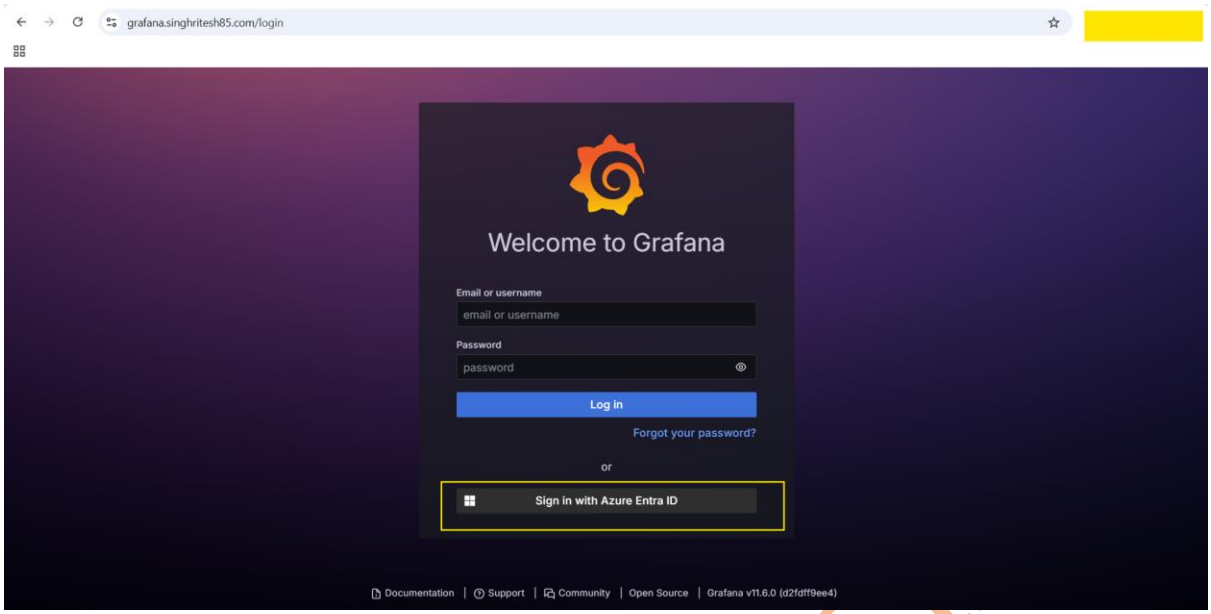
Now go to the Grafana Server and open the file **/etc/grafana/grafana.ini** and edit **root_url** under the [server] then restart the **grafana-server** service as shown in the screenshot attached below.

```
[root@                    ~]# vim /etc/grafana/grafana.ini
```

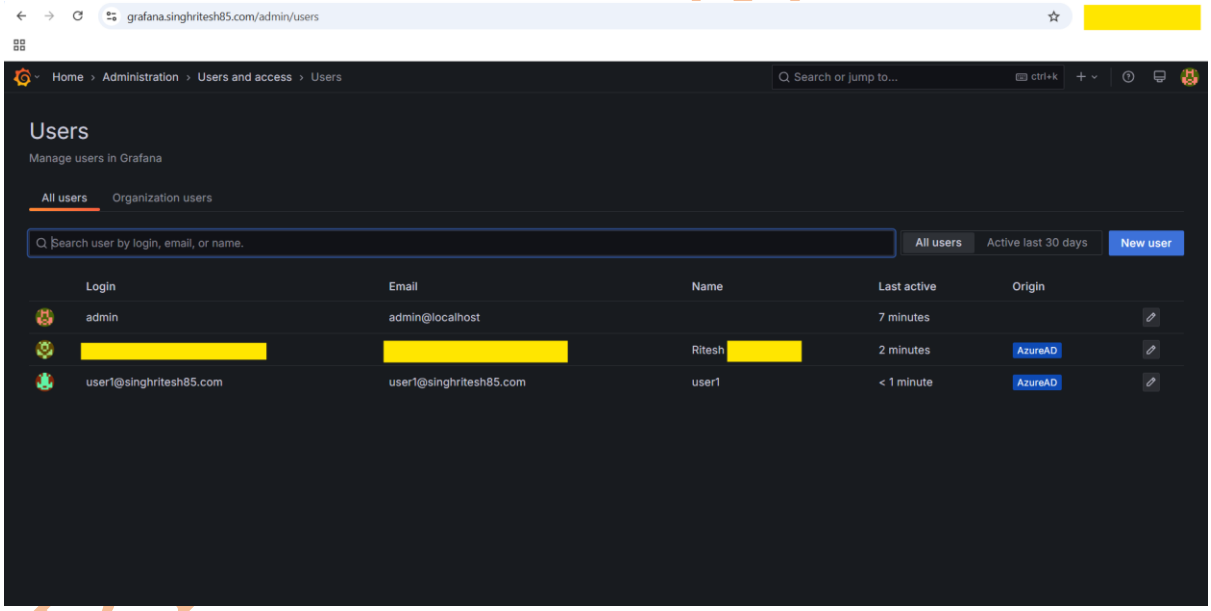
```
#################################### Server ####################################
[server]
# Protocol (http, https, h2, socket)
;protocol = http

# Minimum TLS version allowed. By default, this value is empty. Accepted values are: TLS1.2, TLS1.3. If nothing is set TLS1.2 would be taken
;min_tls_version = ""

# The ip address to bind to, empty will bind to all interfaces
;http_addr =

# The http port to use
;http_port = 3000

# The public facing domain name used to access grafana from a browser
;domain = localhost

# Redirect to correct domain if host header does not match domain
# Prevents DNS rebinding attacks
;enforce_domain = false

# The full public facing url you use in browser, used for redirects and emails
# If you use reverse proxy and sub path specify full url (with sub path)
root_url = "https://grafana.singhritesh85.com/"        ###%(protocol)s://%(domain)s:%(http_port)s/

# Serve Grafana from subpath specified in `root_url` setting. By default it is set to `false` for compatibility reasons.
;serve_from_sub_path = false

# Log web requests
;router_logging = false
```

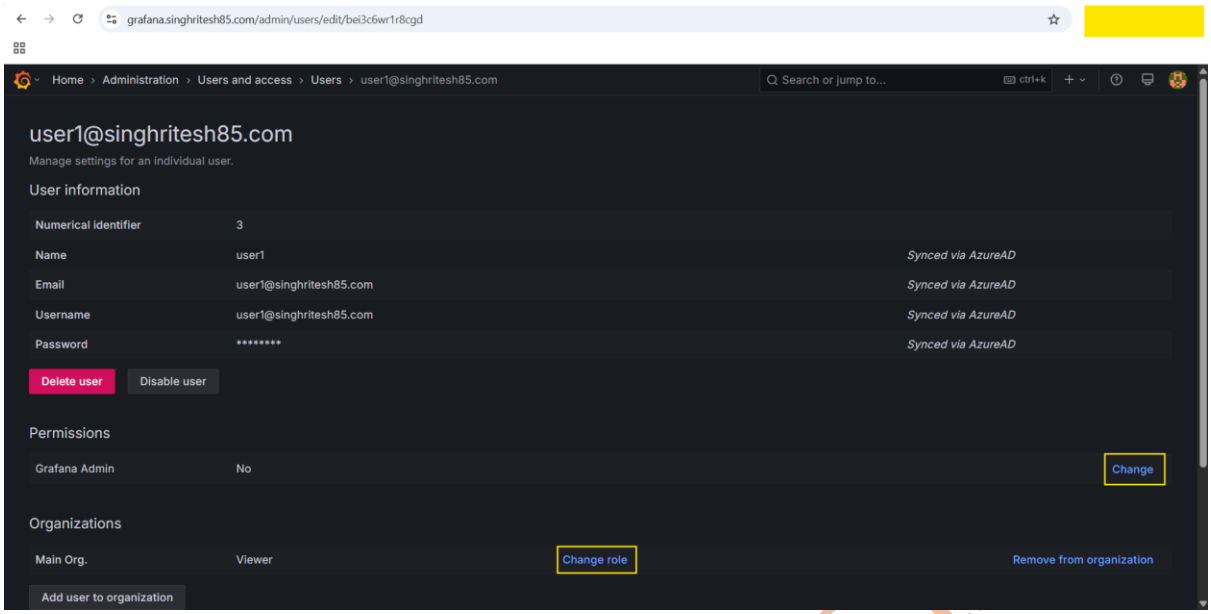
```
[root@                ~]# systemctl restart grafana-server.service
[root@                ~]# systemctl status grafana-server.service
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2025                UTC; 7s ago
     Docs: http://docs.grafana.org
```

Now, Grafana login dashboard will show the option of sign-in with Azure Entra ID as shown in the screenshot attached below. I logged-in to the Grafana dashboard with the Azure Entra ID user as shown in the screenshot attached below.

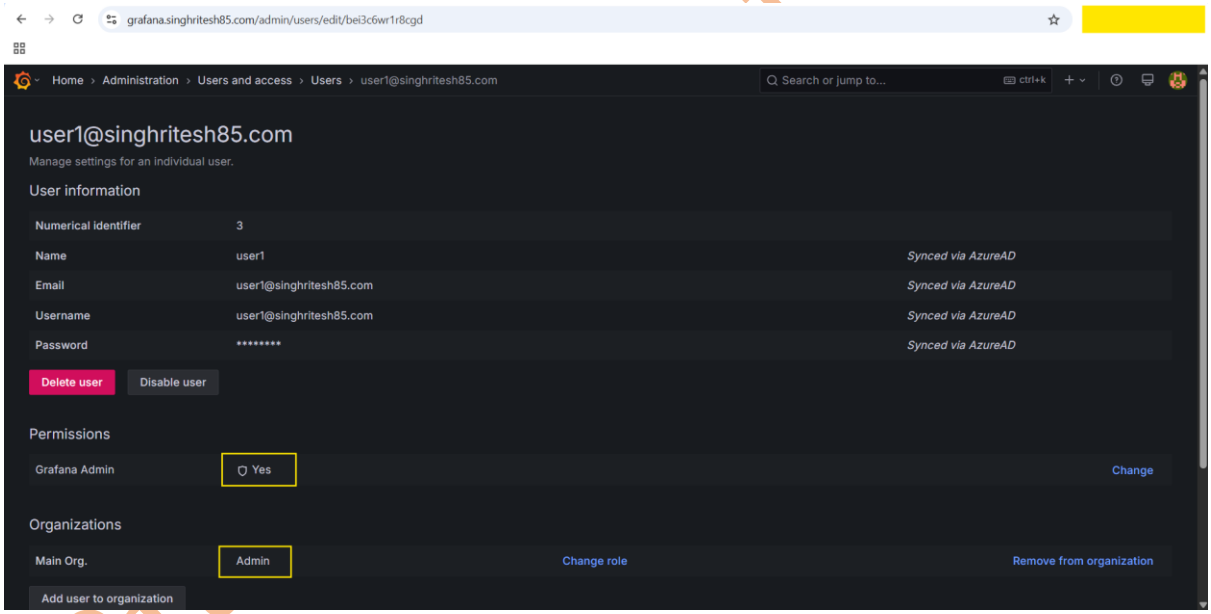


For the first time when a user logged-in, they had viewer access which Grafana Administrator can change as per the requirement as shown in the screenshot attached below.





I will provide Admin access to user1 as shown in the screenshot attached below.

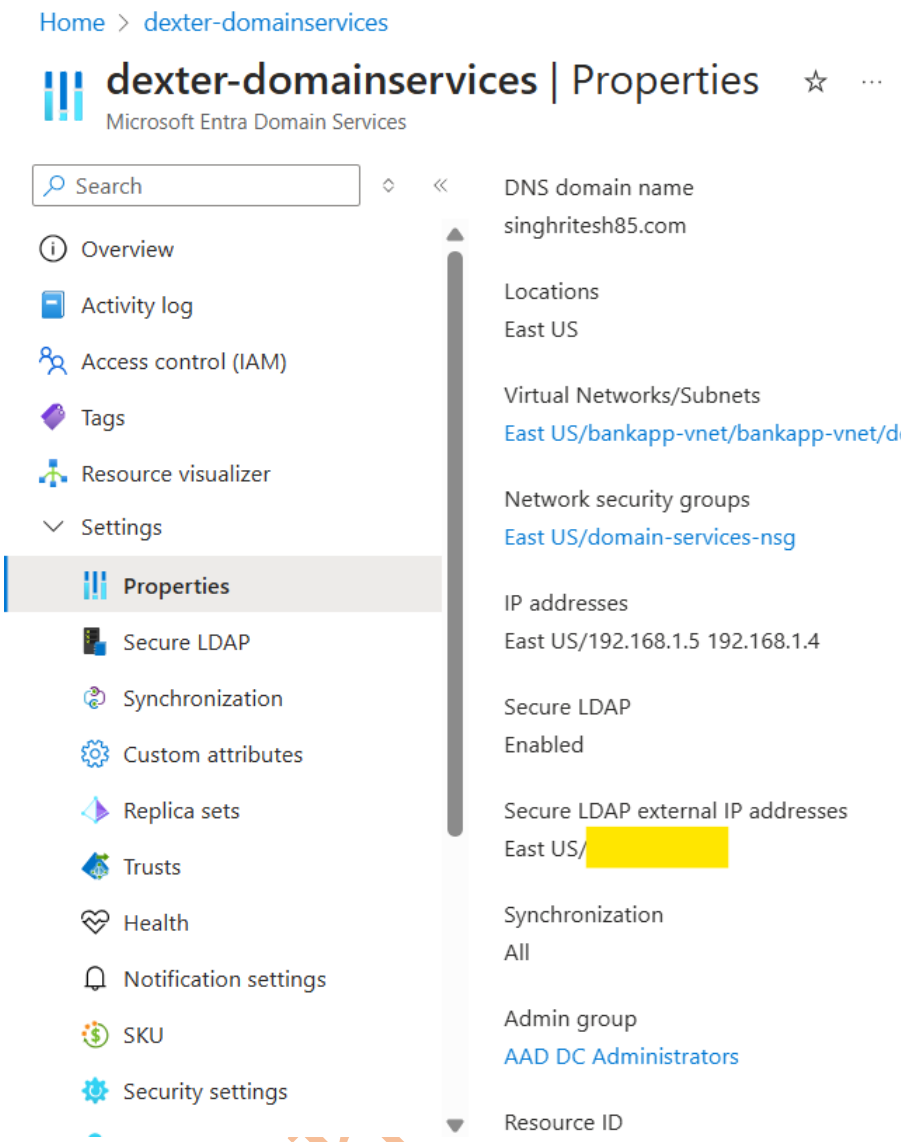


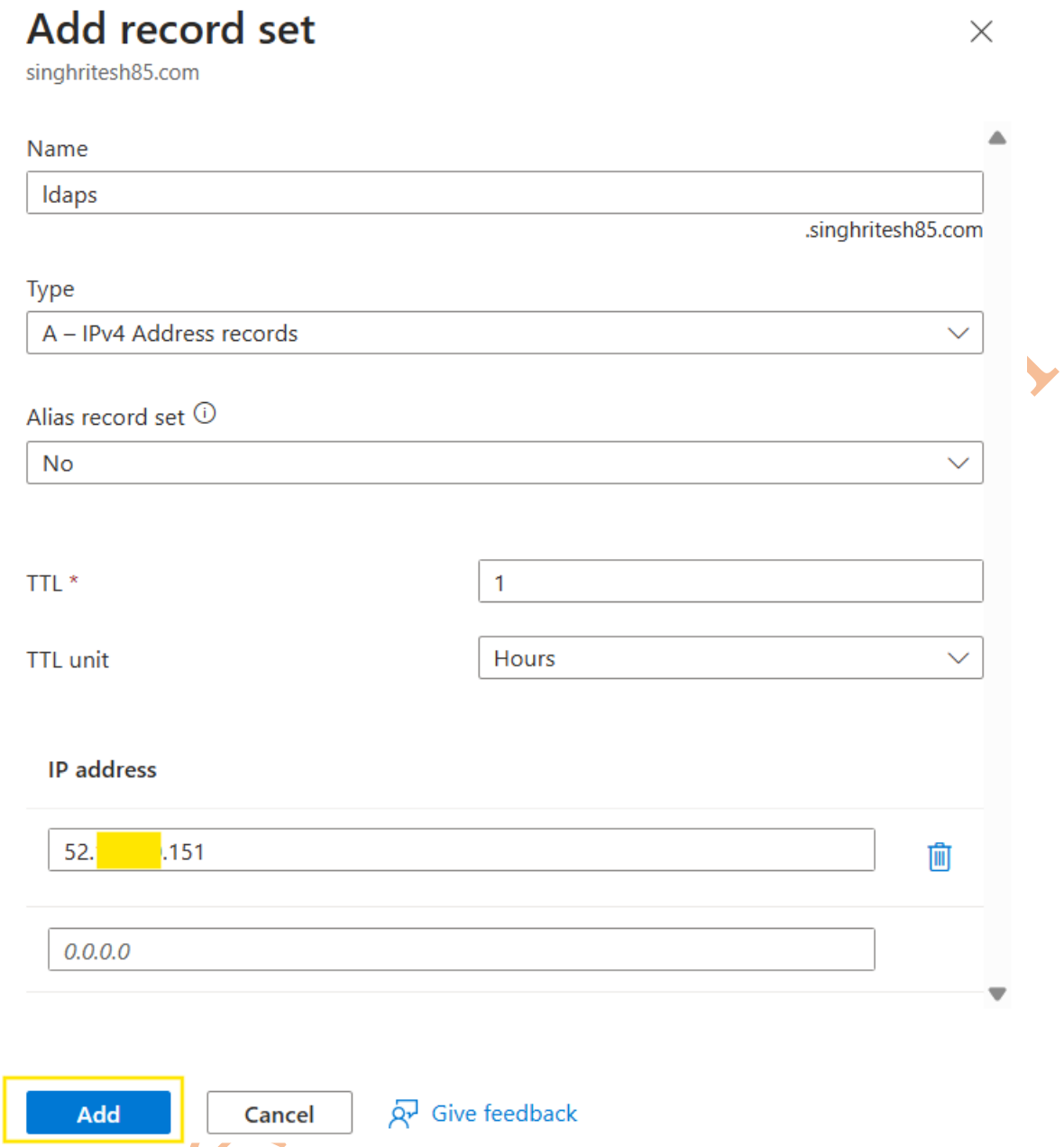
After that user1 will refresh Grafana UI and their Access will be reflected on the UI as well.

Integration of Azure Entra ID with Sonatype Nexus3 Repository

I had integrated Azure Entra ID with Sonatype Nexus3 Repository using the Secure LDAP authentication. To achieve this, I had to use the Microsoft Entra Domain Services (formerly known as Azure Active Directory Domain Services) which I had created earlier using the terraform script.

First this you should do here is go to Azure Microsoft Entra Domain Services > Properties and note down the LDAP external IP address and do the entry in Azure DNS Zone to create the Record Set of A Type as shown in the screenshot attached below.

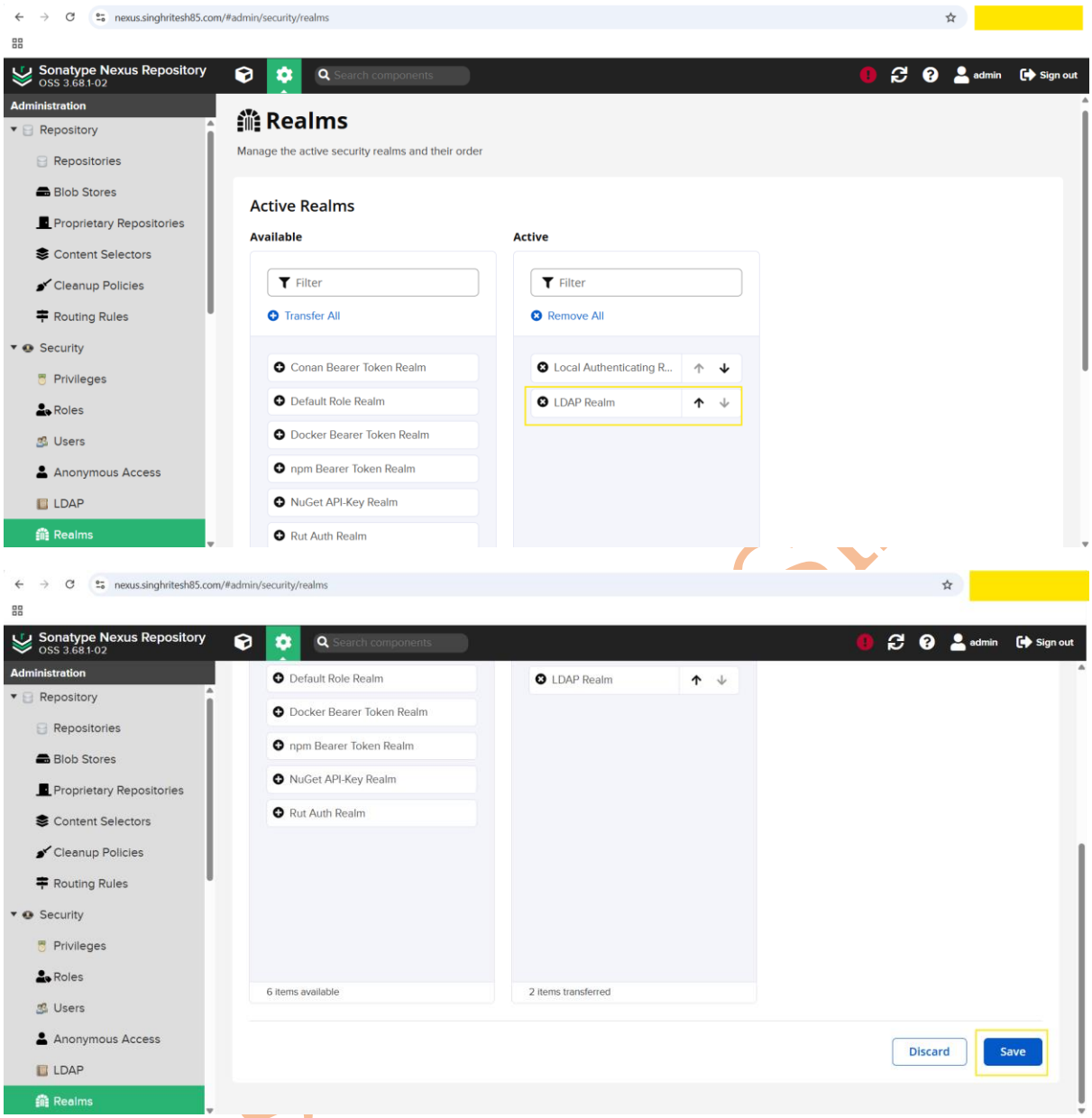

Home > dexter-domainservices
dexter-domainservices | Properties
Microsoft Entra Domain Services
Search
Overview
Activity log
Access control (IAM)
Tags
Resource visualizer
Settings
Properties
Secure LDAP
Synchronization
Custom attributes
Replica sets
Trusts
Health
Notification settings
SKU
Security settings
DNS domain name
singhritesh85.com
Locations
East US
Virtual Networks/Subnets
East US/bankapp-vnet/bankapp-vnet/default
Network security groups
East US/domain-services-nsg
IP addresses
East US/192.168.1.5 192.168.1.4
Secure LDAP
Enabled
Secure LDAP external IP addresses
East US/
Synchronization
All
Admin group
AAD DC Administrators
Resource ID



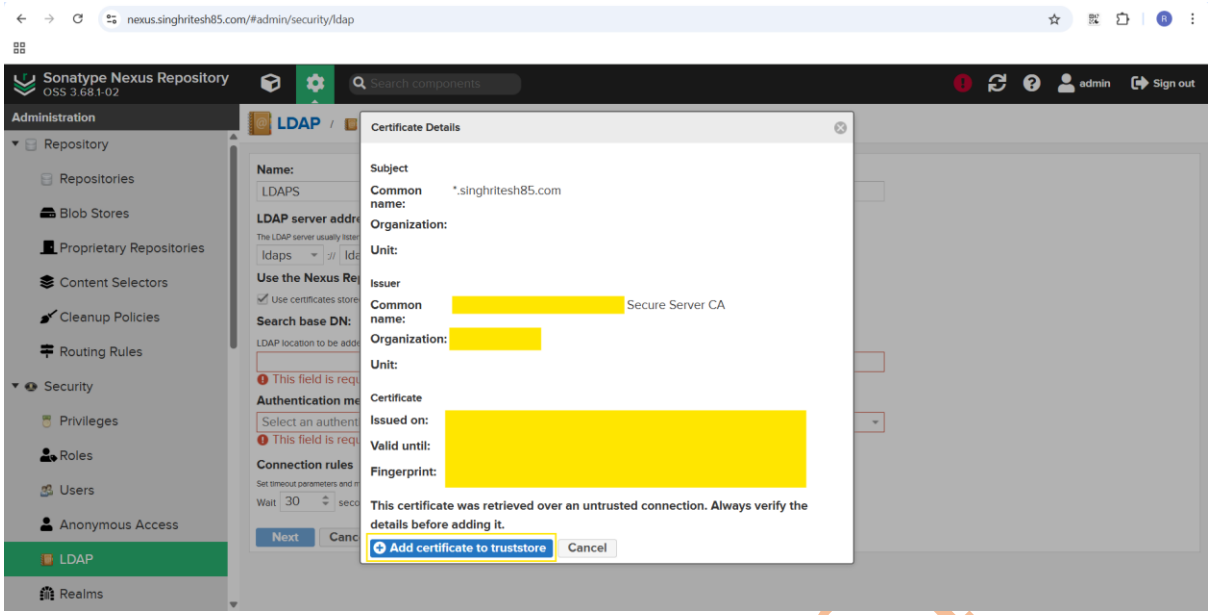
I had updated the **/etc/resolv.conf** file present on **Nexus-Server** and used the Google's Public DNS Server as shown in the screenshot attached below.

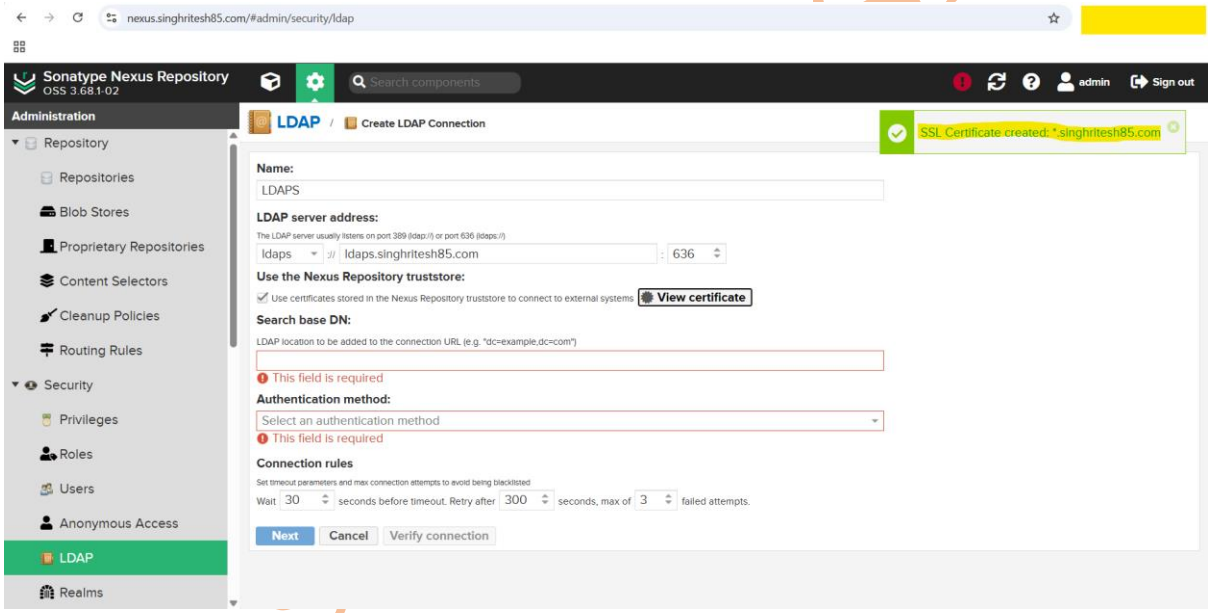
```
[root@                    ~]# cat /etc/resolv.conf
; generated by /usr/sbin/dhclient-script
search us-east-2.compute.internal
options timeout:2 attempts:5
nameserver 8.8.8.8   #10.10.0.2
```

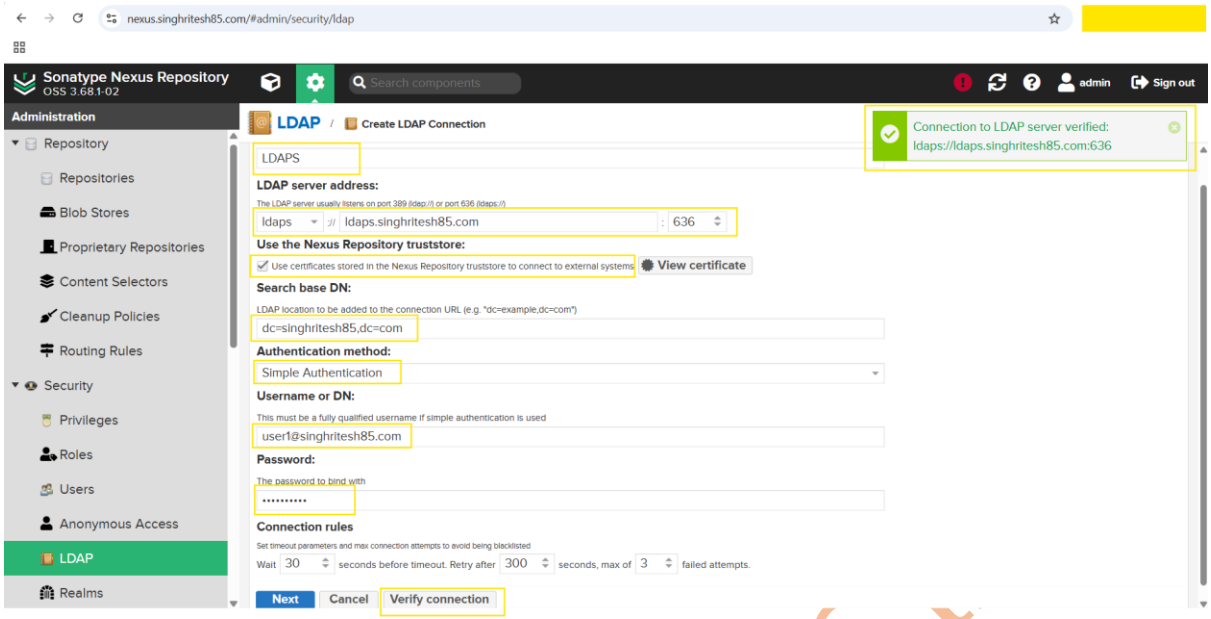
Login into the Nexus and go to **Administration > Security > Realms** and make the LDAP Realm from Available to Active as shown in the screenshot attached below.

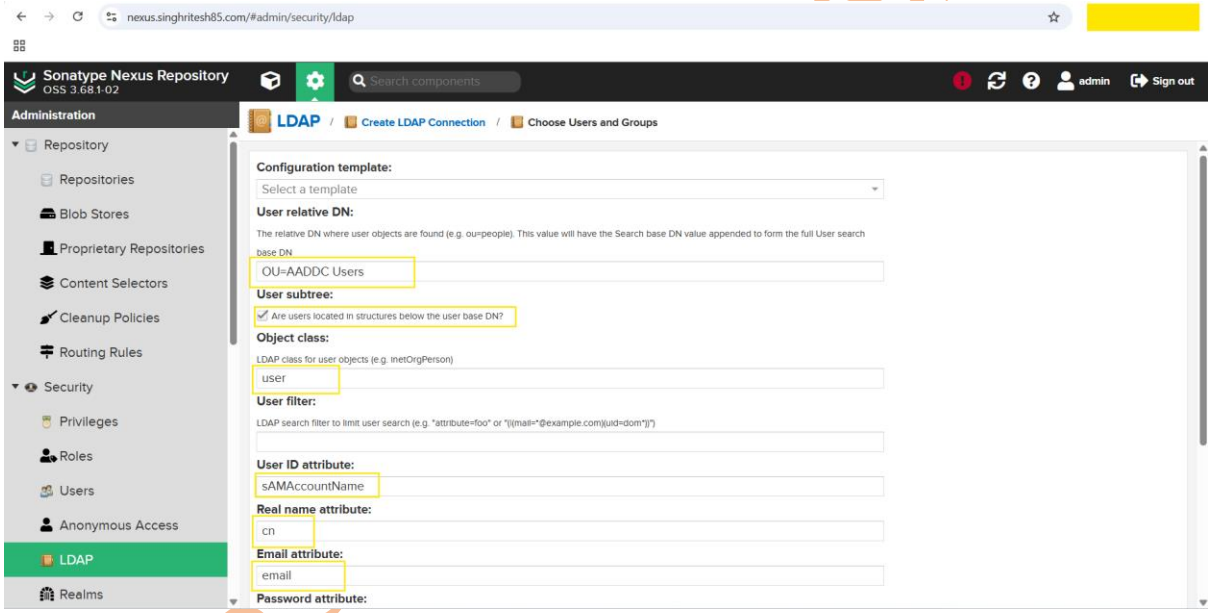


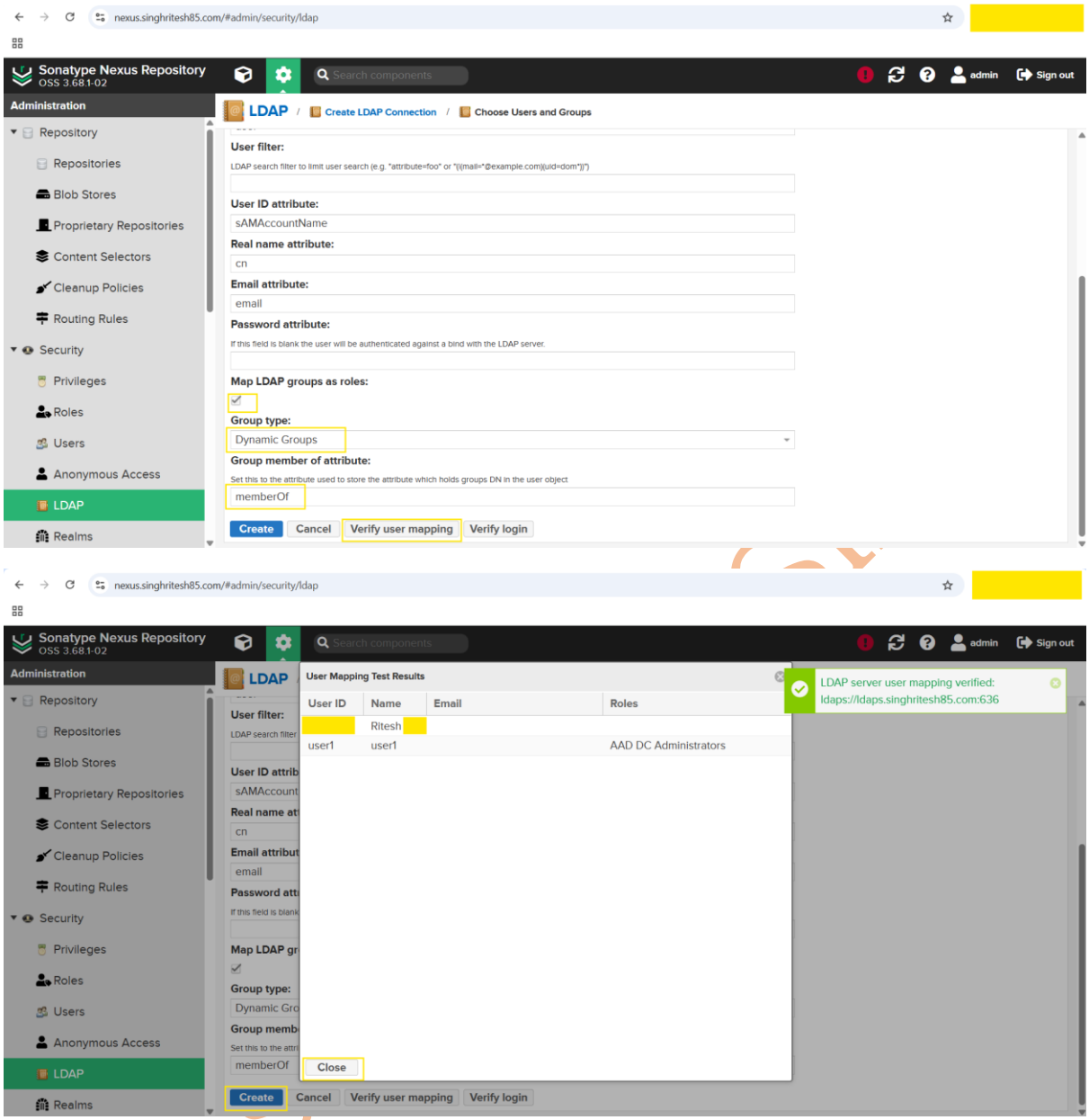
Now go to the **Administration > Security > LDAP** and then **Create Connection**. While creating the connection add the certificate to the truststore as shown in the screenshot attached below.


nexus.singhritesh85.com/#admin/security/ldap
Sonatype Nexus Repository
OSS 3.68.1-02
admin
Sign out
Administration
Repository
Repositories
Blob Stores
Proprietary Repositories
Content Selectors
Cleanup Policies
Routing Rules
Security
Privileges
Roles
Users
Anonymous Access
LDAP
Realms
LDAP
Certificate Details
Subject
Common name: *.singhritesh85.com
Organization:
Unit:
Issuer
Common name: Secure Server CA
Organization:
Unit:
Certificate
Issued on:
Valid until:
Fingerprint:
This certificate was retrieved over an untrusted connection. Always verify the details before adding it.
Add certificate to truststore
Cancel

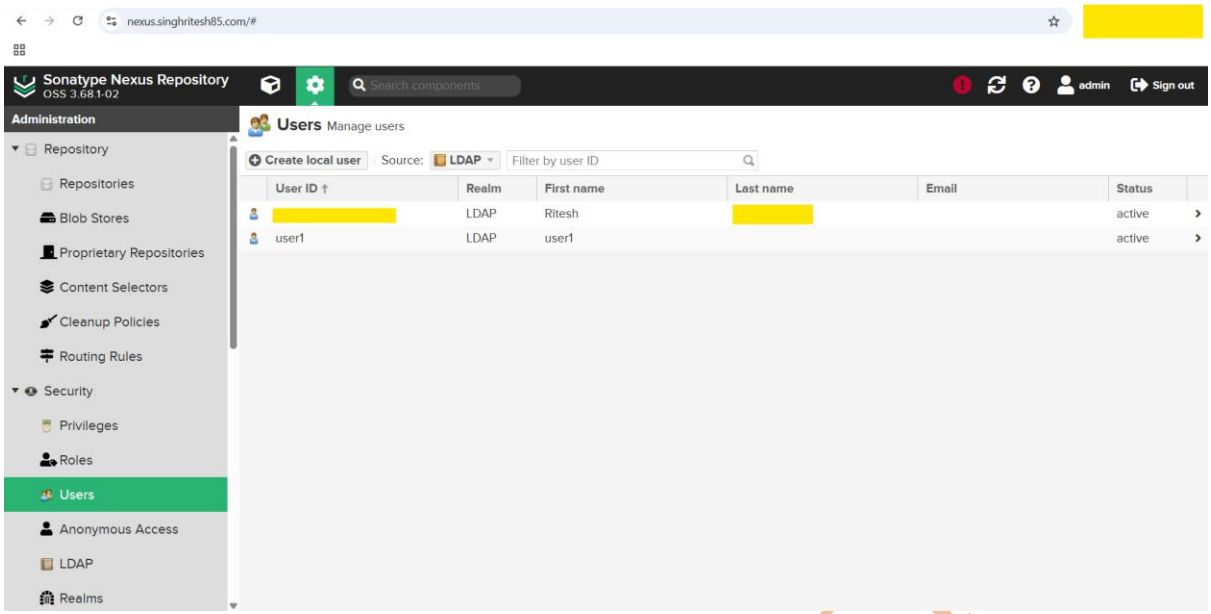

nexus.singhritesh85.com/#admin/security/ldap
Sonatype Nexus Repository
OSS 3.68.1-02
admin
Sign out
Administration
Repository
Repositories
Blob Stores
Proprietary Repositories
Content Selectors
Cleanup Policies
Routing Rules
Security
Privileges
Roles
Users
Anonymous Access
LDAP
Realms
LDAP
Create LDAP Connection
SSL Certificate created: *.singhritesh85.com
Name:
LDAPS
LDAP server address:
ldaps
ldaps.singhritesh85.com
636
Use the Nexus Repository truststore:
Use certificates stored in the Nexus Repository truststore to connect to external systems
View certificate
Search base DN:
This field is required
Authentication method:
Select an authentication method
This field is required
Connection rules
Wait 30 seconds before timeout. Retry after 300 seconds, max of 3 failed attempts.
Next
Cancel
Verify connection


nexus.singhritesh85.com/#admin/security/ldap
Sonatype Nexus Repository
OSS 3.68.1-02
admin
Sign out
Administration
Repository
Repositories
Blob Stores
Proprietary Repositories
Content Selectors
Cleanup Policies
Routing Rules
Security
Privileges
Roles
Users
Anonymous Access
LDAP
Realms
LDAP / Create LDAP Connection
Connection to LDAP server verified:
ldaps://ldaps.singhritesh85.com:636
LDAPS
LDAP server address:
ldaps
ldaps.singhritesh85.com
636
Use the Nexus Repository truststore:
Use certificates stored in the Nexus Repository truststore to connect to external systems
View certificate
Search base DN:
dc=singhritesh85,dc=com
Authentication method:
Simple Authentication
Username or DN:
user1@singhritesh85.com
Password:
Connection rules
Wait 30 seconds before timeout. Retry after 300 seconds, max of 3 failed attempts.
Next
Cancel
Verify connection

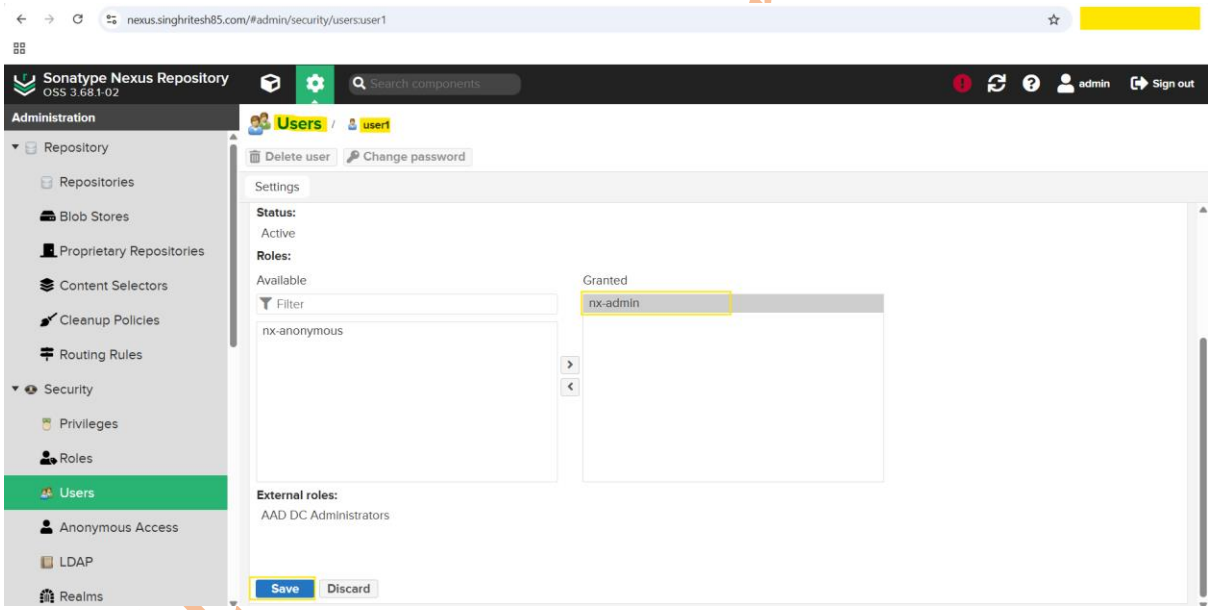

nexus.singhritesh85.com/#admin/security/ldap
Sonatype Nexus Repository
OSS 3.68.1-02
admin
Sign out
Administration
Repository
Repositories
Blob Stores
Proprietary Repositories
Content Selectors
Cleanup Policies
Routing Rules
Security
Privileges
Roles
Users
Anonymous Access
LDAP
Realms
LDAP / Create LDAP Connection / Choose Users and Groups
Configuration template:
Select a template
User relative DN:
OU=AADDC Users
User subtree:
Are users located in structures below the user base DN?
Object class:
user
User filter:
User ID attribute:
sAMAccountName
Real name attribute:
cn
Email attribute:
email
Password attribute:



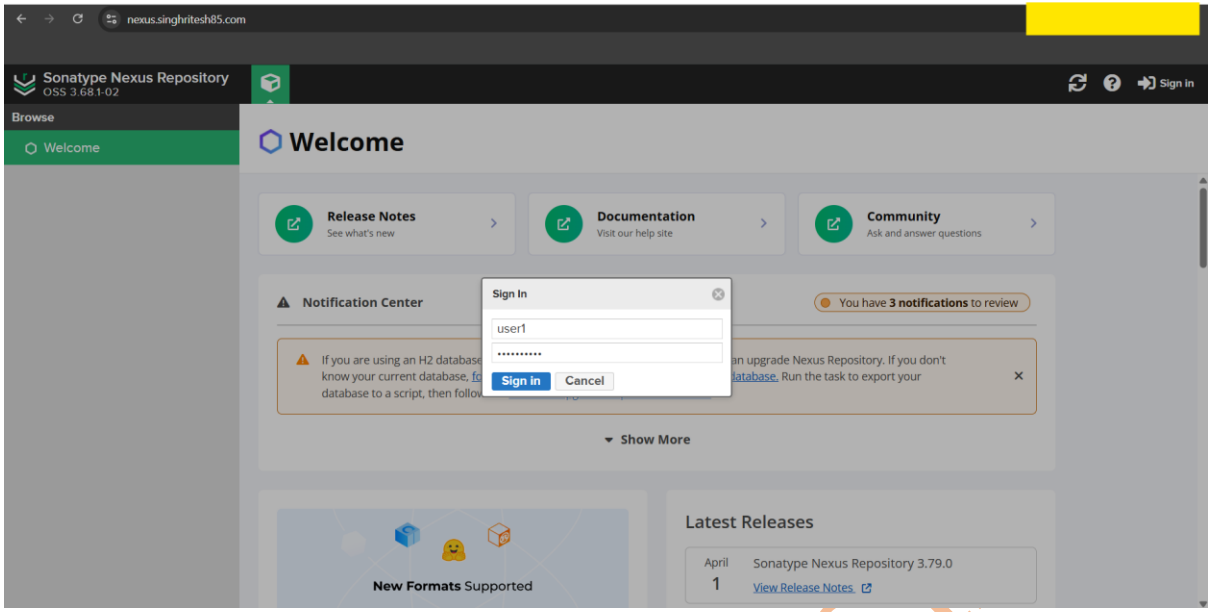
Now go to Administration > Users and Source: LDAP then you will see your Azure Entra ID Users will be listed here as shown in the screenshot attached below.

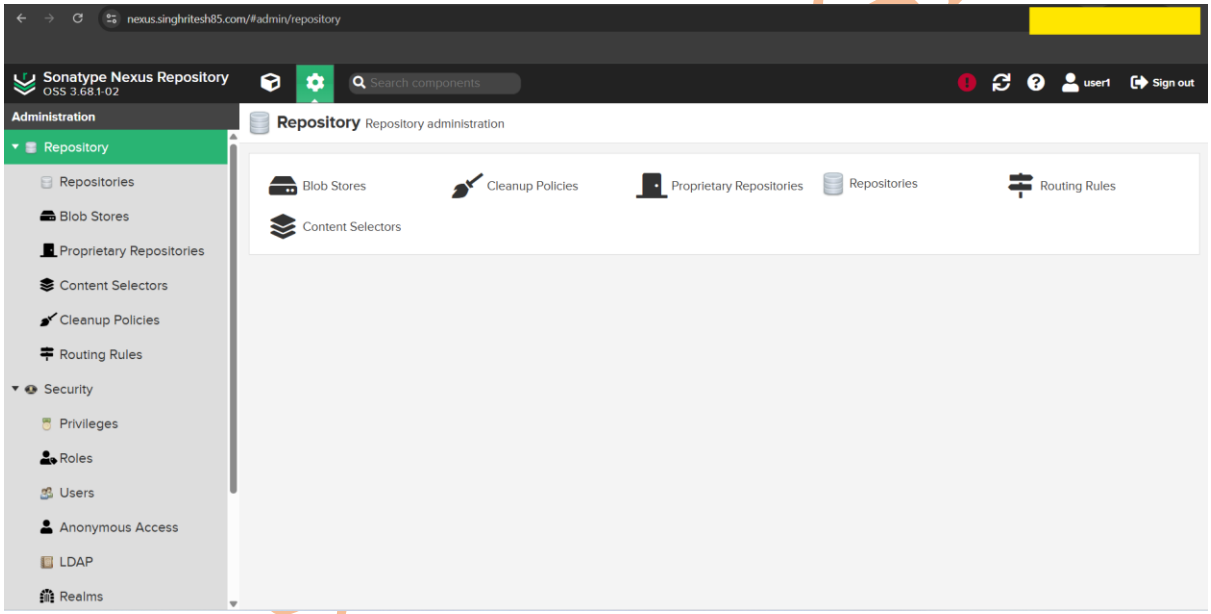


Then I logged-in as user1 and the Nexus Administrator will provide the sufficient privileges to user1. In this project I provide user1 as Administrator privilege as shown in the screenshot attached below.

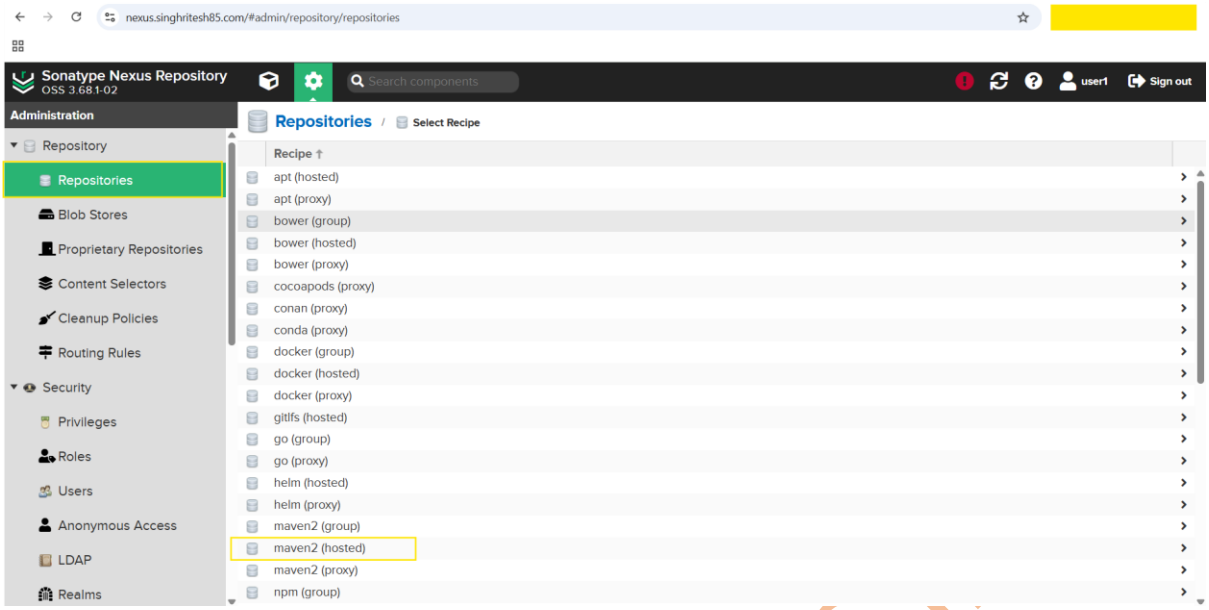


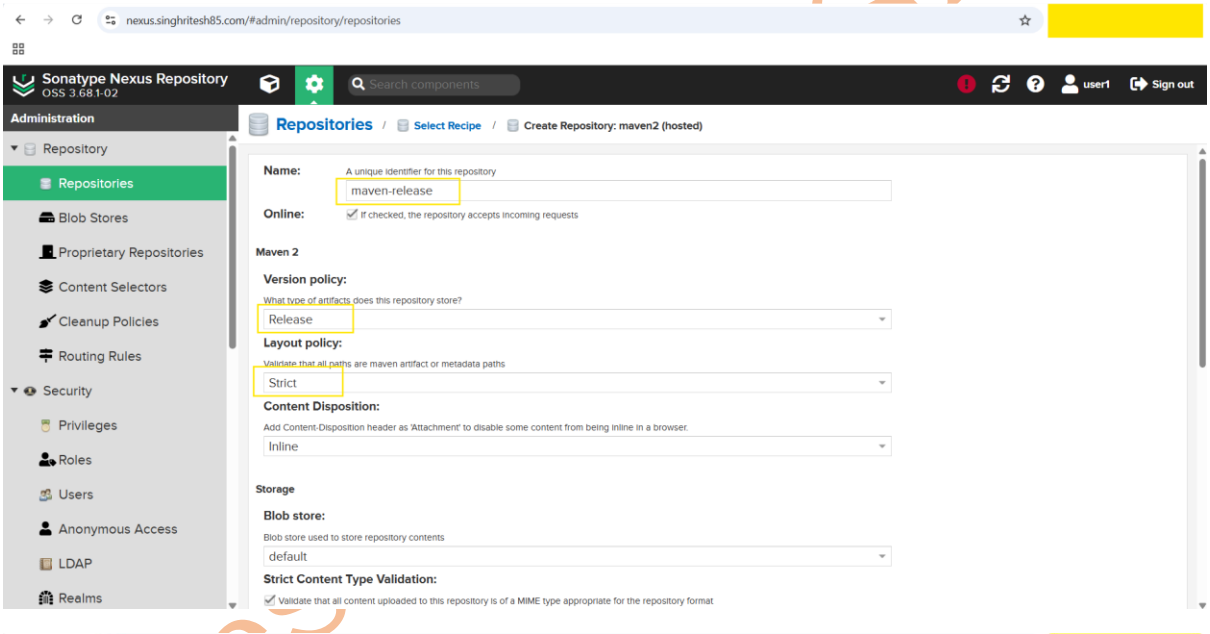
Finally, I logged-in as user1 as shown in the screenshot attached below and found that user got the access as provided above.

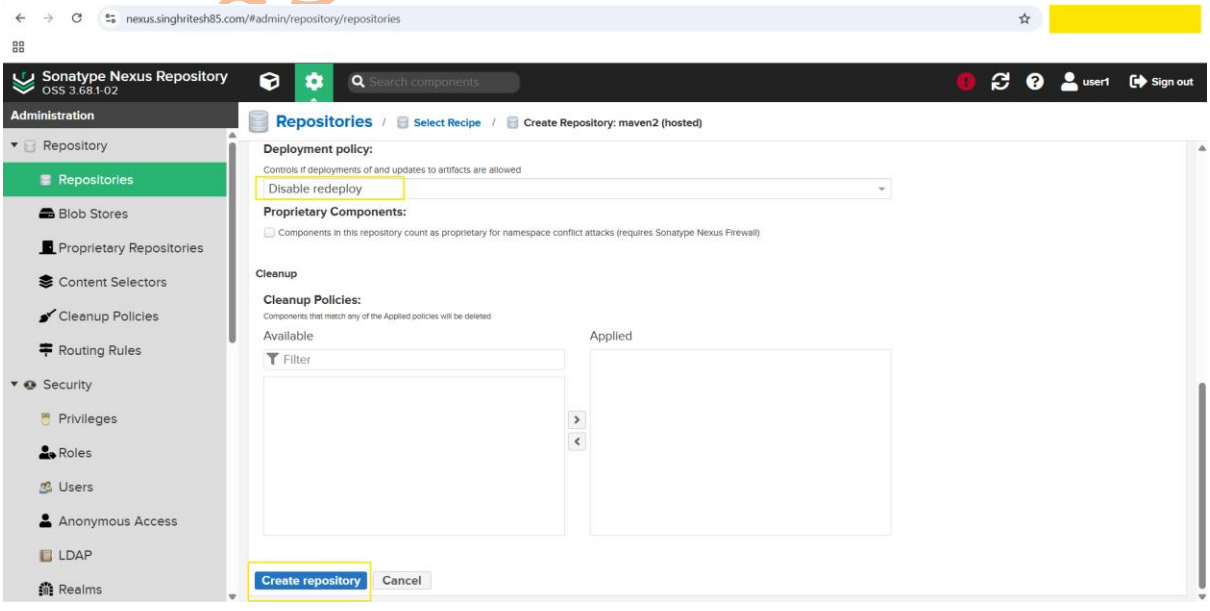


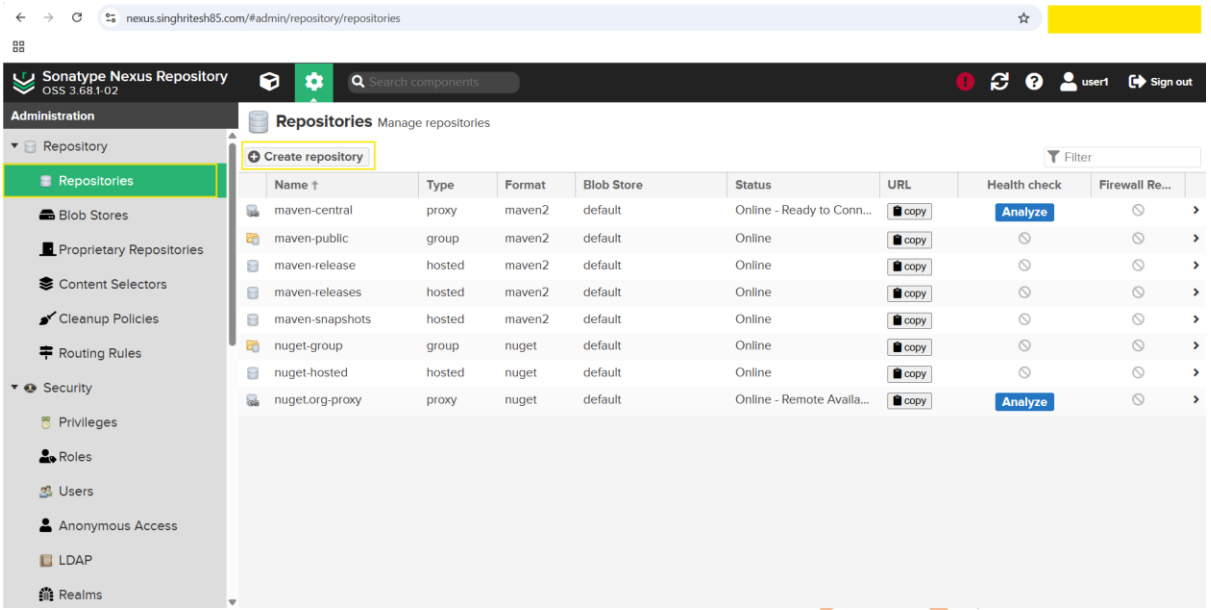


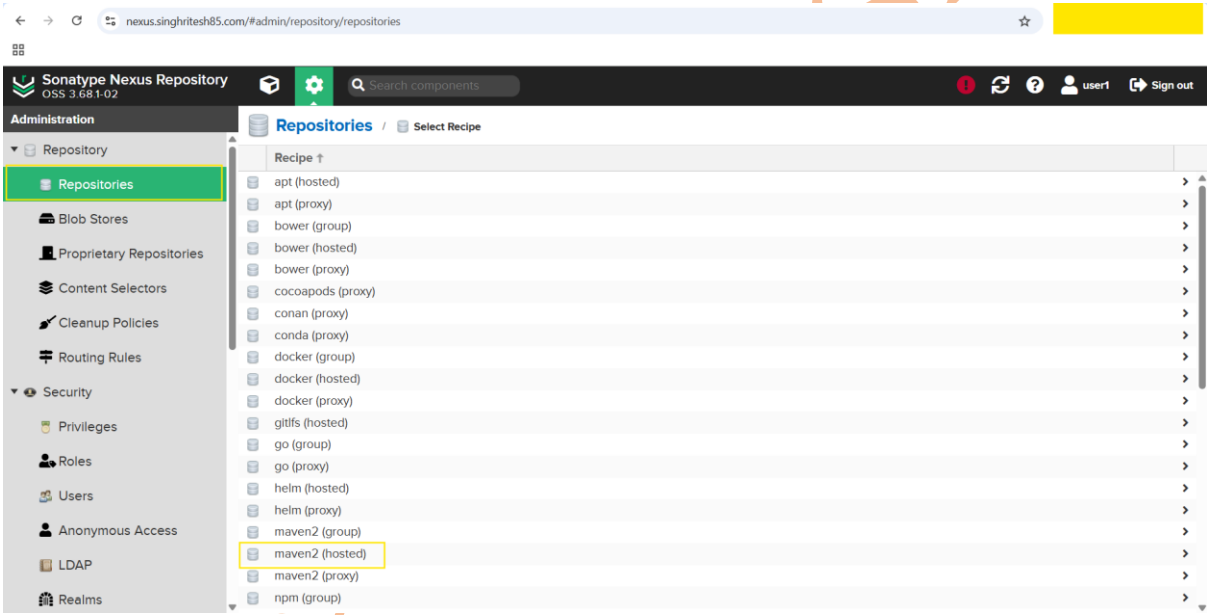
For this project in Nexus, I created two repositories named as **maven-release** and **maven-snapshot** as shown in the screenshot attached below.

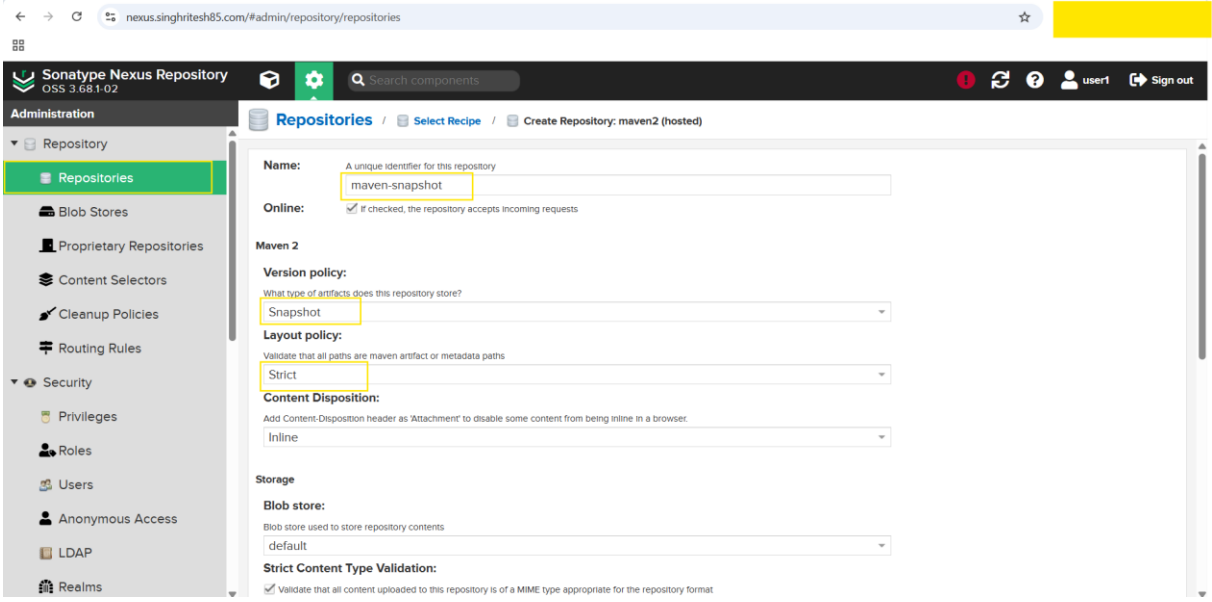

nexus.singhritesh85.com/#admin/repository/repositories
Sonatype Nexus Repository
OSS 3.68.1-02
Search components
user1
Sign out
Administration
Repository
Repositories
Blob Stores
Proprietary Repositories
Content Selectors
Cleanup Policies
Routing Rules
Security
Privileges
Roles
Users
Anonymous Access
LDAP
Realms
Repositories / Select Recipe
Recipe
apt (hosted)
apt (proxy)
bower (group)
bower (hosted)
bower (proxy)
cocoapods (proxy)
conan (proxy)
conda (proxy)
docker (group)
docker (hosted)
docker (proxy)
gitlfs (hosted)
go (group)
go (proxy)
helm (hosted)
helm (proxy)
maven2 (group)
maven2 (hosted)
maven2 (proxy)
npm (group)

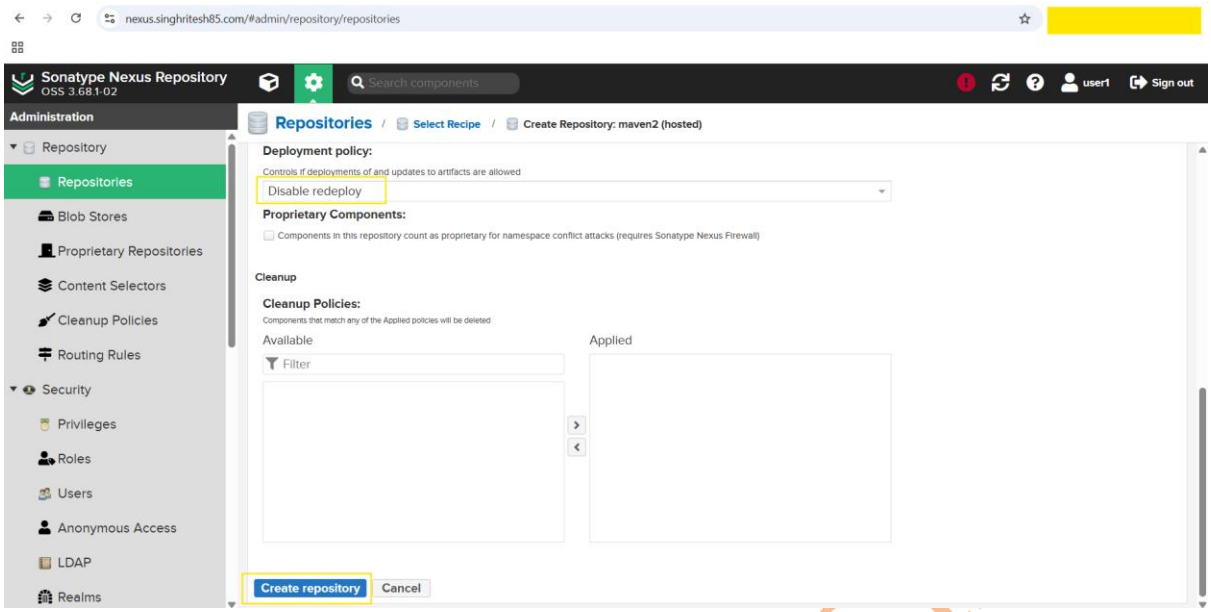

nexus.singhritesh85.com/#admin/repository/repositories
Repositories / Select Recipe / Create Repository: maven2 (hosted)
Name:
A unique identifier for this repository
maven-release
Online:
If checked, the repository accepts incoming requests
Maven 2
Version policy:
What type of artifacts does this repository store?
Release
Layout policy:
Validate that all paths are maven artifact or metadata paths
Strict
Content Disposition:
Add Content-Disposition header as 'Attachment' to disable some content from being inline in a browser.
Inline
Storage
Blob store:
Blob store used to store repository contents
default
Strict Content Type Validation:
Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format


nexus.singhritesh85.com/#admin/repository/repositories
Repositories / Select Recipe / Create Repository: maven2 (hosted)
Deployment policy:
Controls if deployments of and updates to artifacts are allowed
Disable redeploy
Proprietary Components:
Components in this repository count as proprietary for namespace conflict attacks (requires Sonatype Nexus Firewall)
Cleanup
Cleanup Policies:
Components that match any of the Applied policies will be deleted
Available
Filter
Applied
Create repository
Cancel

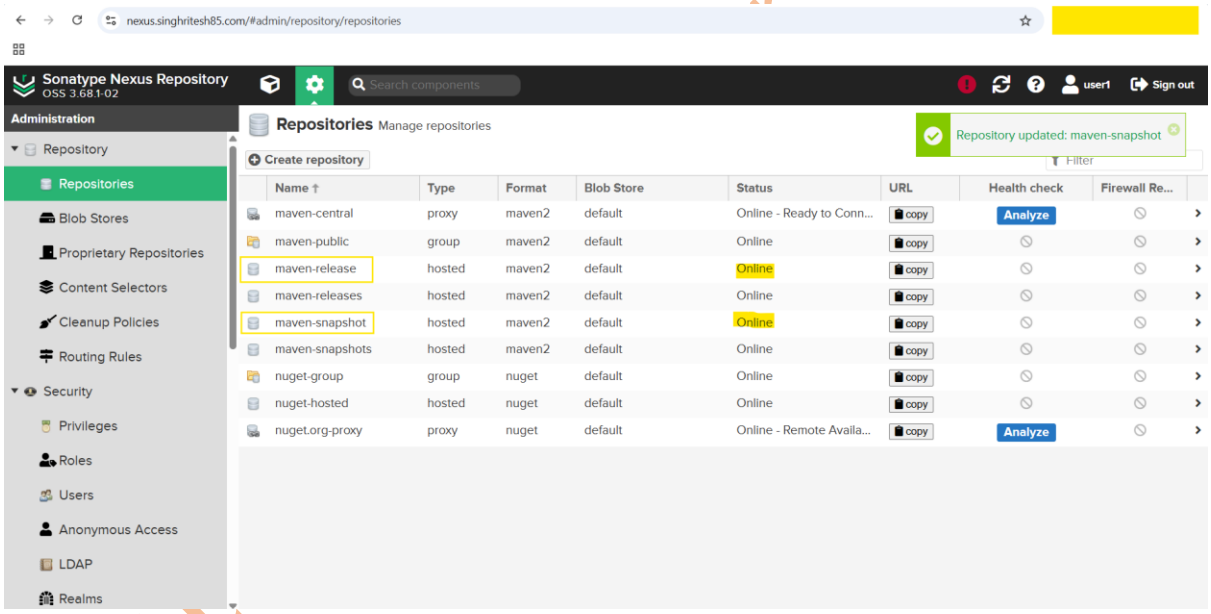

nexus.singhritesh85.com/#admin/repository/repositories
Sonatype Nexus Repository
OSS 3.68.1-02
Search components
user1
Sign out
Administration
Repository
Repositories
Blob Stores
Proprietary Repositories
Content Selectors
Cleanup Policies
Routing Rules
Security
Privileges
Roles
Users
Anonymous Access
LDAP
Realms
Repositories Manage repositories
Create repository
Filter
Name
Type
Format
Blob Store
Status
URL
Health check
Firewall Re...
maven-central proxy maven2 default Online - Ready to Conn... copy Analyze
maven-public group maven2 default Online copy
maven-release hosted maven2 default Online copy
maven-releases hosted maven2 default Online copy
maven-snapshots hosted maven2 default Online copy
nuget-group group nuget default Online copy
nuget-hosted hosted nuget default Online copy
nuget.org-proxy proxy nuget default Online - Remote Availa... copy Analyze


nexus.singhritesh85.com/#admin/repository/repositories
Sonatype Nexus Repository
OSS 3.68.1-02
Search components
user1
Sign out
Administration
Repository
Repositories
Blob Stores
Proprietary Repositories
Content Selectors
Cleanup Policies
Routing Rules
Security
Privileges
Roles
Users
Anonymous Access
LDAP
Realms
Repositories / Select Recipe
Recipe
apt (hosted)
apt (proxy)
bower (group)
bower (hosted)
bower (proxy)
cocoapods (proxy)
conan (proxy)
conda (proxy)
docker (group)
docker (hosted)
docker (proxy)
gitlfs (hosted)
go (group)
go (proxy)
helm (hosted)
helm (proxy)
maven2 (group)
maven2 (hosted)
maven2 (proxy)
npm (group)


nexus.singhritesh85.com/#admin/repository/repositories
Sonatype Nexus Repository
OSS 3.68.1-02
Search components
user1
Sign out
Administration
Repository
Repositories
Blob Stores
Proprietary Repositories
Content Selectors
Cleanup Policies
Routing Rules
Security
Privileges
Roles
Users
Anonymous Access
LDAP
Realms
Repositories / Select Recipe / Create Repository: maven2 (hosted)
Name:
A unique identifier for this repository
maven-snapshot
Online:
If checked, the repository accepts incoming requests
Maven 2
Version policy:
What type of artifacts does this repository store?
Snapshot
Layout policy:
Validate that all paths are maven artifact or metadata paths
Strict
Content Disposition:
Add Content-Disposition header as 'Attachment' to disable some content from being inline in a browser.
Inline
Storage
Blob store:
Blob store used to store repository contents
default
Strict Content Type Validation:
Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format



Finally, the two repositories in Nexus named as **maven-release** and **maven-snapshot** had been created as shown in the screenshot attached below.



Installation of Nginx Ingress Controller in EKS Cluster

To install Nginx ingress controller in EKS Cluster use the Command as provided below and use you own SSL Certificate ARN instead of arn:aws:acm:us-east-2:02XXXXXXXXX6:certificate/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX.

kubectl create ns ingress-nginx
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo update

helm install ingress-nginx ingress-nginx/ingress-nginx -n ingress-nginx --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-ssl-cert"=arn:aws:acm:us-east-2:02XXXXXXXXX6:certificate/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-connection-idle-timeout"="60" --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-cross-zone-load-balancing-enabled"="true" --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-type"="elb" --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-backend-protocol"=http --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-ssl-ports"=https --set controller.service.targetPorts.https=http --set-string controller.config.use-forwarded-headers="true"

```
[root@                    ~]# kubectl create ns ingress-nginx
namespace/ingress-nginx created
```

```
[root@                    ~]# helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
"ingress-nginx" has been added to your repositories
[root@                    ~]# helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "ingress-nginx" chart repository
Update Complete. ⎈Happy Helming!⎈
```

```
[root@                    ~]# helm install ingress-nginx ingress-nginx/ingress-nginx -n ingress-nginx --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-ssl-cert"=arn:aws:acm:us-east-2:02        6:certificate/                                    --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-connection-idle-timeout"="60" --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-cross-zone-load-balancing-enabled"="true" --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-type"="elb" --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-backend-protocol"=http --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-ssl-ports"=https --set controller.service.targetPorts.https=http --set-string controller.config.use-forwarded-headers="true"
```

```
[root@                    ~]# kubectl get pods -n ingress-nginx
NAME                                        READY   STATUS    RESTARTS   AGE
ingress-nginx-controller-                   1/1     Running   0          98s
[root@                    ~]# kubectl get svc -n ingress-nginx
NAME                                 TYPE           CLUSTER-IP       EXTERNAL-IP                                                              PORT(S)
          AGE
ingress-nginx-controller             LoadBalancer   172.      49     a                                        .us-east-2.elb.amazonaws.com   80:30009/TCP,443
:32743/TCP   116s
ingress-nginx-controller-admission   ClusterIP      172.      .196   <none>                                                                   443/TCP
          116s
```

Installation of ArgoCD in EKS Cluster

In this project for EKS Cluster deployment I used ArgoCD and the ArgoCD CLI, I installed the ArgoCD as shown in the screenshot attached below.

kubectl create namespace argocd

kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml

```
[root@                ~]# kubectl get nodes
NAME                                          STATUS   ROLES    AGE    VERSION
ip-            .us-east-2.compute.internal   Ready    <none>          v1.30.4-eks-a      9
ip-            .us-east-2.compute.internal   Ready    <none>          v1.30.4-eks-a      9
[root@                ~]# kubectl create namespace argocd
namespace/argocd created
[root@                ~]# kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
```

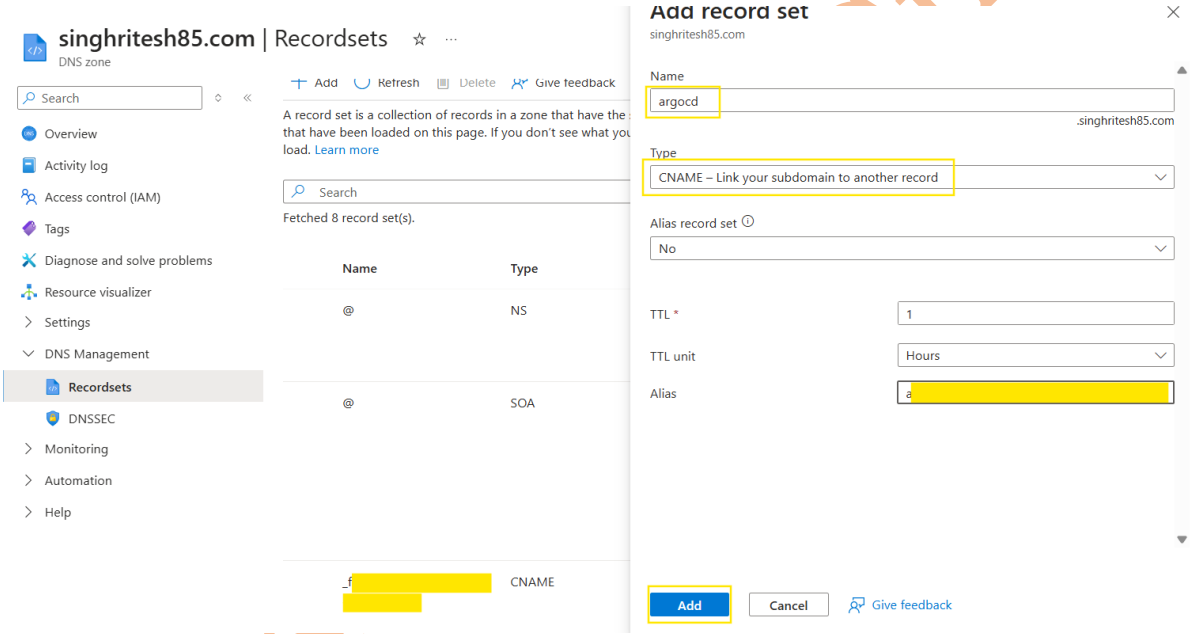
The ingress rule for ArgoCD is as shown in the screenshot attached below.

Ritesh Kumar Singh || Email Address: - riteshkumarsingh9559@gmail.com **|| LinkedIn: -** https://www.linkedin.com/in/ritesh-kumar-singh-41113128b/ **|| GitHub: -** https://github.com/singhritesh85

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: minimal-ingress
  namespace: argocd
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/backend-protocol: "HTTPS"    ###  You can use this option for this particular case for ArgoCD but not for all
#    nginx.ingress.kubernetes.io/ssl-redirect: "false"
spec:
  ingressClassName: nginx
  rules:
  - host: argocd.singhritesh85.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: argocd-server   ### Provide your service Name
            port:
              number: 80    #### Provide your service port for this particular example you can also choose 443
```

```
[root@                ~]# cat argocd-ingress-rule.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: minimal-ingress
  namespace: argocd
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/backend-protocol: "HTTPS"    ###  You can use this option for this particular case for ArgoCD but not for all
#    nginx.ingress.kubernetes.io/ssl-redirect: "false"
spec:
  ingressClassName: nginx
  rules:
  - host: argocd.singhritesh85.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: argocd-server   ### Provide your service Name
            port:
              number: 80     #### Provide your service port for this particular example you can also choose 443
[root@                ~]# kubectl apply -f argocd-ingress-rule.yaml
ingress.networking.k8s.io/minimal-ingress created
[root@                ~]# kubectl get ing -A
NAMESPACE   NAME              CLASS   HOSTS                   ADDRESS                                                                  PORTS   AGE
argocd      minimal-ingress   nginx   argocd.singhritesh85.com   a                                        .us-east-2.elb.amazonaws.com   80      8s
```

The entry for DNS Name corresponding to HOST **argocd.singhritesh85.com** in Azure DNS Zone to create the record set is as shown in the screenshot attached below.

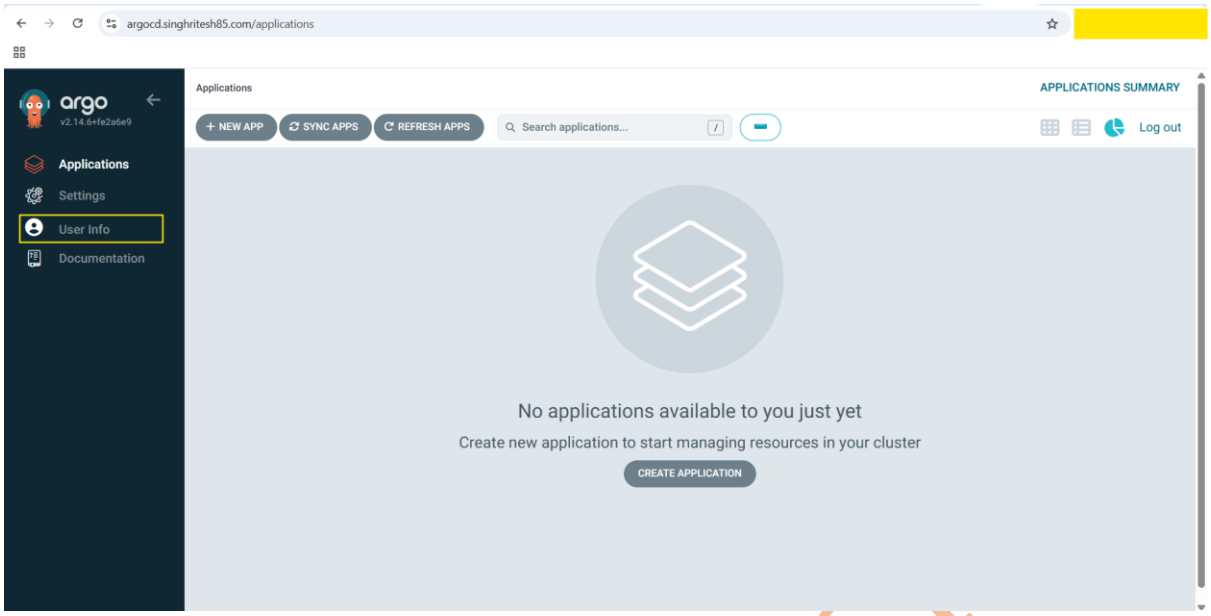


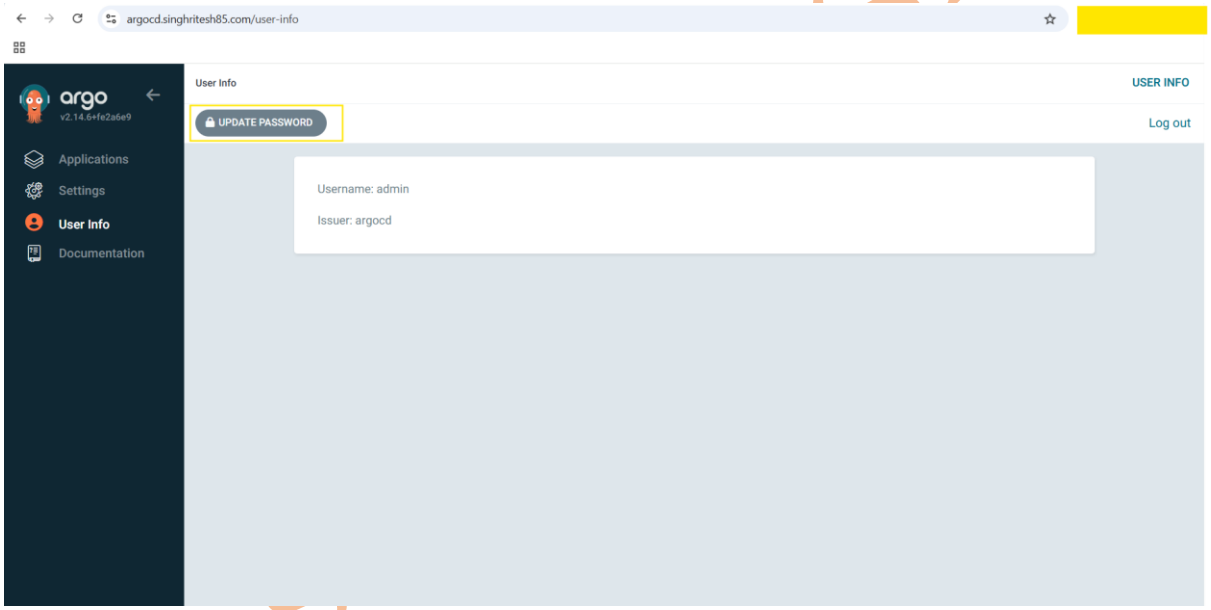
I generated the password for ArgoCD is as shown in the screenshot attached below.

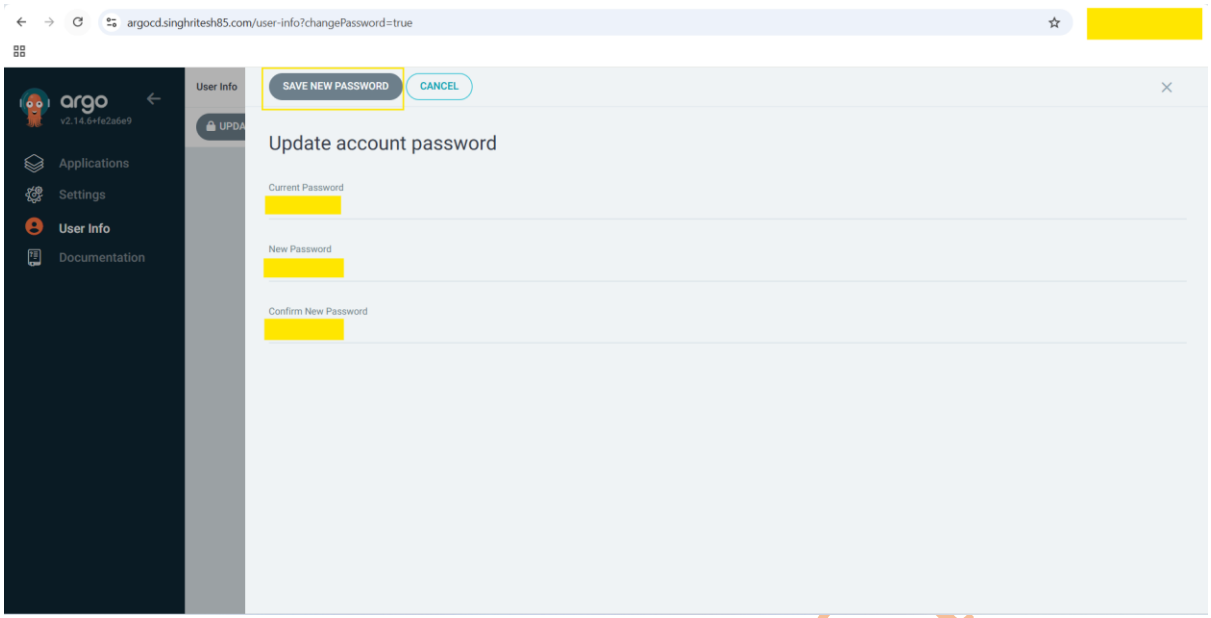
kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath='{.data.password}' | base64 -d

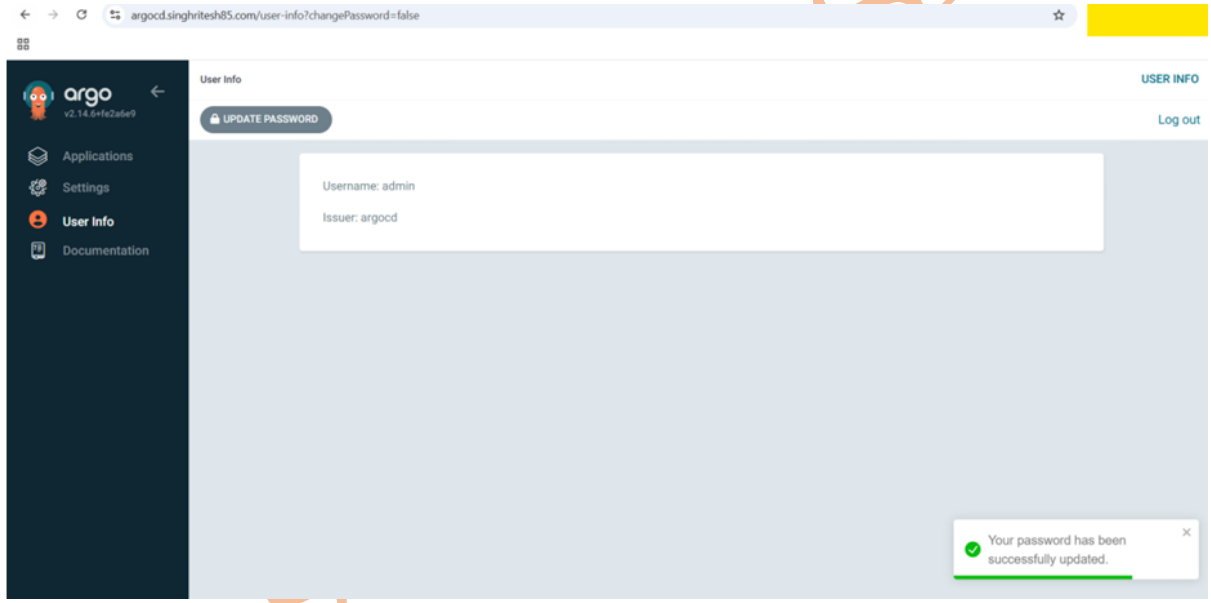
```
[root@                ~]# kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath='{.data.password}' | base64 -d
e                A[root@                ~]#
```

After login into the ArgoCD for the first time I updated the ArgoCD Password as shown in the screenshot attached below.


argocd.singhritesh85.com/applications
argo
v2.14.6+fe2a6e9
Applications
Settings
User Info
Documentation
Applications
APPLICATIONS SUMMARY
+ NEW APP
SYNC APPS
REFRESH APPS
Search applications...
Log out
No applications available to you just yet
Create new application to start managing resources in your cluster
CREATE APPLICATION


argocd.singhritesh85.com/user-info
argo
v2.14.6+fe2a6e9
Applications
Settings
User Info
Documentation
User Info
USER INFO
UPDATE PASSWORD
Log out
Username: admin
Issuer: argocd





Installation of ArgoCD CLI using the commands as shown in the screenshot attached below.

curl -sSL -o argocd-linux-amd64 https://github.com/argoproj/argo-cd/releases/latest/download/argocd-linux-amd64

sudo install -m 555 argocd-linux-amd64 /usr/local/bin/argocd

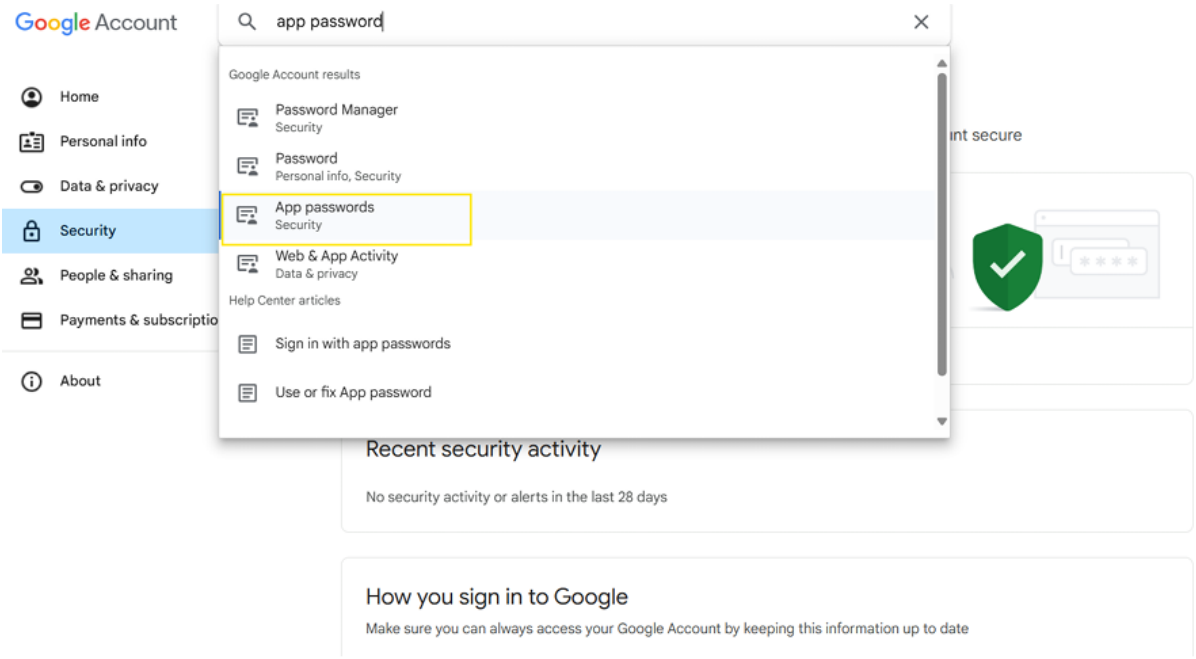
rm argocd-linux-amd64

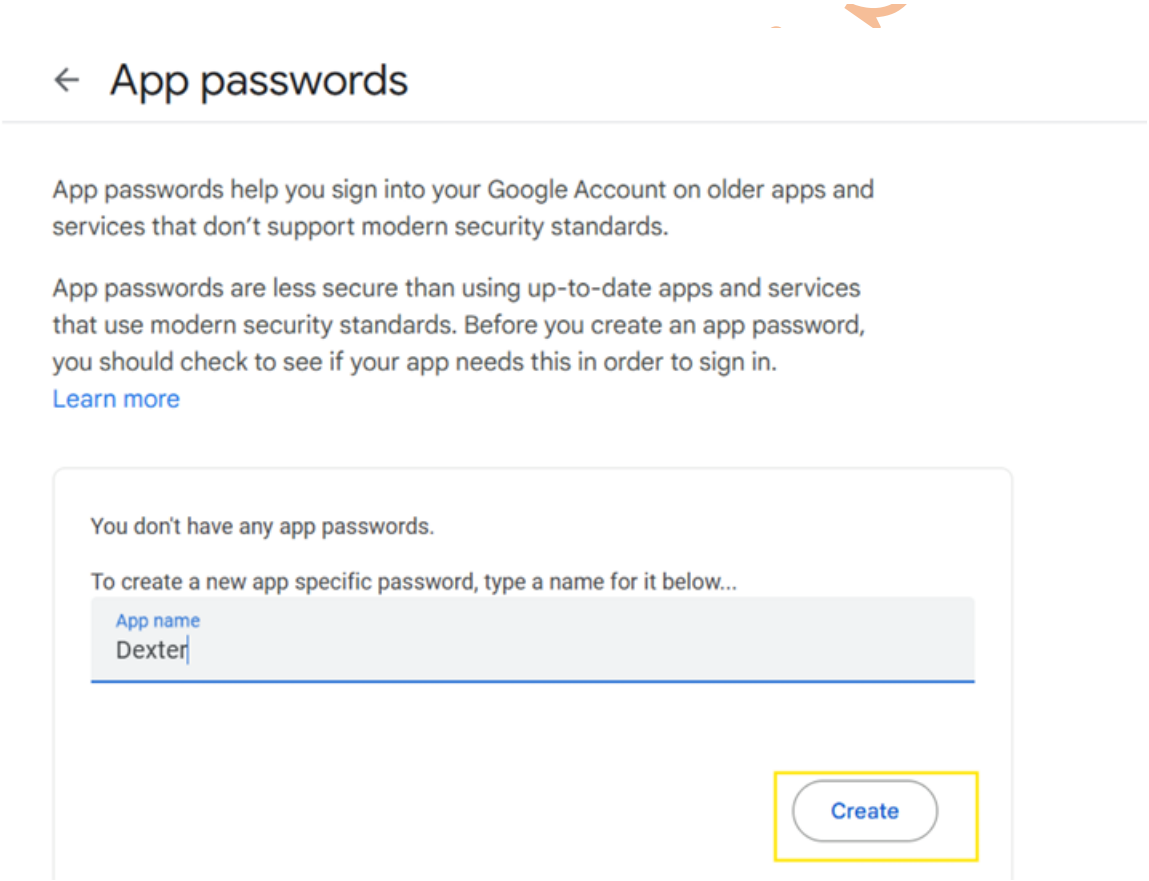
```
[root@          ~]# curl -sSL -o argocd-linux-amd64 https://github.com/argoproj/argo-cd/releases/latest/download/argocd-linux-amd64
[root@          ~]# sudo install -m 555 argocd-linux-amd64 /usr/local/bin/argocd
[root@          ~]# rm argocd-linux-amd64
rm: remove regular file ‘argocd-linux-amd64’? yes
```

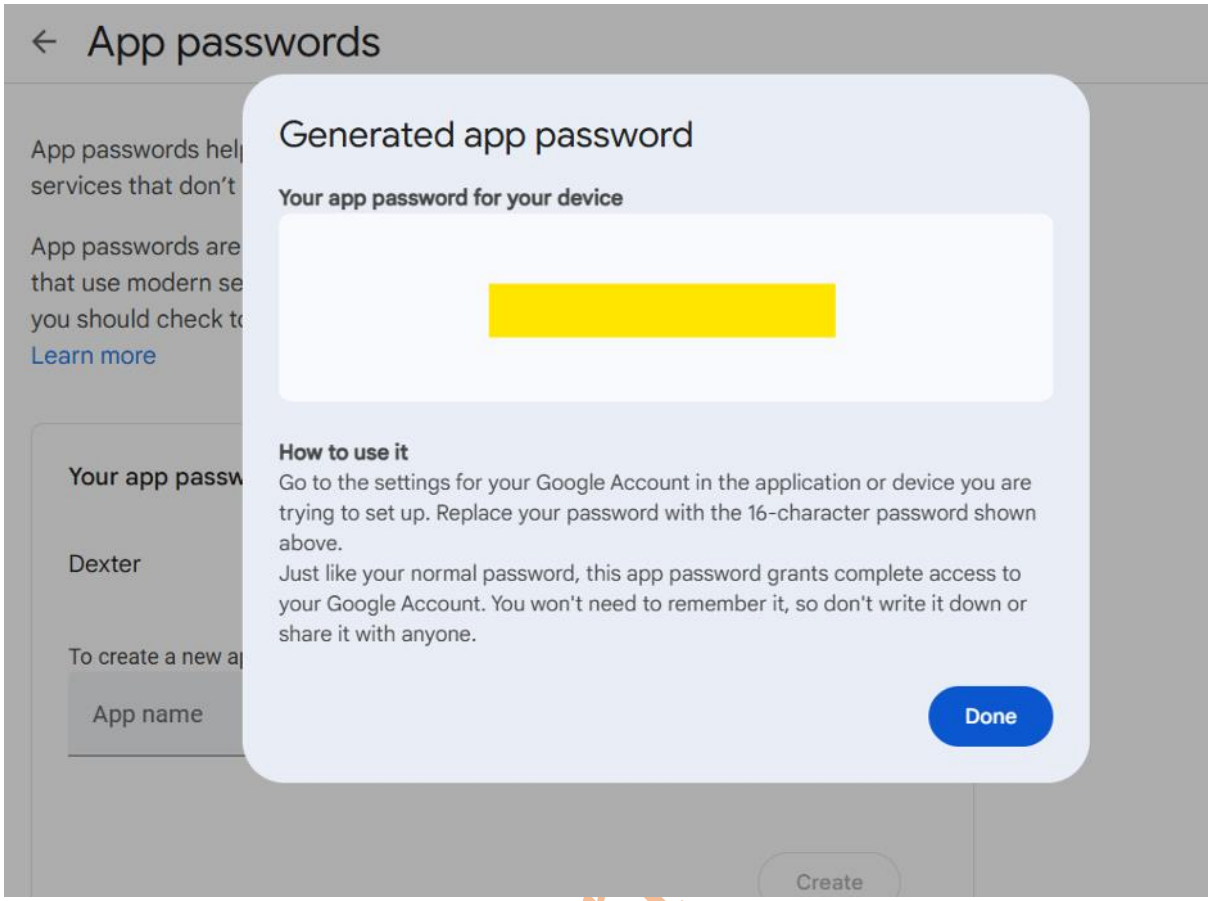
Configuration of Email to Send notification on Group Email-ID using Jenkins and Grafana

To configure Gmail to send notification to group Email ID I should have App Password for my Gmail account as shown in the screenshot attached below.

Go to your **Gmail Account > Manage your Google Account > Security** and then search for **app password** and click on **App Passwords** as shown in the screenshot attached below.


Google Account
app password
Google Account results
Password Manager
Security
Password
Personal info, Security
App passwords
Security
Web & App Activity
Data & privacy
Help Center articles
Sign in with app passwords
Use or fix App password
Home
Personal info
Data & privacy
Security
People & sharing
Payments & subscriptio
About
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

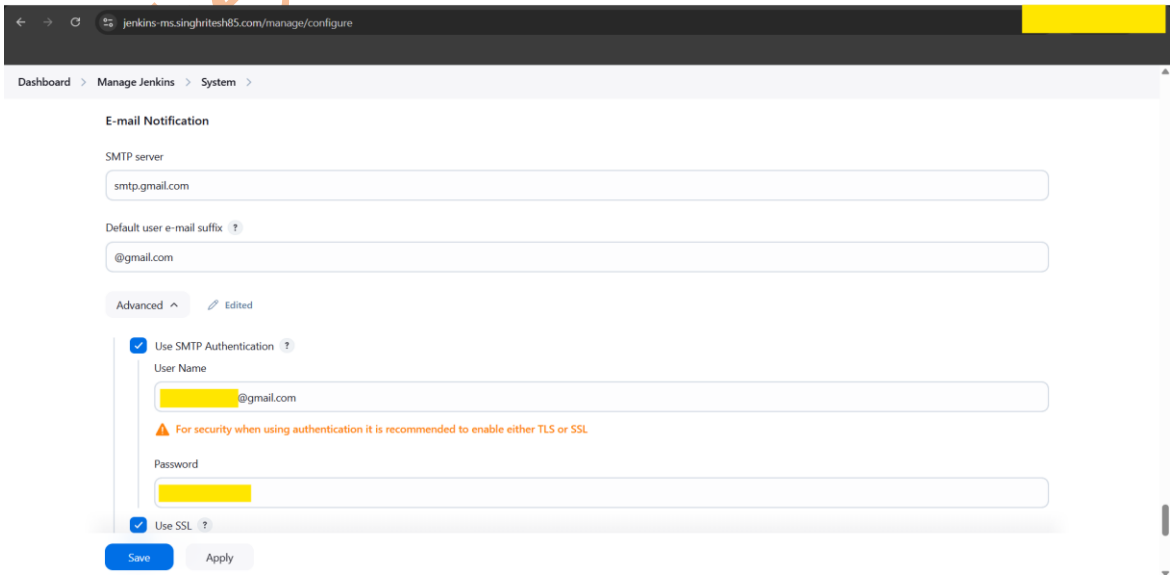

App passwords
App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.
App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to 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
Dexter
Create

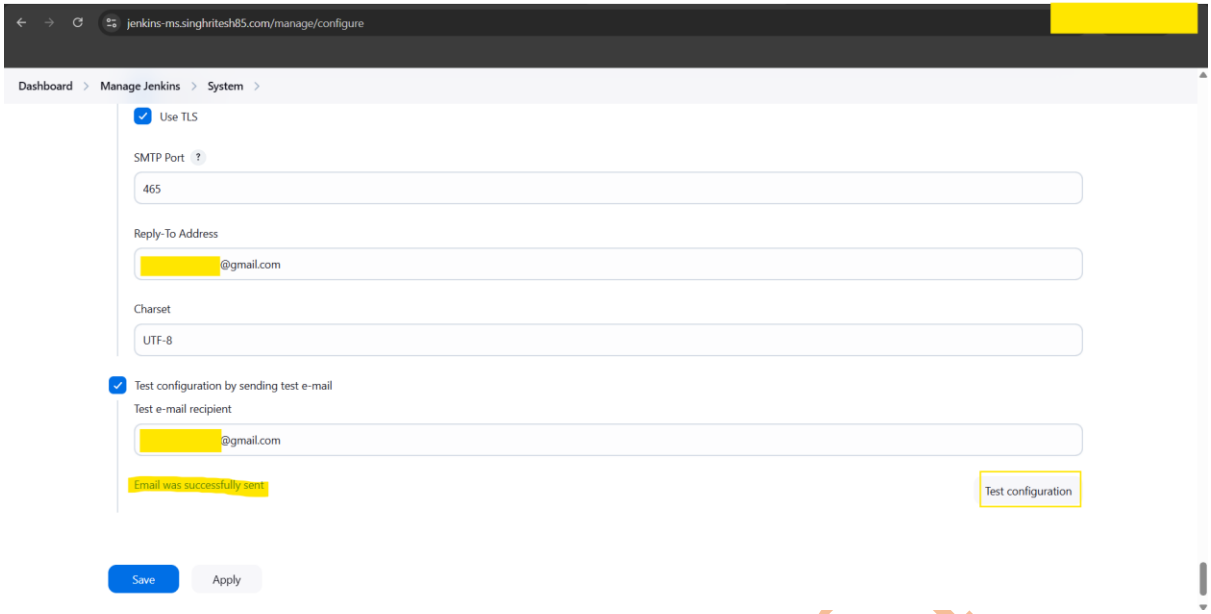


I had deleted this App Password after completion of this project, this App password does not exist anymore. You can use your own Gmail Account's App Passwords.

It is Possible to Send notification to group Email ID using Jenkins and Grafana through Amazon Simple Email Services (Amazon SES). To know more you can refer the project present in my GitHub Repo https://github.com/singhritesh85/DevOps-Project-2tier-WebApp-Deployment.git. However, for this project I used Gmail App Passwords as explained above.

Now, configuration of email to Send notification on Group Email-ID using Jenkins and Grafana is as discussed below.





To configure Alerts in Grafana, first I created contact points with the Email ID and changed smtp settings in the configuration file **/etc/grafana/grafana.ini** of Grafana as shown in the screenshot attached below.

```
################################### SMTP / Emailing ##########################
[smtp]
enabled = true
host = smtp.gmail.com:587
user = @gmail.com
# If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"""
password =
;cert_file =
;key_file =
skip_verify = true
from_address = @gmail.com
from_name = Grafana Alert for BankApp
# EHLO identity in SMTP dialog (defaults to instance_name)
;ehlo_identity = dashboard.example.com
# SMTP startTLS policy (defaults to 'OpportunisticStartTLS')
;startTLS_policy = NoStartTLS
# Enable trace propagation in e-mail headers, using the 'traceparent', 'tracestate' and (optionally) 'baggage' fields (defaults to false)
;enable_tracing = false
```

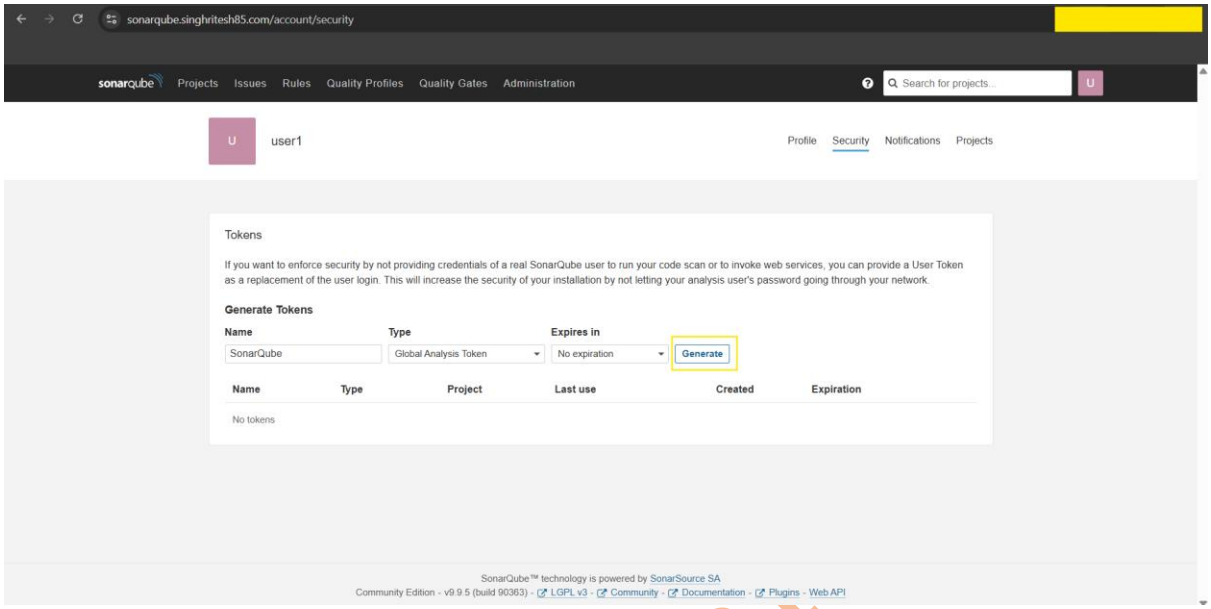
Then restart the **grafana-server** service as shown in the screenshot attached below.

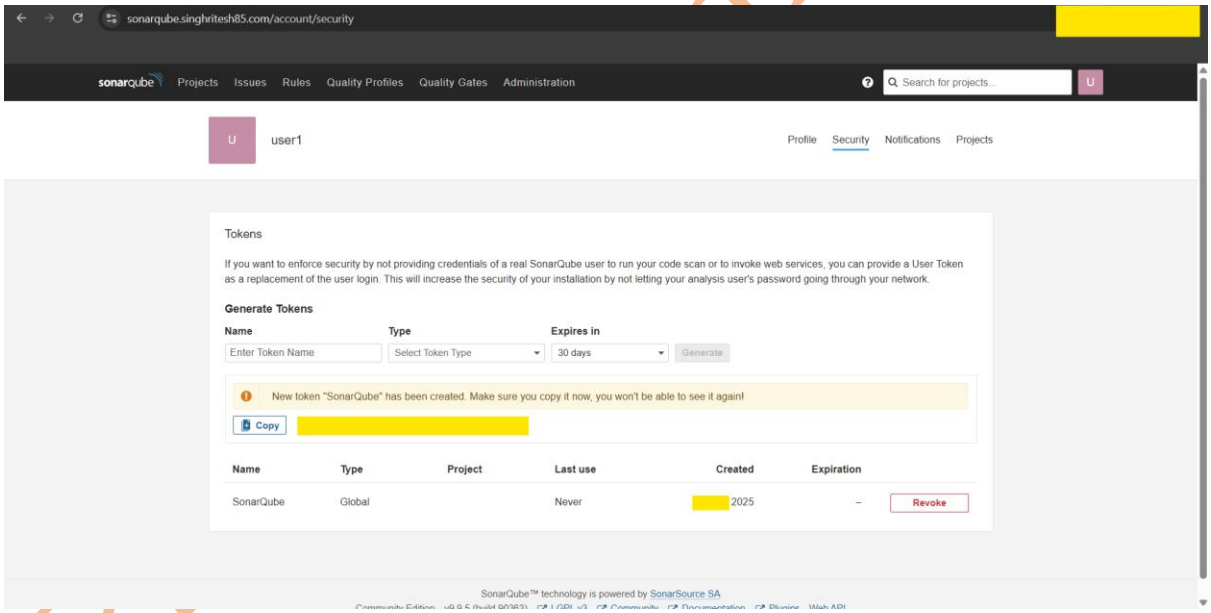
```
[root@ ~]# systemctl restart grafana-server.service
[root@ ~]# systemctl status grafana-server.service
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; vendor preset: disabled)
   Active: active (running) since  2025  UTC; 9s ago
     Docs: http://docs.grafana.org
```

Creation of Alerts in Grafana I will discuss later here I will discuss first integration of Jenkins with SonarQube and Sonatype Nexus3 Repository then will create the Jenkins Job and deploy the Bank Application. To do so I used the Jenkinsfile as shown in the screenshot attached below. This Jenkinsfile is also available in my GitHub Repo https://github.com/singhritesh85/DevOps-Project-BankApplication-BlueGreen-Deployment-MultiCloud.git.

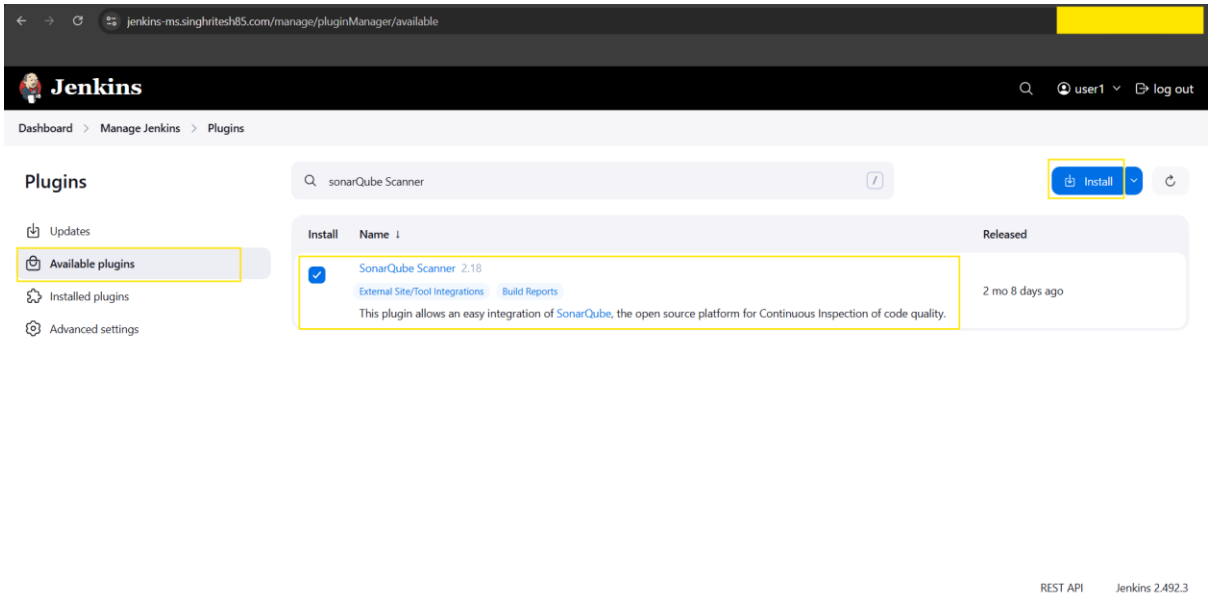
Integration of Jenkins with SonarQube

To Integrate Jenkins with SonarQube I need a Security Token in SonarQube which I created as shown in the screenshot attached below.

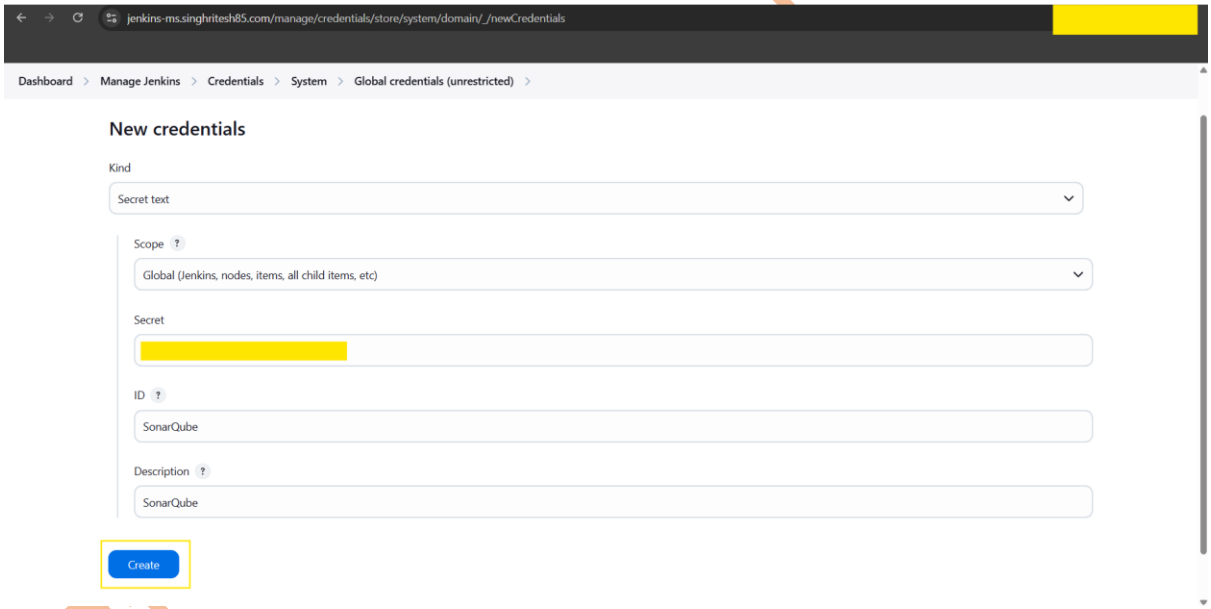




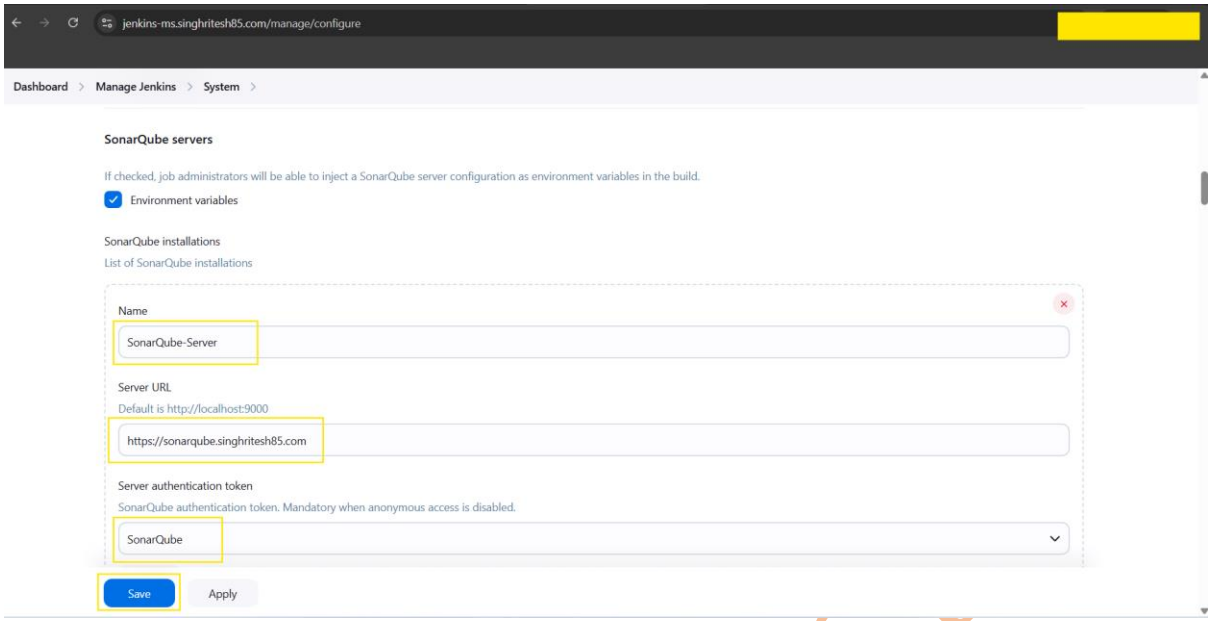
Now go to Jenkins **Manage Jenkins > System > Plugins > Available Plugins** and install the **SonarQube Scanner** Plugin as shown in the screenshot attached below.



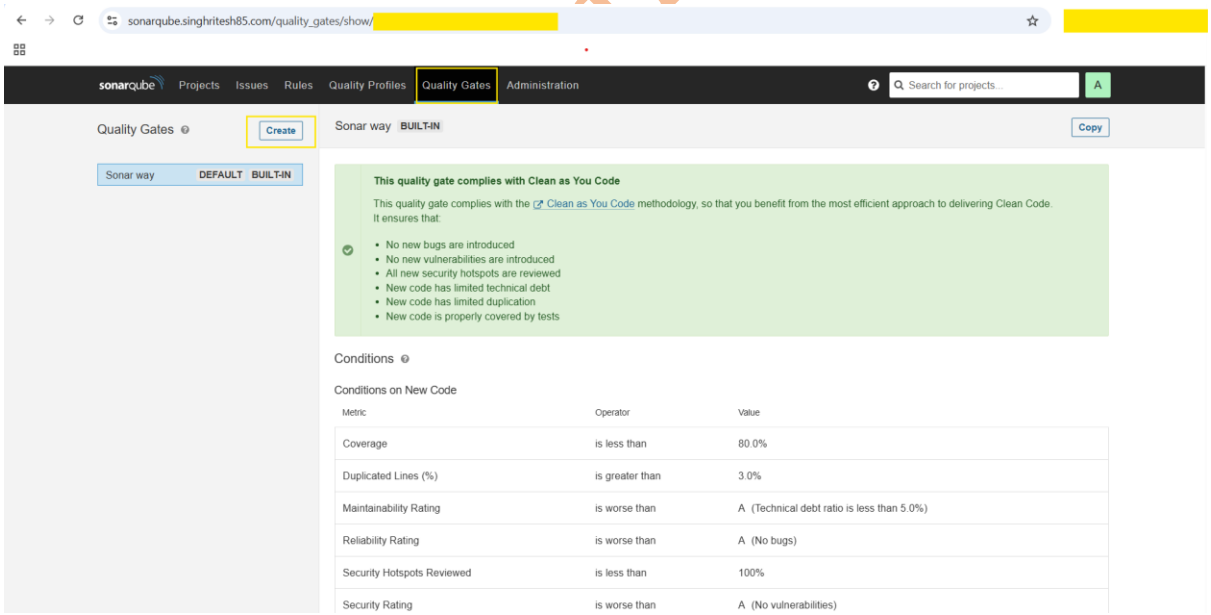
Do not restart the Jenkins after installation of **SonarQube-Scanner** Plugin. Then create a Jenkins Credential of kind Secret text with the SonarQube Secret Token which I created earlier as shown in the screenshot attached below.

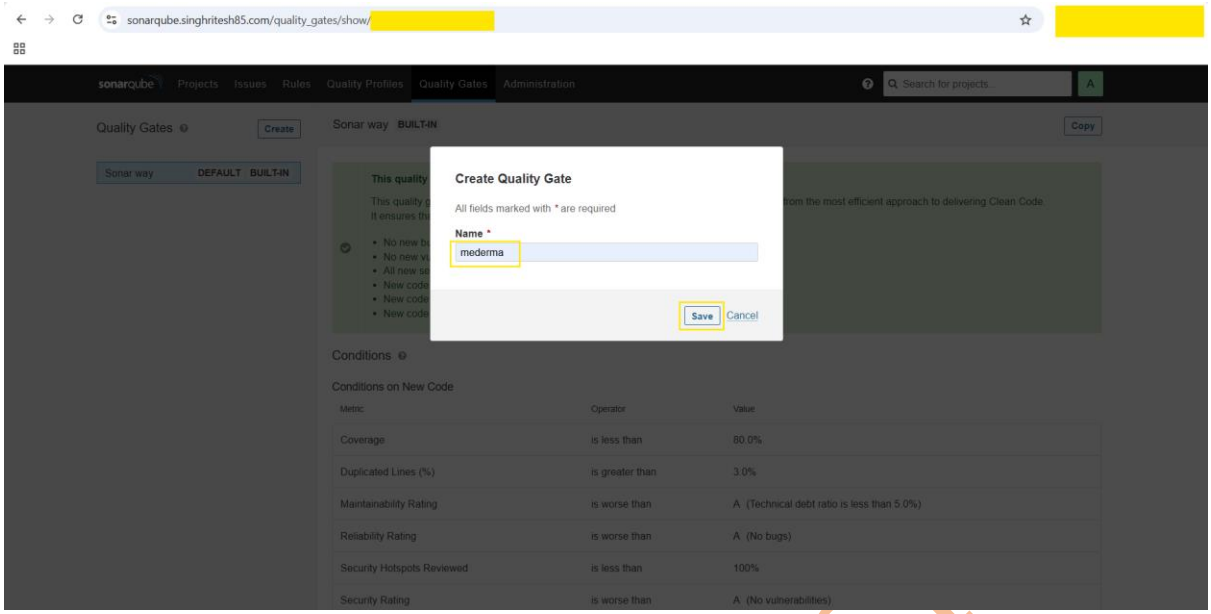


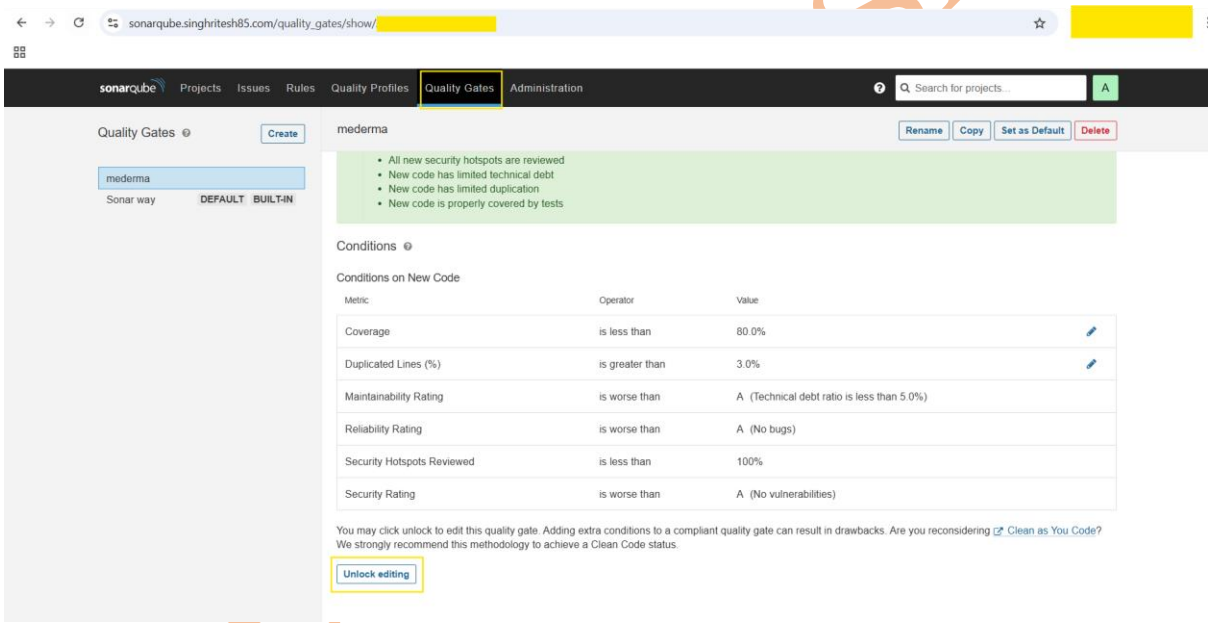
Now go to Jenkins **Manage Jenkins > System** and search for SonarQube and do the configuration as shown in the screenshot attached below.

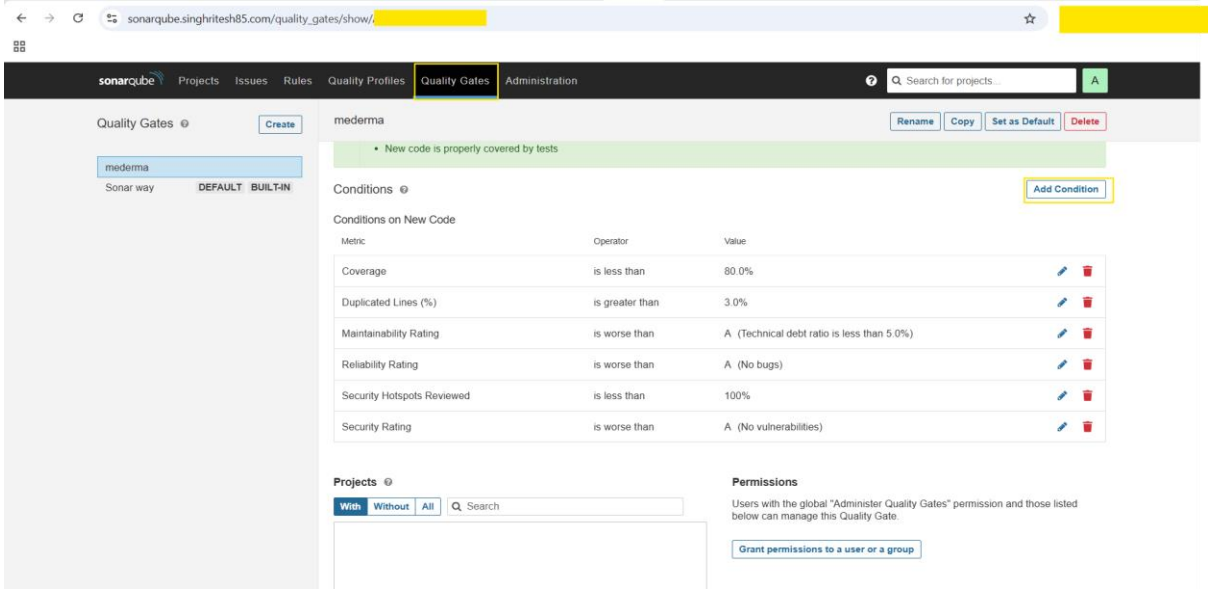


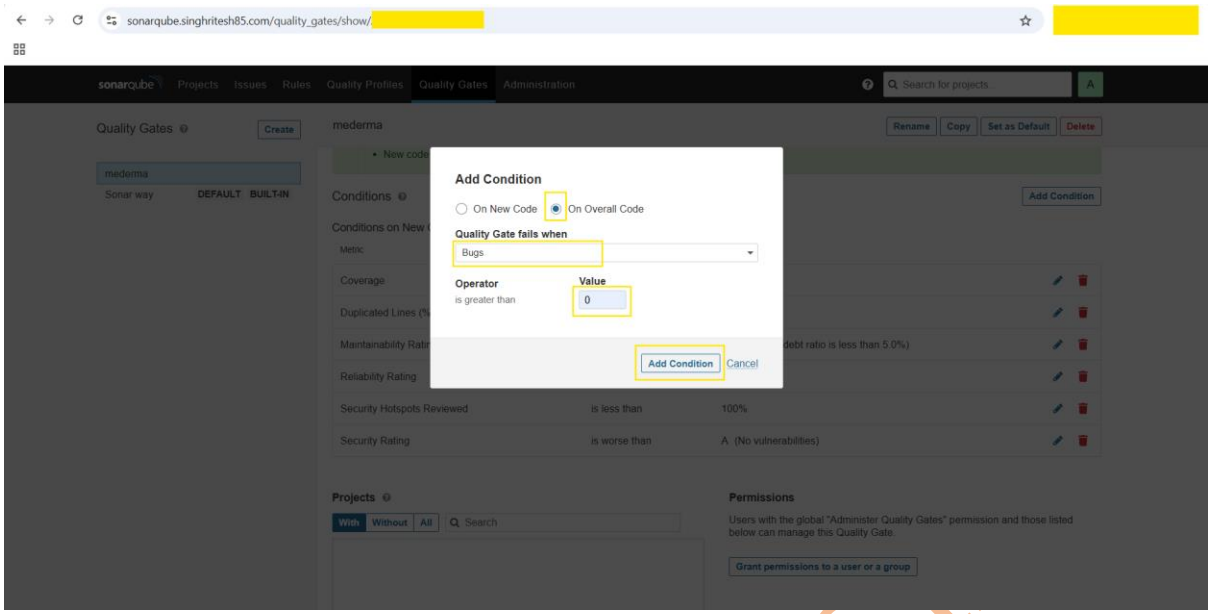
In SonarQube the **Quality Gate** is the predefined condition and threshold which decides the minimum Quality Standards of the code. During Jenkins Pipeline I had tested whether the Code passed the Quality Gate or not. If it does not pass the Quality Gate then Jenkins Job will fail there itself and do not move for further stages. Below screenshot shows how I created the Quality Gate in SonarQube for this Project.

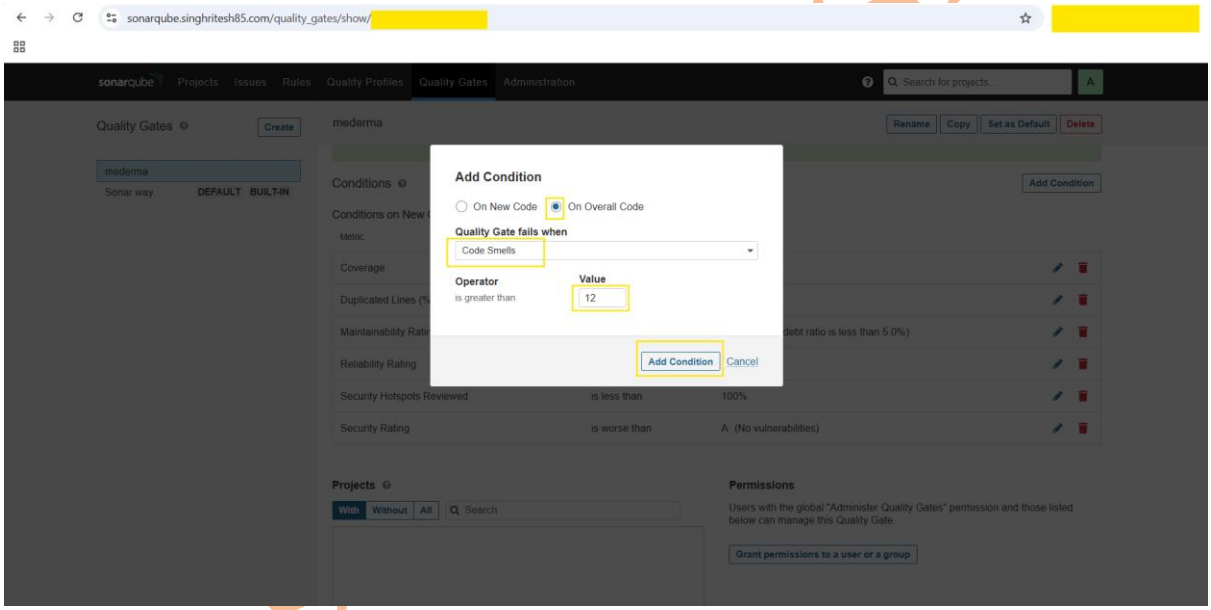


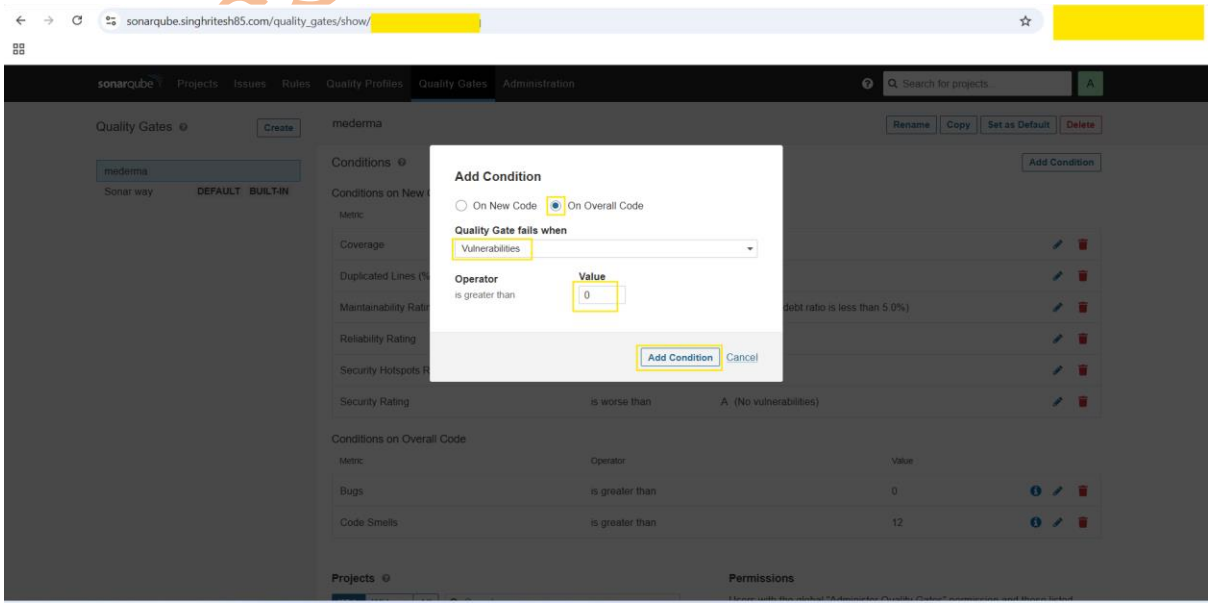

sonarqube.singhritesh85.com/quality_gates/show/
Quality Gates
Create
Sonar way BUILT-IN
Copy
Create Quality Gate
All fields marked with * are required
Name *
mederma
Save
Cancel
Conditions
Conditions on New Code
Metric
Operator
Value
Coverage
is less than
80.0%
Duplicated Lines (%)
is greater than
3.0%
Maintainability Rating
is worse than
A (Technical debt ratio is less than 5.0%)
Reliability Rating
is worse than
A (No bugs)
Security Hotspots Reviewed
is less than
100%
Security Rating
is worse than
A (No vulnerabilities)


sonarqube.singhritesh85.com/quality_gates/show/
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
Quality Gates
Create
mederma
Sonar way DEFAULT BUILT-IN
mederma
Rename
Copy
Set as Default
Delete
All new security hotspots are reviewed
New code has limited technical debt
New code has limited duplication
New code is properly covered by tests
Conditions
Conditions on New Code
Metric
Operator
Value
Coverage
is less than
80.0%
Duplicated Lines (%)
is greater than
3.0%
Maintainability Rating
is worse than
A (Technical debt ratio is less than 5.0%)
Reliability Rating
is worse than
A (No bugs)
Security Hotspots Reviewed
is less than
100%
Security Rating
is worse than
A (No vulnerabilities)
You may click unlock to edit this quality gate. Adding extra conditions to a compliant quality gate can result in drawbacks. Are you reconsidering Clean as You Code?
We strongly recommend this methodology to achieve a Clean Code status.
Unlock editing

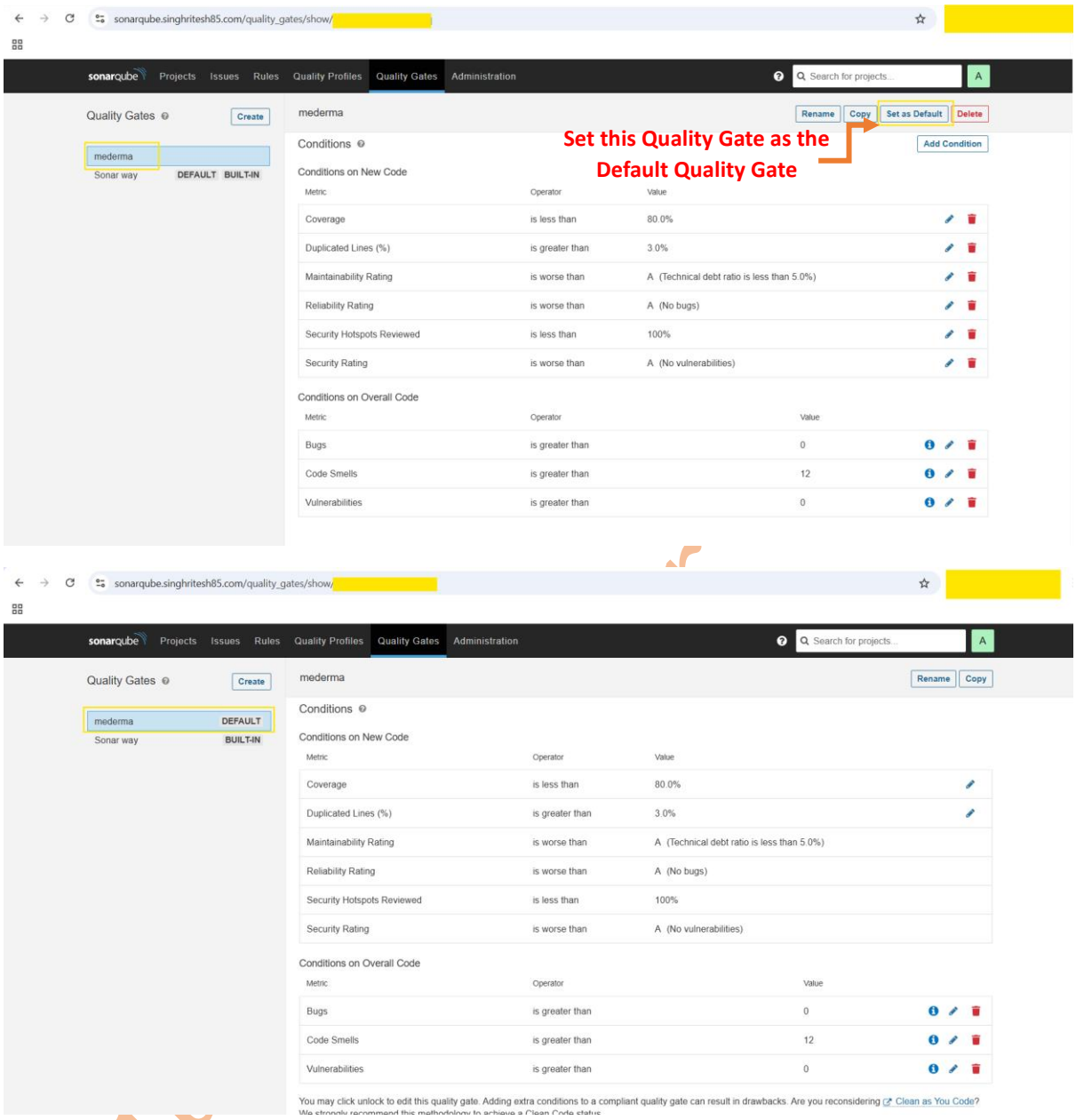

sonarqube.singhritesh85.com/quality_gates/show/
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
Quality Gates
Create
mederma
Sonar way DEFAULT BUILT-IN
mederma
Rename
Copy
Set as Default
Delete
New code is properly covered by tests
Conditions
Add Condition
Conditions on New Code
Metric
Operator
Value
Coverage
is less than
80.0%
Duplicated Lines (%)
is greater than
3.0%
Maintainability Rating
is worse than
A (Technical debt ratio is less than 5.0%)
Reliability Rating
is worse than
A (No bugs)
Security Hotspots Reviewed
is less than
100%
Security Rating
is worse than
A (No vulnerabilities)
Projects
With
Without
All
Search
Permissions
Users with the global "Administer Quality Gates" permission and those listed below can manage this Quality Gate.
Grant permissions to a user or a group


sonarqube.singhritesh85.com/quality_gates/show/
Add Condition
On New Code
On Overall Code
Quality Gate fails when
Bugs
Operator
is greater than
Value
0
Add Condition
Cancel

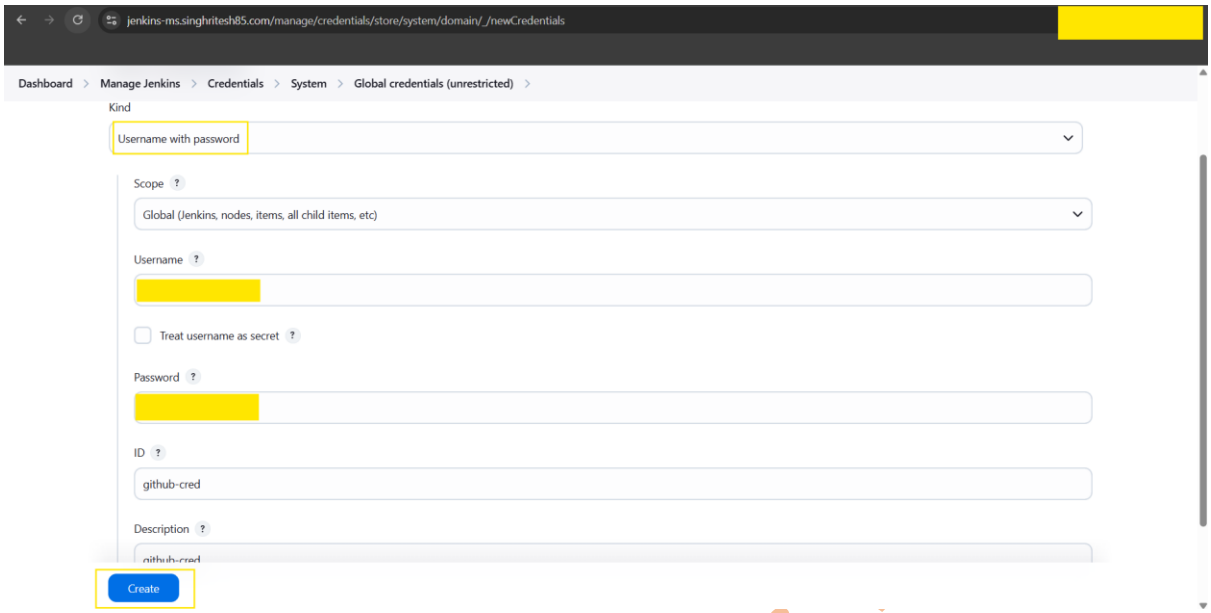

sonarqube.singhritesh85.com/quality_gates/show/
Add Condition
On New Code
On Overall Code
Quality Gate fails when
Code Smells
Operator
is greater than
Value
12
Add Condition
Cancel


sonarqube.singhritesh85.com/quality_gates/show/
Add Condition
On New Code
On Overall Code
Quality Gate fails when
Vulnerabilities
Operator
is greater than
Value
0
Add Condition
Cancel

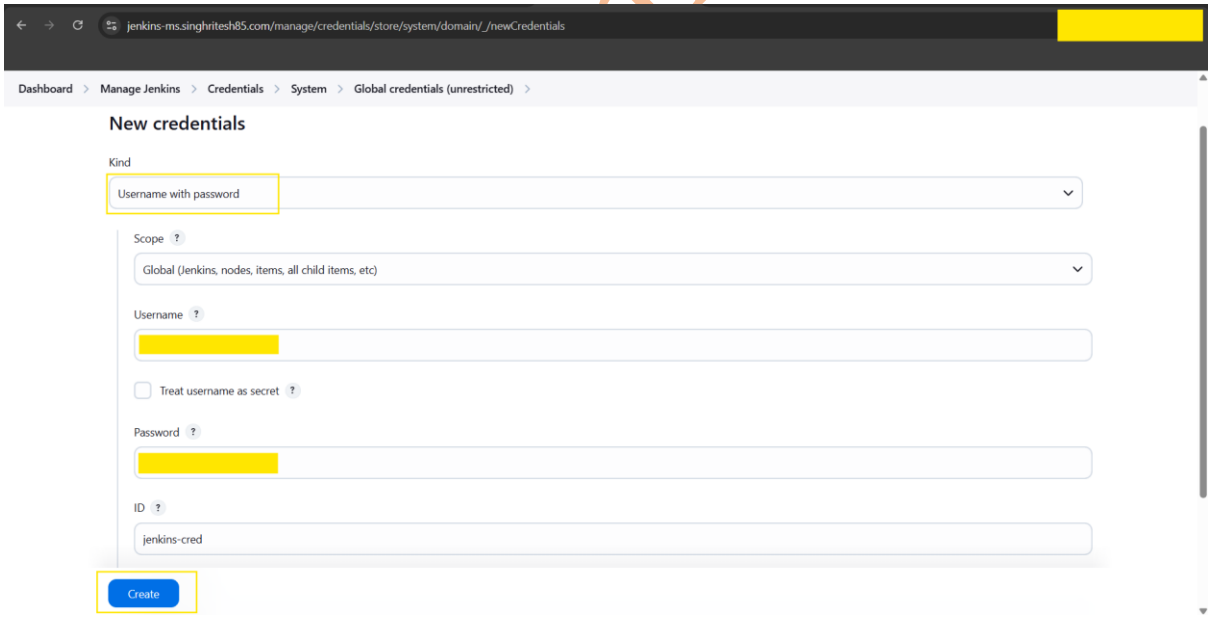
Finally, the predefine condition/threshold for Quality Gate of the SonarQube is as shown in the screenshot attached below.



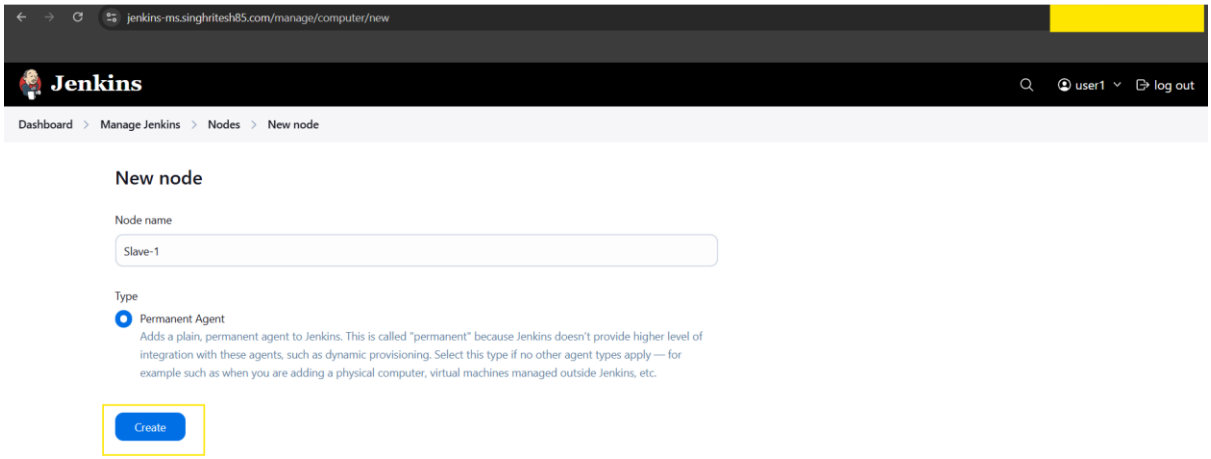
I had created the credential named as github-cred in Jenkins for Authenticating GitHub with Jenkins as shown in the screenshot attached below.

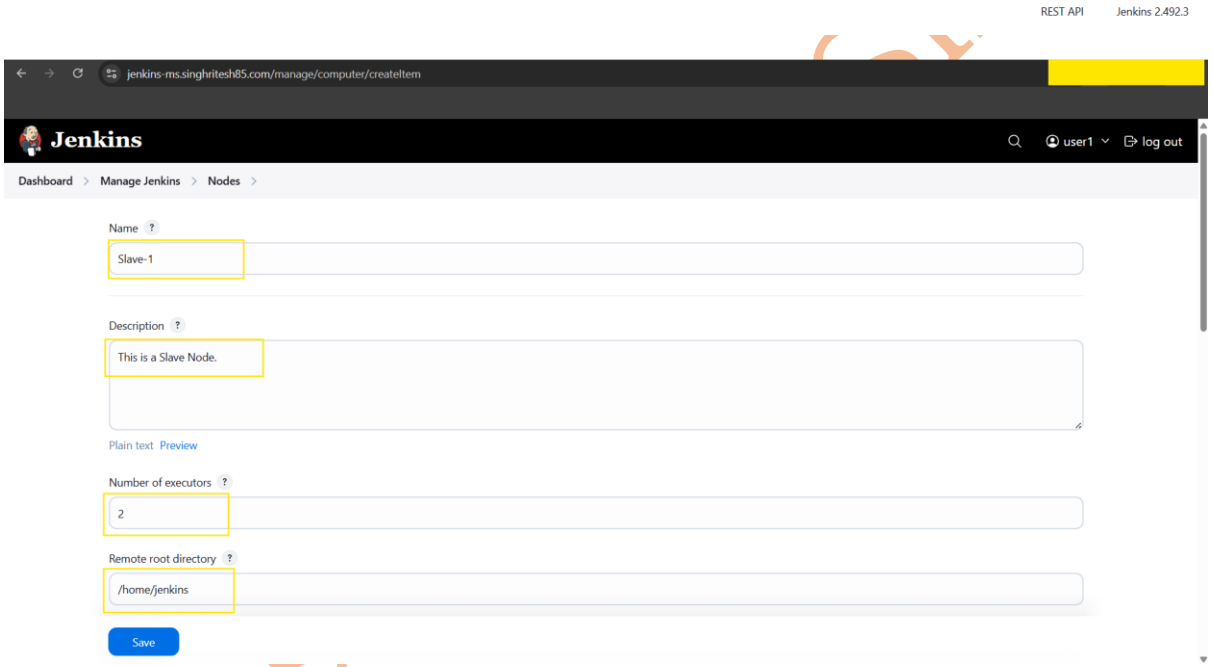


Create a Jenkins Secret to integrate Jenkins Slave Node with Jenkins Master as shown in the screenshot attached below.

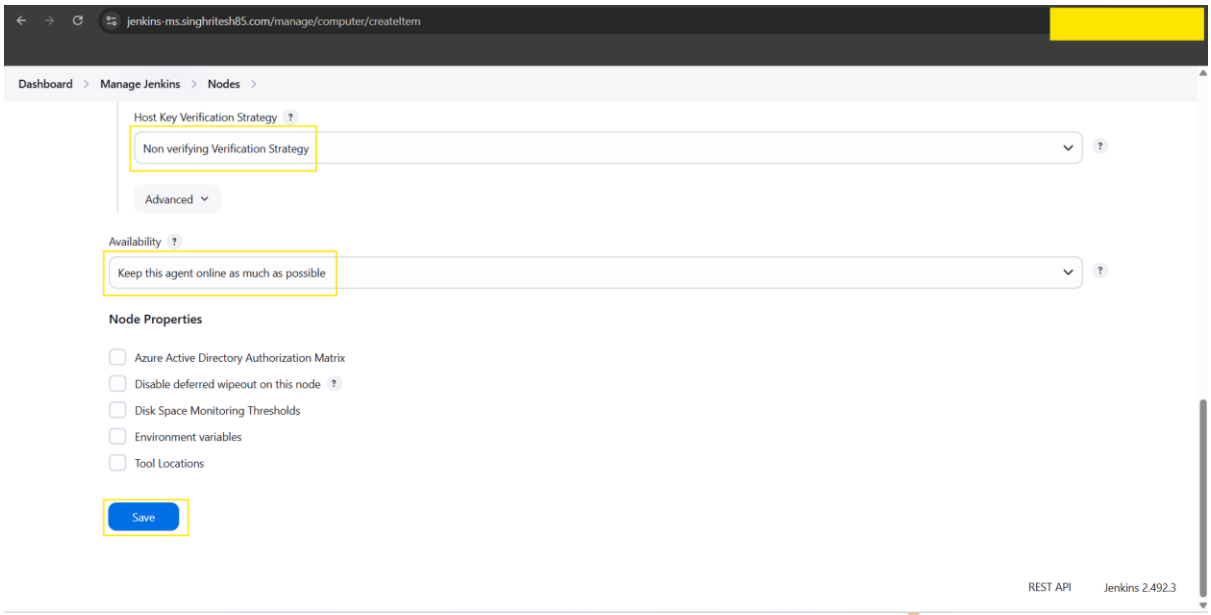


I had integrated a Slave node in Jenkins with the name of **Slave-1**. To do so go to **Manage Jenkins > Nodes > New Node** as shown in the screenshot attached below.

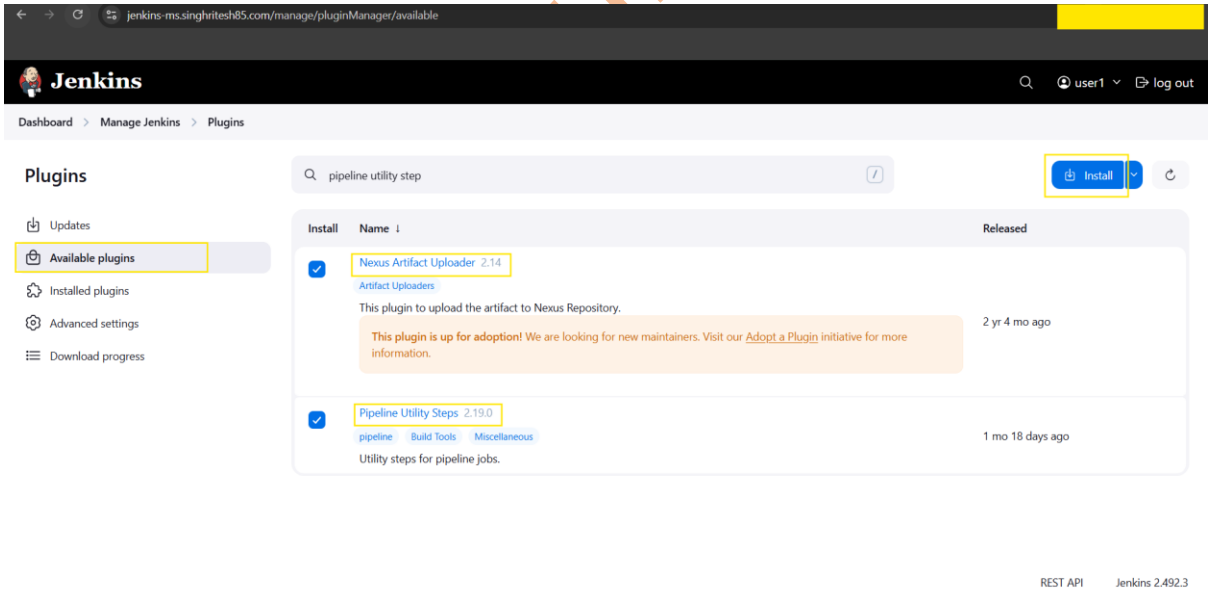

jenkins-ms.singhritesh85.com/manage/computer/new
Jenkins
user1
log out
Dashboard > Manage Jenkins > Nodes > New node
New node
Node name
Slave-1
Type
Permanent Agent
Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these 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.
Create


REST API
Jenkins 2.492.3
jenkins-ms.singhritesh85.com/manage/computer/createItem
Jenkins
user1
log out
Dashboard > Manage Jenkins > Nodes >
Name
Slave-1
Description
This is a Slave Node.
Plain text Preview
Number of executors
2
Remote root directory
/home/jenkins
Save

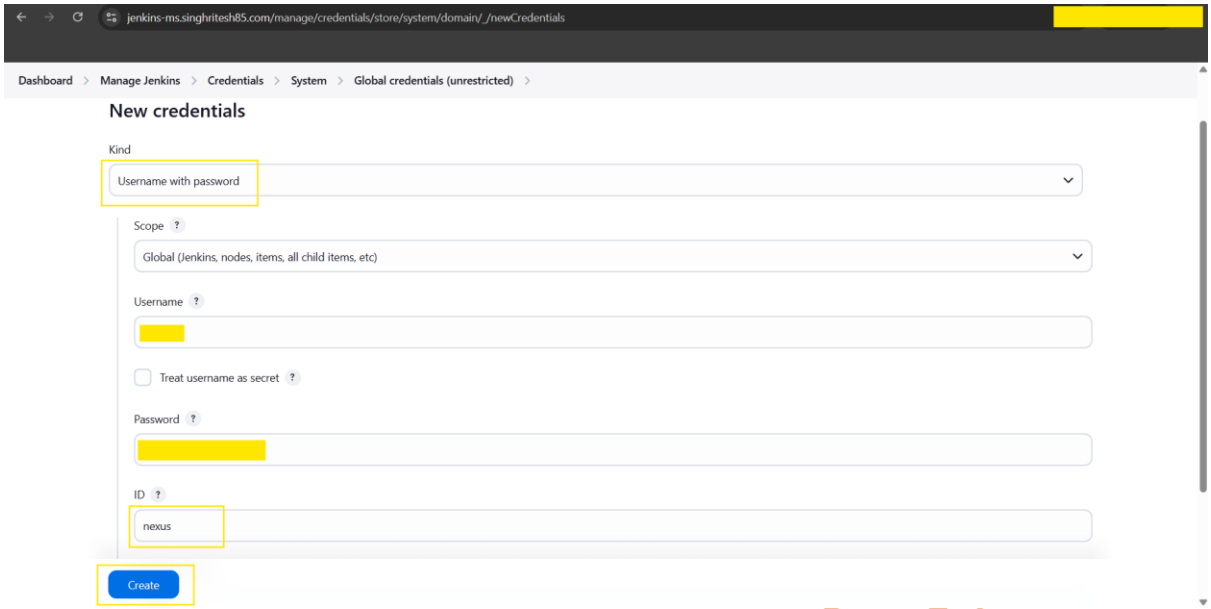

jenkins-ms.singhritesh85.com/manage/computer/createItem
Dashboard > Manage Jenkins > Nodes >
Labels
Slave-1
Usage
Use this node as much as possible
Launch method
Launch agents via SSH
Host
Private IP Address of Jenkins Slave
Credentials
+ Add
Host Key Verification Strategy
Save



Finally, the Slave-1 is in Online mode. To **integrate Jenkins with Sonatype Nexus3** and **to use it in Jenkins Pipeline** I installed the **Nexus Artifact Uploader** and **Pipeline Utility Step** Plugin which is as shown in the screenshot attached below. **Do not restart the Jenkins** after installation of these two plugins.



I had created a credential in Jenkins for Sonatype Nexus3 as shown in the screenshot attached below.



I had provided restricted access to the deployment user **jenkins** using Service Account, Role and Role Binding as shown in the screenshot attached below. The deployment user had all the accesses in the namespaces **bankapp** and **mysql** but does not had access for the entire EKS cluster. That means deployment user **jenkins** access was restricted to the namespaces **bankapp** and **mysql** in the EKS Cluster.

```
[root@                ]# cat sa-role-rolebinding.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: bankapp
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins-bankapp
  namespace: bankapp
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: user-role
  namespace: bankapp
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: user-rolebinding
  namespace: bankapp
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: user-role
subjects:
- namespace: bankapp
  kind: ServiceAccount
  name: jenkins-bankapp
[root@              ~]# kubectl apply -f sa-role-rolebinding.yaml
namespace/bankapp created
serviceaccount/jenkins-bankapp created
role.rbac.authorization.k8s.io/user-role created
rolebinding.rbac.authorization.k8s.io/user-rolebinding created
```

```
cat sa-role-rolebinding.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: bankapp
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins-bankapp
  namespace: bankapp
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: user-role
  namespace: bankapp
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: user-rolebinding
  namespace: bankapp
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: user-role
subjects:
- namespace: bankapp
  kind: ServiceAccount
  name: jenkins-bankapp
```

Created Kubernetes Secrets Which Token was utilized in the kubeconfig file (which was shared with the deployment user jenkins) as shown in the screenshot attached below.

```
[root@                ~]# cat secrets.yaml
apiVersion: v1
kind: Secret
type: kubernetes.io/service-account-token
metadata:
  name: mysecretname
  namespace: bankapp
  annotations:
    kubernetes.io/service-account.name: jenkins-bankapp
[root@                ~]# kubectl apply -f secrets.yaml
secret/mysecretname created
[root@                ~]# kubectl get secrets -n bankapp
NAME           TYPE                                  DATA   AGE
mysecretname   kubernetes.io/service-account-token   3      9s
```

cat secrets.yaml

apiVersion: v1

kind: Secret

type: kubernetes.io/service-account-token

metadata:

name: mysecretname

namespace: bankapp

annotations:

kubernetes.io/service-account.name: jenkins-bankapp

```
[root@          ~]# kubectl describe secrets mysecretname -n bankapp
Name:         mysecretname
Namespace:    bankapp
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: jenkins-bankapp
              kubernetes.io/service-account.uid: 8                              e

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1107 bytes
namespace:  7 bytes
token:
```

```
[root@                    ]# cat sa-role-rolebinding-mysql.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: mysql
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins-mysql
  namespace: mysql
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: user-role
  namespace: mysql
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: user-rolebinding
  namespace: mysql
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: user-role
subjects:
- namespace: mysql
  kind: ServiceAccount
  name: jenkins-mysql

[root@                    ~]# kubectl apply -f sa-role-rolebinding-mysql.yaml
namespace/mysql created
serviceaccount/jenkins-mysql created
role.rbac.authorization.k8s.io/user-role created
```

cat sa-role-rolebinding-mysql.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: mysql
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: jenkins-mysql
  namespace: mysql
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: user-role
  namespace: mysql
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: user-rolebinding
  namespace: mysql
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: user-role
subjects:
- namespace: mysql
  kind: ServiceAccount
  name: jenkins-mysql
```

Created Kubernetes Secrets Which Token was utilized in the kubeconfig file (which was shared with the deployment user jenkins) as shown in the screenshot attached below.

```
[root@ ~]# cat secrets-mysql.yaml
apiVersion: v1
kind: Secret
type: kubernetes.io/service-account-token
metadata:
  name: mysecretname-mysql
  namespace: mysql
  annotations:
    kubernetes.io/service-account.name: jenkins-mysql
[root@ ~]# kubectl apply -f secrets-mysql.yaml
secret/mysecretname-mysql created
[root@ ~]# kubectl get secrets -n mysql
NAME                  TYPE                                  DATA   AGE
mysecretname-mysql    kubernetes.io/service-account-token   3      13s
```

```
[root@ ~]# kubectl get secrets -n mysql
NAME                  TYPE                                  DATA   AGE
mysecretname-mysql    kubernetes.io/service-account-token   3      13s
[root@ ~]# kubectl describe secrets mysecretname-mysql -n mysql
Name:         mysecretname-mysql
Namespace:    mysql
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: jenkins-mysql
              kubernetes.io/service-account.uid: c

Type:  kubernetes.io/service-account-token

Data
====
token:

ca.crt:     1107 bytes
namespace:  5 bytes
```

Below is the screenshot of kubeconfig file which I shared with the deployment user **jenkins**. The deployment user **jenkins** will create a directory named as **.kube** and will keep the kubeconfig file at this path as shown in the screenshot attached. The 600 permissions had been provided to the file .kube/config using the command **chmod 600 ~/.kube/config**.

```
[jenkins@ ~]$ chmod 600 .kube/config
[jenkins@ ~]$ cat .kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data:

    server: https://9.us-east-2.eks.amazonaws.com
  name: arn:aws:eks:us-east-2:02:cluster/eks-demo-cluster-dev
contexts:
- context:
    cluster: arn:aws:eks:us-east-2:02:cluster/eks-demo-cluster-dev
    user: jenkins
  name: mederma
current-context: mederma
kind: Config
preferences: {}
users:
- name: jenkins
  user:
    token:
```

After getting the kubeconfig file the deployment user **jenkins** checked its access in EKS Cluster and found that they have only the access inside the **bankapp** and **mysql** namespaces and not for the entire EKS Cluster as shown in the screenshot attached below.

```
[jenkins@                ~]$ kubectl get nodes
Error from server (Forbidden): nodes is forbidden: User "system:serviceaccount:bankapp:jenkins-bankapp" cannot list resource "nodes" in API group "" at the cl
uster scope
[jenkins@                ~]$ kubectl get pods -n mysql --context=mysql
No resources found in mysql namespace.
[jenkins@                ~]$ kubectl get pods -n bankapp --context=bankapp
No resources found in bankapp namespace.
```

I had created two Jenkins Job one for Bank Application and another for MySQL 8 Pod deployment as shown in the screenshot attached below.

Jenkinsfile-MySQL

```
pipeline{
  agent{
    node{
      label "Slave-1"
      customWorkspace "/home/jenkins/mysql"
    }
  }
  environment{
    JAVA_HOME="/usr/lib/jvm/java-17-amazon-corretto.x86_64"
    PATH="$PATH:$JAVA_HOME/bin:/opt/apache-maven/bin:/opt/node-v16.0.0/bin:/usr/local/bin"
  }
  stages{
    stage("MySQL-Deployment"){
      steps{
        //MySQL
        sh 'argocd login argocd.singhritesh85.com --username admin --password Admin@123 --skip-test-tls  --grpc-web'
        sh 'argocd app create mysql --project default --repo https://github.com/singhritesh85/helm-repo-for-bitnami.git --path ./bitnami/mysql --dest-namespace mysql --sync-option CreateNamespace=true --dest-server https://kubernetes.default.svc --helm-set secondary.replicaCount=1 --helm-set primary.persistence.enabled=true --helm-set primary.persistence.size=1Gi --helm-set architecture=replication --helm-set secondary.persistence.enabled=true --helm-set secondary.persistence.size=1Gi --helm-set primary.service.type=ClusterIP --helm-set auth.rootPassword=Dexter@123 --helm-set auth.database=bankappdb --upsert'
        sh 'argocd app sync mysql'
      }
    }
  }
  post {
    always {
      mail bcc: '', body: "A Jenkins Job with Job Name ${env.JOB_NAME} has been executed", cc: '', from: '', replyTo: '', subject: "Jenkins Job ${env.JOB_NAME} has been executed", to: 'abc@gmail.com'
    }
    success {
```

```
        mail bcc: '', body: "A Jenkins Job with Job Name ${env.JOB_NAME} and Build
Number=${env.BUILD_NUMBER} has been executed Successfully, Please Open the URL
${env.BUILD_URL} and click on Console Output to see the Log. The Result of execution is
${currentBuild.currentResult}", cc: '', from: '', replyTo: '', subject: "Jenkins Job ${env.JOB_NAME} has
been Sucessfully Executed", to: 'abc@gmail.com'

    }

    failure {

        mail bcc: '', body: "A Jenkins Job with Job Name ${env.JOB_NAME} and Build
Number=${env.BUILD_NUMBER} has been Failed, Please Open the URL ${env.BUILD_URL} and click
on Console Output to see the Log. The Result of execution is ${currentBuild.currentResult}", cc: '',
from: '', replyTo: '', subject: "Jenkins Job ${env.JOB_NAME} has been Failed", to: 'abc@gmail.com'

    }

  }

}
```

Jenkinsfile-BankApp-BlueGreen-Deployment

```
pipeline{
   agent{
       node{
          label "Slave-1"
          customWorkspace "/home/jenkins/bankapp"
       }
   }
   environment{
      JAVA_HOME="/usr/lib/jvm/java-17-amazon-corretto.x86_64"
      PATH="$PATH:$JAVA_HOME/bin:/opt/apache-maven/bin:/opt/node-v16.0.0/bin:/usr/local/bin"
   }
   parameters {
      string(name: 'COMMIT_ID', defaultValue: '', description: 'Provide the Commit ID')
      string(name: 'REPO_NAME', defaultValue: '', description: 'Provide the ECR Repository Name for Application Image')
      string(name: 'TAG_NAME', defaultValue: '', description: 'Provide the TAG Name')
   }
   stages{
      stage("Clone-Code"){
         steps{
            cleanWs()
            checkout scmGit(branches: [[name: '${COMMIT_ID}']], extensions: [], userRemoteConfigs: [[credentialsId: 'github-cred', url: 'https://github.com/singhritesh85/Bank-App.git']])
         }
      }
      stage("SonarQube-Analysis"){
         steps{
            withSonarQubeEnv('SonarQube-Server') {
               sh 'mvn clean install sonar:sonar'
            }
         }
```

```
    }
    stage("Quality Gate") {
      steps {
        timeout(time: 1, unit: 'HOURS') {
          waitForQualityGate abortPipeline: true
        }
      }
    }
    stage("Nexus-Artifact Upload"){
      steps{
        script{
          def mavenPom = readMavenPom file: 'pom.xml'
          def nexusRepoName = mavenPom.version.endsWith("SNAPSHOT") ? "maven-snapshot" : "maven-release"
          nexusArtifactUploader artifacts: [[artifactId: 'bankapp', classifier: '', file: "target/bankapp-${mavenPom.version}.jar", type: 'jar']], credentialsId: 'nexus', groupId: 'com.example', nexusUrl: 'nexus.singhritesh85.com', nexusVersion: 'nexus3', protocol: 'https', repository: "${nexusRepoName}", version: "${mavenPom.version}"
        }
      }
    }
    stage("Docker Build"){
      steps{
        sh 'docker system prune -f --all'
        sh 'docker build -t myimage:1.01 -f Dockerfile-Project-1 .'
        sh 'docker tag myimage:1.01 ${REPO_NAME}:${TAG_NAME}'
        sh 'trivy image --exit-code 0 --severity MEDIUM,HIGH ${REPO_NAME}:${TAG_NAME}'
        sh 'trivy image  --exit-code 1 --severity CRITICAL ${REPO_NAME}:${TAG_NAME}'
        sh 'aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 027330342406.dkr.ecr.us-east-2.amazonaws.com'
        sh 'docker push ${REPO_NAME}:${TAG_NAME}'
      }
    }
```

```
    stage("Deployment"){
      steps{
        sh 'argocd login argocd.singhritesh85.com --username admin --password Admin@123 --skip-test-tls  --grpc-web'
        sh 'argocd app create bankapp --project default --repo https://github.com/singhritesh85/helm-repo-for-Blue-Green-Deployment.git --path ./folo --dest-namespace bankapp --sync-option CreateNamespace=true --dest-server https://kubernetes.default.svc --helm-set service.port=80 --helm-set image.repository=${REPO_NAME} --helm-set image.tag=${TAG_NAME} --helm-set replicaCount=1 --upsert'
        sh 'argocd app sync bankapp'
      }
    }
  }
  post {
    always {
      mail bcc: '', body: "A Jenkins Job with Job Name ${env.JOB_NAME} has been executed", cc: '', from: '', replyTo: '', subject: "Jenkins Job ${env.JOB_NAME} has been executed", to: 'abc@gmail.com'
    }
    success {
      mail bcc: '', body: "A Jenkins Job with Job Name ${env.JOB_NAME} and Build Number=${env.BUILD_NUMBER} has been executed Successfully, Please Open the URL ${env.BUILD_URL} and click on Console Output to see the Log. The Result of execution is ${currentBuild.currentResult}", cc: '', from: '', replyTo: '', subject: "Jenkins Job ${env.JOB_NAME} has been Sucessfully Executed", to: 'abc@gmail.com'
    }
    failure {
      mail bcc: '', body: "A Jenkins Job with Job Name ${env.JOB_NAME} and Build Number=${env.BUILD_NUMBER} has been Failed, Please Open the URL ${env.BUILD_URL} and click on Console Output to see the Log. The Result of execution is ${currentBuild.currentResult}", cc: '', from: '', replyTo: '', subject: "Jenkins Job ${env.JOB_NAME} has been Failed", to: 'abc@gmail.com'
    }
  }
}
```

The Argo Rollout Controller and Argo Rollout Kubectl Plugins are needed for Blue-Green Deployment using Argo Rollout. Which I installed as discussed below.

Install Argo Rollout Controller using the commands as written below.

kubectl create namespace argo-rollouts

kubectl apply -n argo-rollouts -f https://github.com/argoproj/argo-rollouts/releases/latest/download/install.yaml

```
[root@                    ~]# kubectl create namespace argo-rollouts
```

```
[root@                ~]# kubectl apply -n argo-rollouts -f https://github.com/argoproj/argo-rollouts/releases/latest/download/install.yaml
```

Install Argo Rollouts Kubectl plugin

curl -LO https://github.com/argoproj/argo-rollouts/releases/latest/download/kubectl-argo-rollouts-linux-amd64

chmod +x ./kubectl-argo-rollouts-linux-amd64

sudo mv ./kubectl-argo-rollouts-linux-amd64 /usr/local/bin/kubectl-argo-rollouts

```
[root@                ~]# curl -LO https://github.com/argoproj/argo-rollouts/releases/latest/download/kubectl-argo-rollouts-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  123M  100  123M    0     0   138M      0 --:--:-- --:--:-- --:--:--  266M
[root@                ~]# chmod +x ./kubectl-argo-rollouts-linux-amd64
[root@                ~]# sudo mv ./kubectl-argo-rollouts-linux-amd64 /usr/local/bin/kubectl-argo-rollouts
```

Before Running the Jenkins Job, I changed the Nameserver on Jenkins Slave node as shown in the screenshot attached below. Because for this project the DHCP Option which I am using is the default DHCP Option Set.

```
[root@                    ~]# cat /etc/resolv.conf
; generated by /usr/sbin/dhclient-script
search us-east-2.compute.internal
options timeout:2 attempts:5
nameserver 8.8.8.8    #10.10.0.2
```

As mentioned earlier I had created two Jenkins Job **mysql** and **bankapp.** The **mysql** Jenkins Job do not require any parameter for its execution while **bankapp** Jenkins Job required the string parameters for its execution as shown in the screenshot attached below.

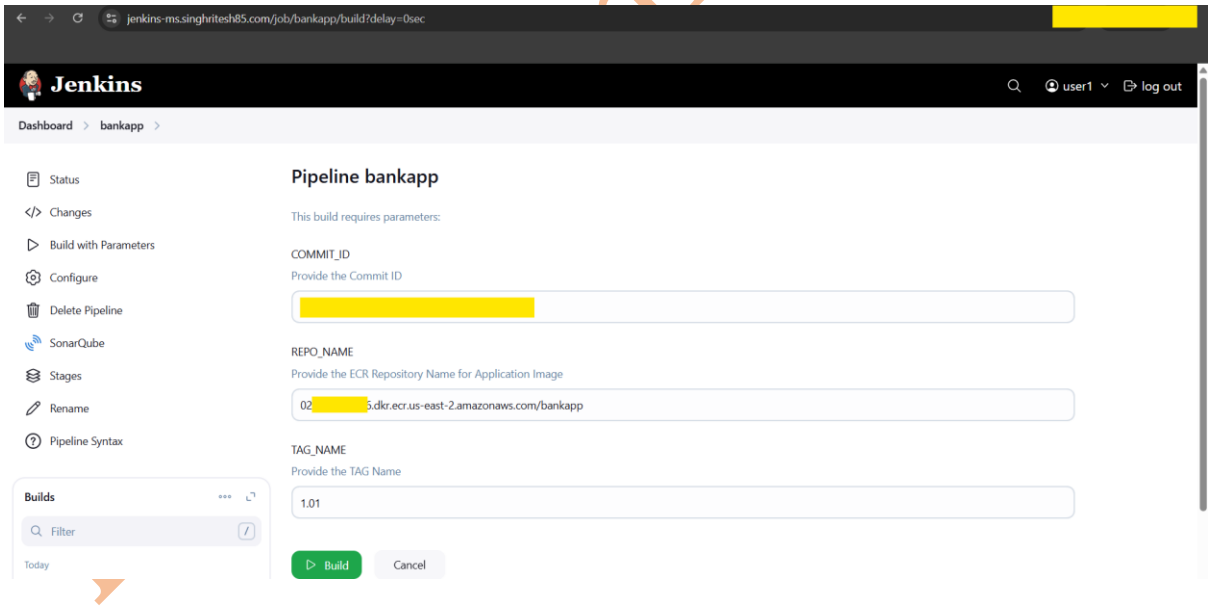
Remember for execution of the **mysql** Jenkins Job I used the helm chart present in my GitHub Repo https://github.com/singhritesh85/helm-repo-for-bitnami.git at the path **bitnami/mysql.** In this project I am using EKS Cluster 1.30 and EKS Cluster 1.30 and later do not support any default storage class while for earlier version it was **gp2**. So, for EKS Cluster 1.30 and later versions you need to explicitly mention the storage class which you want to use. In this project I used the storage class gp2 which I mentioned in the values.yaml file of the helm chart as shown in the screenshot attached below.

helm-repo-for-bitnami / bitnami / mysql / values.yaml

singhritesh85 Update values.yaml

Code Blame 1419 lines (1407 loc) · 63.2 KB

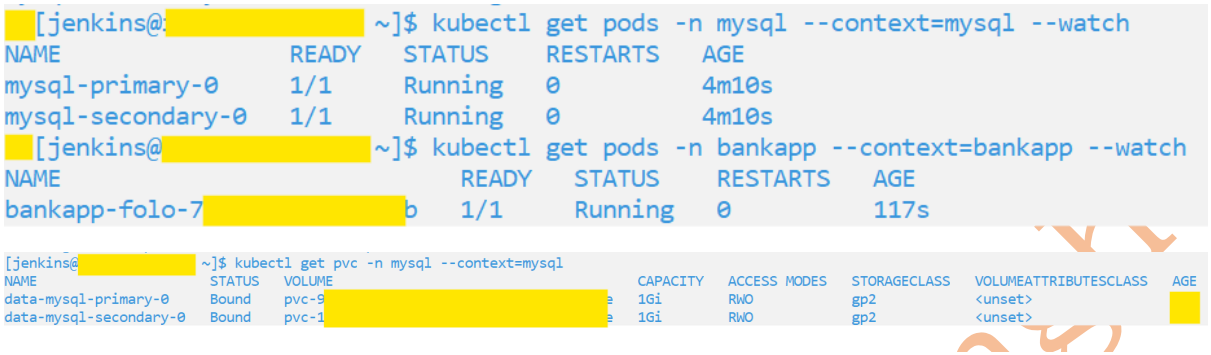
```
1   # Copyright Broadcom, Inc. All Rights Reserved.
2   # SPDX-License-Identifier: APACHE-2.0
3
4   ## @section Global parameters
5   ## Global Docker image parameters
6   ## Please, note that this will override the image parameters, including dependencies, configured to use the global valu
7   ## Current available global Docker image parameters: imageRegistry, imagePullSecrets and storageClass
8   ##
9
10  ## @param global.imageRegistry Global Docker image registry
11  ## @param global.imagePullSecrets Global Docker registry secret names as an array
12  ## @param global.storageClass Global StorageClass for Persistent Volume(s)
13  ##
14  global:
15    imageRegistry: ""
16    ## E.g.
17    ## imagePullSecrets:
18    ##   - myRegistryKeySecretName
19    ##
20    imagePullSecrets: []
21    storageClass: "gp2"  ### EKS 1.30 and later do not support any default storage class, You need to explicitly mention
```

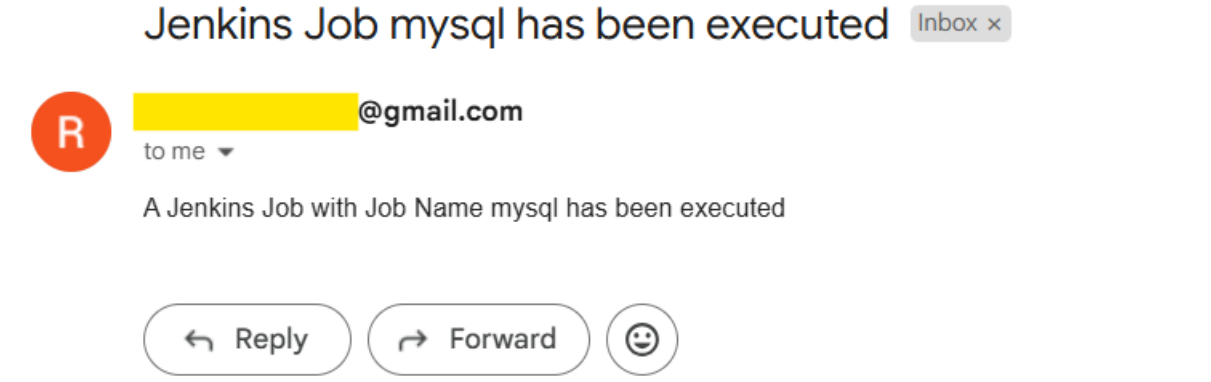


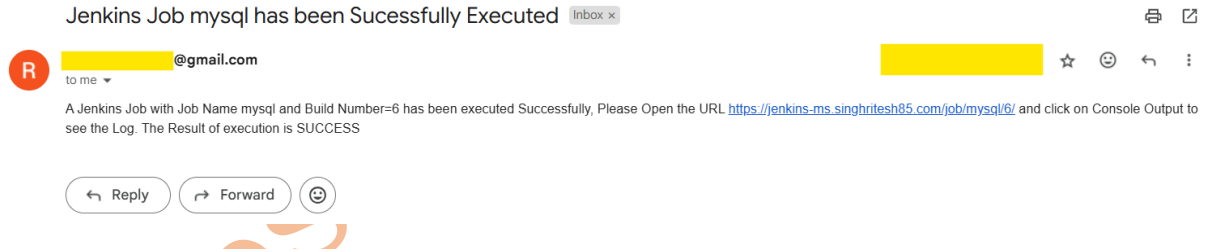
The screenshot for the two Jenkins Job after its execution is as shown below.

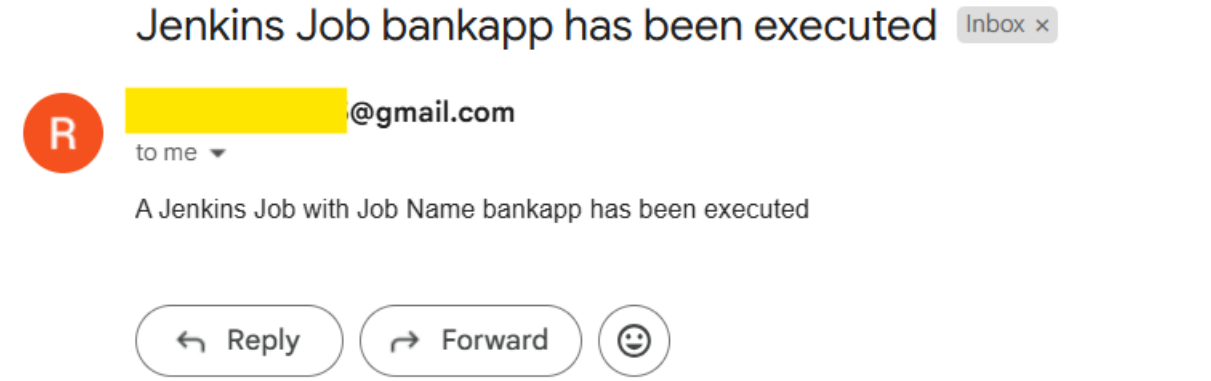
S	W	Name ↑	Last Success	Last Failure	Last Duration
✓	☀	mysql	16 min #2	N/A	2 min 35 sec
✓	⛅	bankapp	9 min 14 sec #3	12 min #1	1 min 52 sec

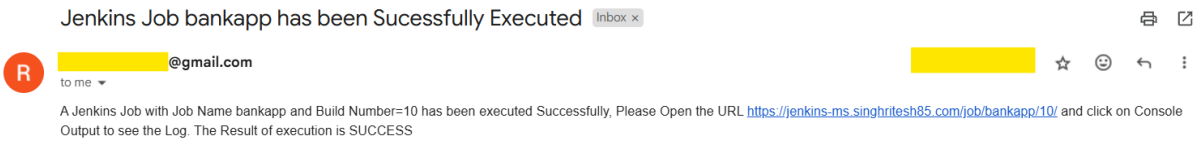
After the successful deployment of the two Jenkins Job **bankapp** and **mysql** I checked the pods and was able to see the Running pods as shown in the screenshot attached below. As I configured Email Notification for Running of Jenkins Job and its Successful completion and found that an Email was sent to the Group Email ID configured as shown in the screenshot attached below.







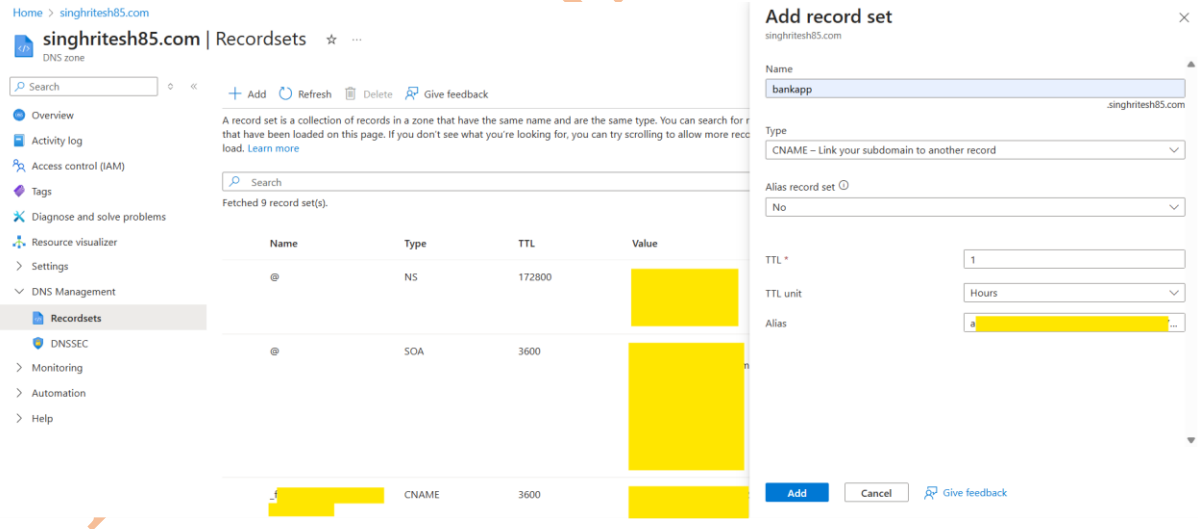




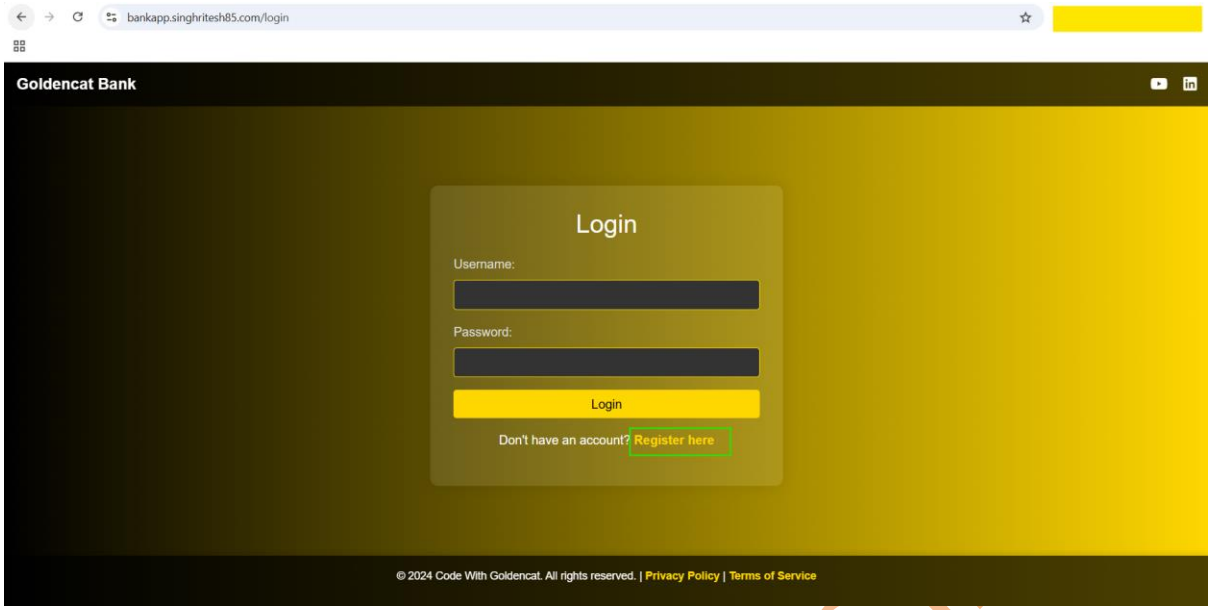
I created the ingress rule for the bank application as shown in the screenshot attached below.

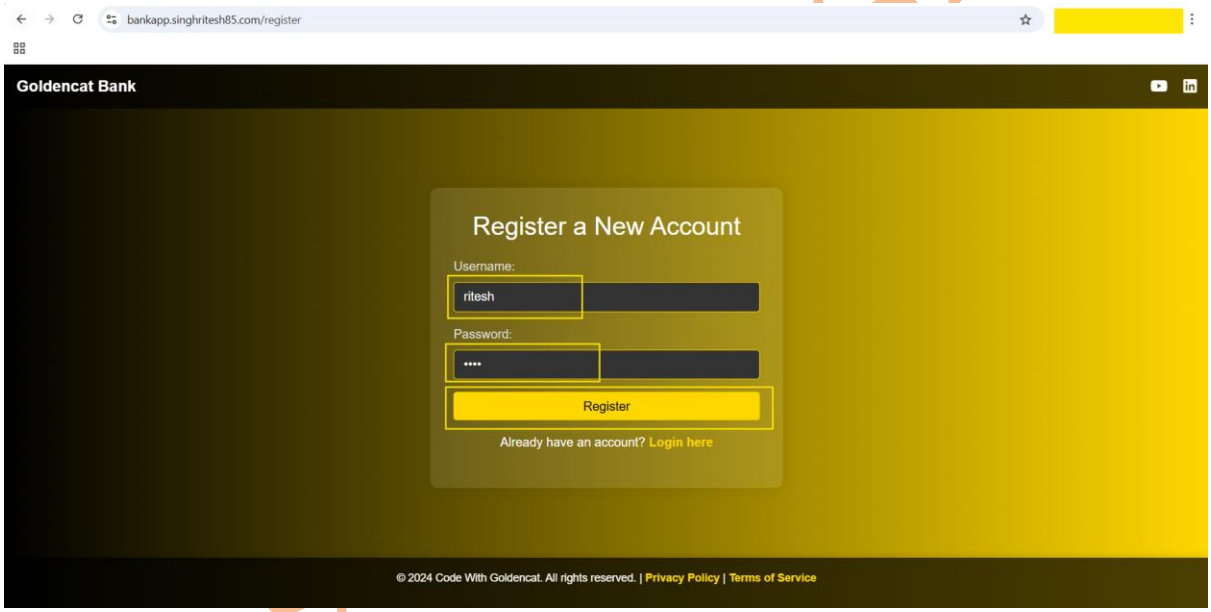
```
[jenkins@ ~]$ cat ingress-rule.yaml
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: bankapp-ingress
  namespace: bankapp
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  ingressClassName: nginx
  rules:
  - host: bankapp.singhritesh85.com
    http:
      paths:
      - path: /
        backend:
          service:
            name: bankapp-folo-active
            port:
              number: 80
        pathType: Prefix
[jenkins@ ~]$ kubectl apply -f ingress-rule.yaml
ingress.networking.k8s.io/bankapp-ingress created
[jenkins@ ~]$ kubectl get ing -n bankapp --context=bankapp
NAME              CLASS   HOSTS                      ADDRESS   PORTS   AGE
bankapp-ingress   nginx   bankapp.singhritesh85.com            80      13s
[jenkins@ ~]$ kubectl get ing -n bankapp --context=bankapp --watch
NAME              CLASS   HOSTS                      ADDRESS   PORTS   AGE
bankapp-ingress   nginx   bankapp.singhritesh85.com            80      17s
bankapp-ingress   nginx   bankapp.singhritesh85.com   a                                                0.us-east-2.elb.amazonaws.com   80      50s
```

I did the entry for DNS Name of the LoadBalancer in Azure DNS Zone to create the Record Set for the HOSTS bankapp.singhritesh85.com as shown in the screenshot attached below.

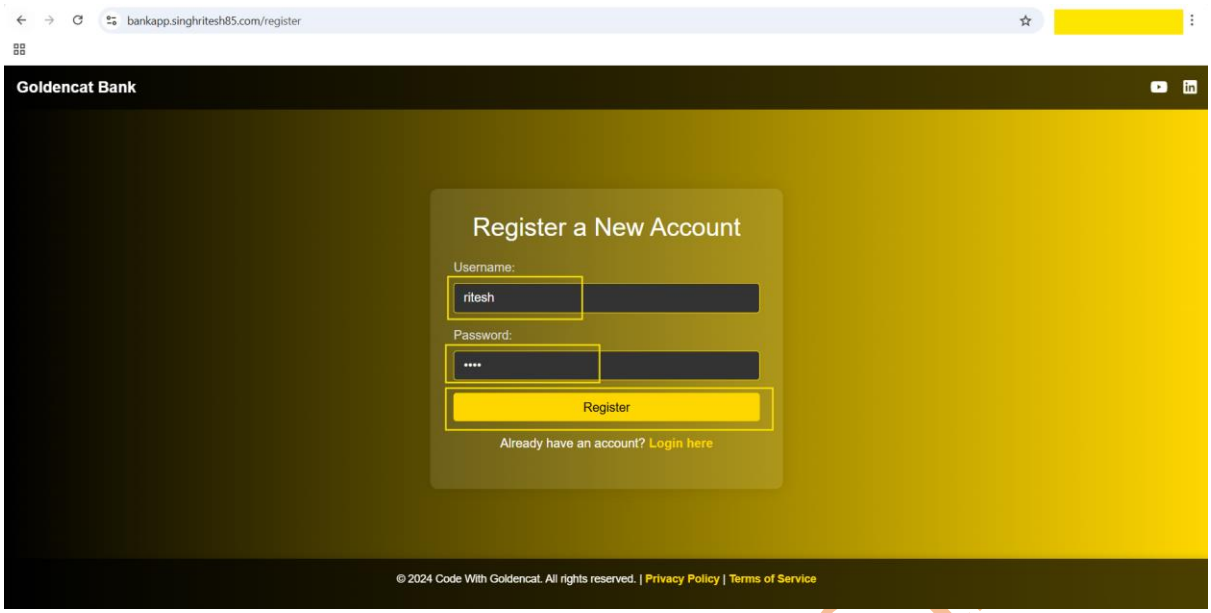


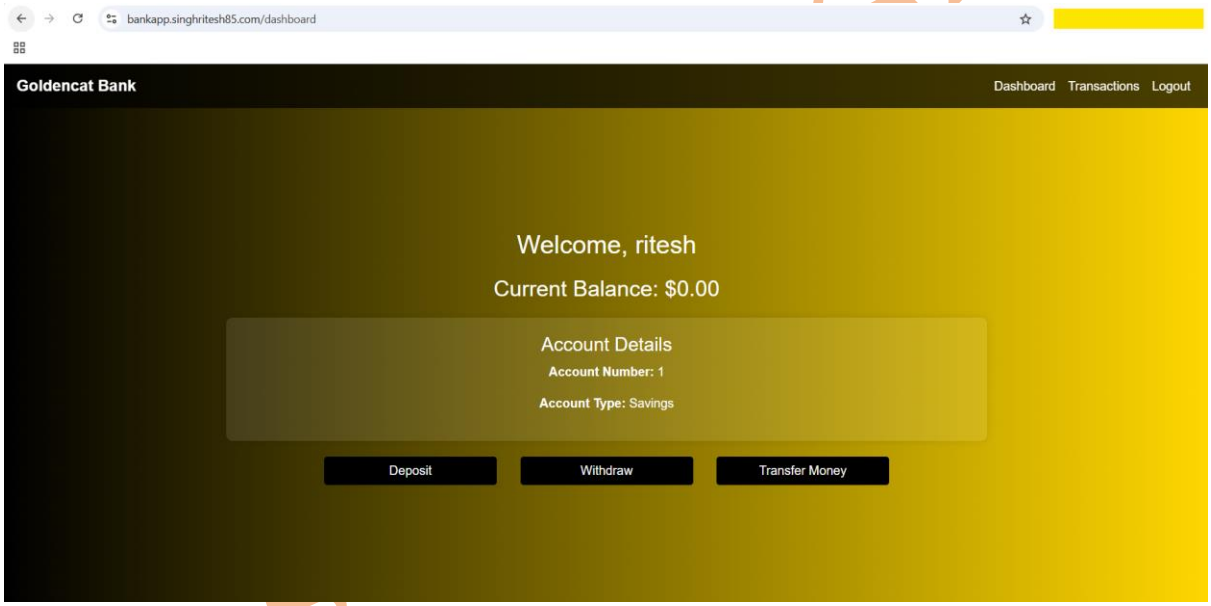
Finally, I was able to access the Bank Application as shown in the screenshot attached below.





Now I tried to login with the created user and found that I was able to login successfully as shown in the screenshot attached below.



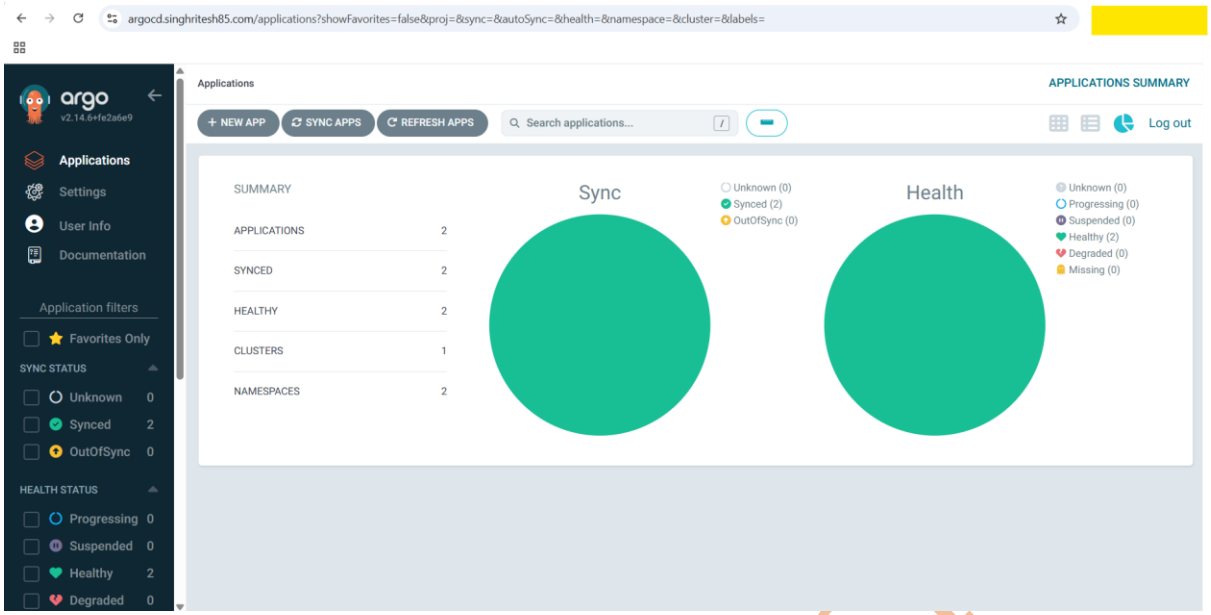


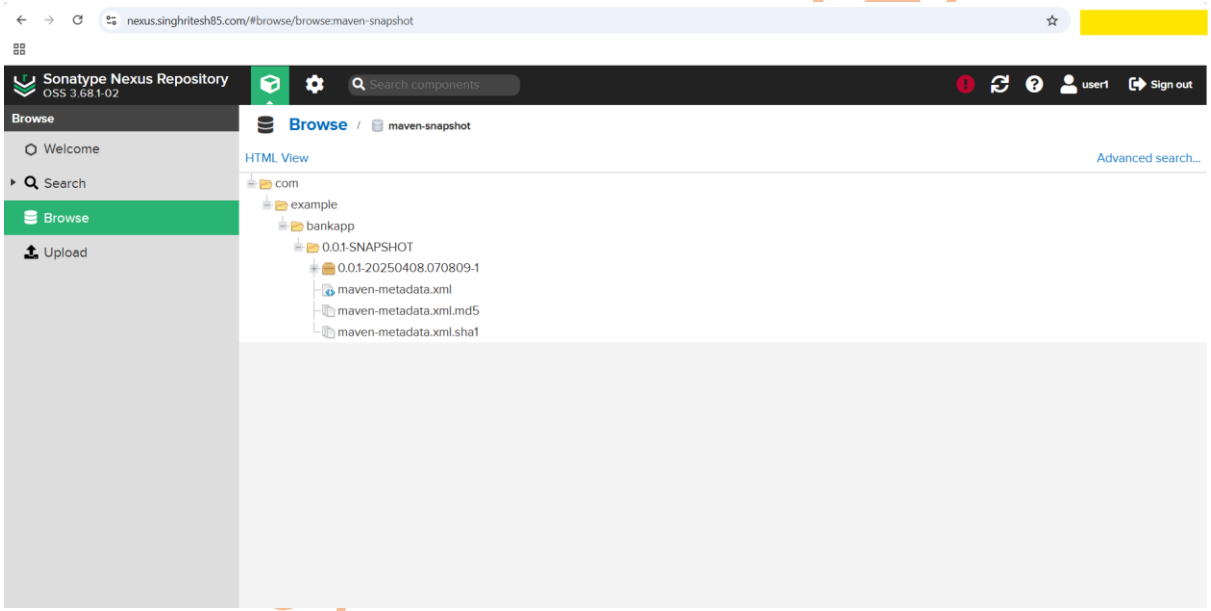
Now I checked the MySQL 8 Pod entry and found that same user was existing in the MySQL 8 database as shown in the screenshot attached below.

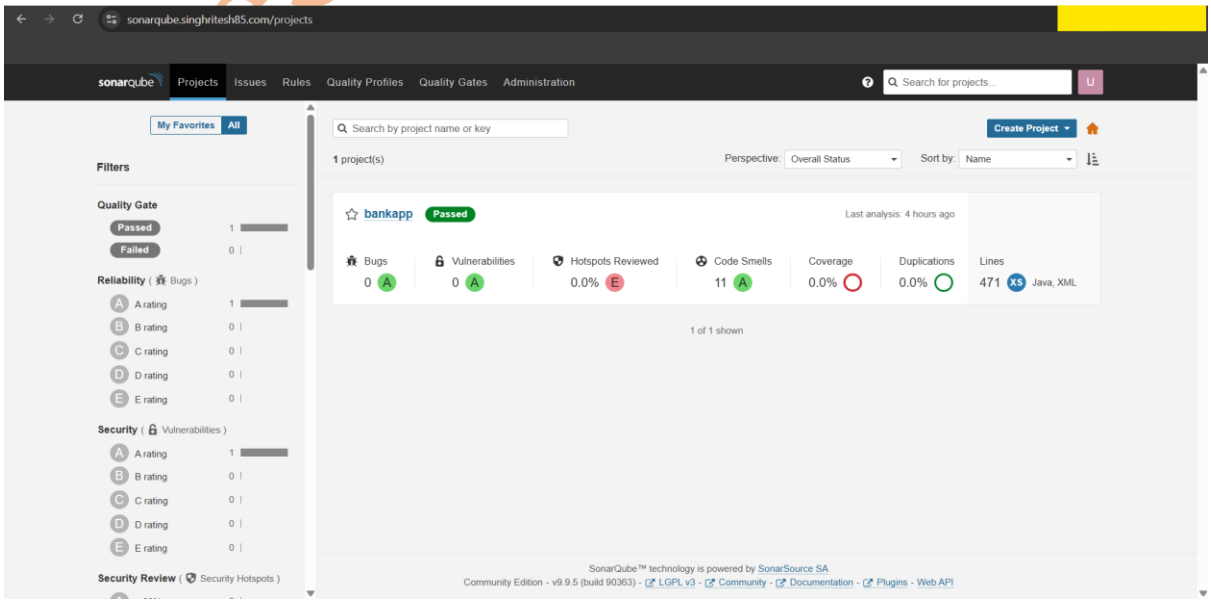
```
[jenkins@                ~]$ kubectl get pods -n mysql --context=mysql
NAME                READY   STATUS    RESTARTS   AGE
mysql-primary-0     1/1     Running   0          18m
mysql-secondary-0   1/1     Running   0          18m
[jenkins@                ~]$ kubectl exec -it mysql-primary-0 -n mysql bash --context=mysql
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
Defaulted container "mysql" out of: mysql, preserve-logs-symlinks (init)
I have no name!@mysql-primary-0:/$ mysql -h localhost -u root --password
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 244
Server version: 8.4.0 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| bankappdb          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

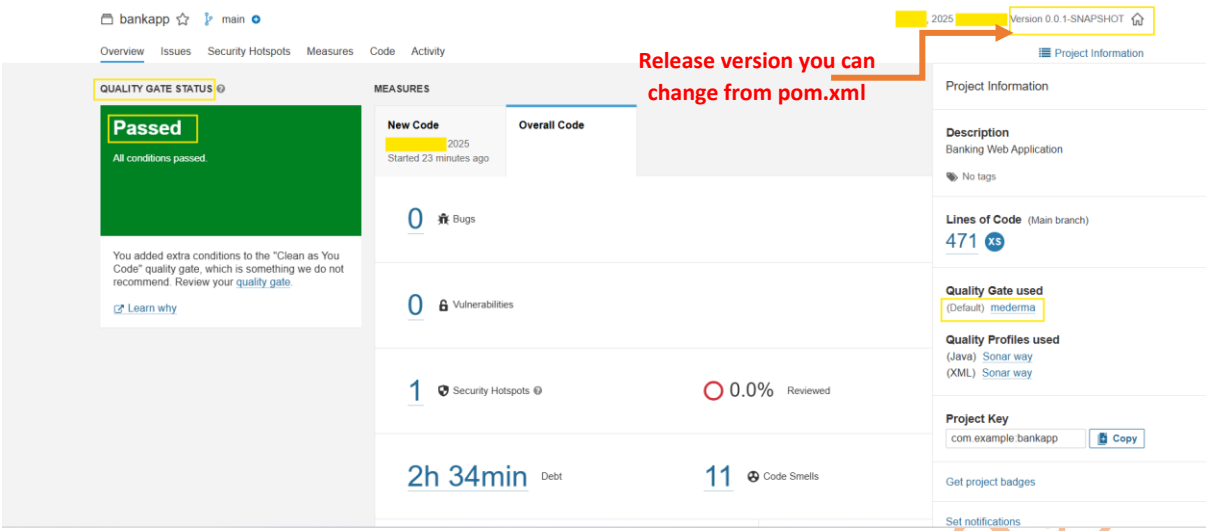
```
mysql> use bankappdb;
Database changed
mysql> show tables;
+----------------------+
| Tables_in_bankappdb |
+----------------------+
| account              |
| transaction          |
+----------------------+
2 rows in set (0.01 sec)

mysql> select * from account;
+----+---------+----------------------------------------------------------------+----------+
| id | balance | password                                                       | username |
+----+---------+----------------------------------------------------------------+----------+
|  1 |    0.00 |                                                                | ritesh   |
+----+---------+----------------------------------------------------------------+----------+
1 row in set (0.00 sec)

mysql>
```

The screenshot for SonarQube UI, Sonatype Nexus3 Repository and ArgoCD is as shown in the screenshot attached below.

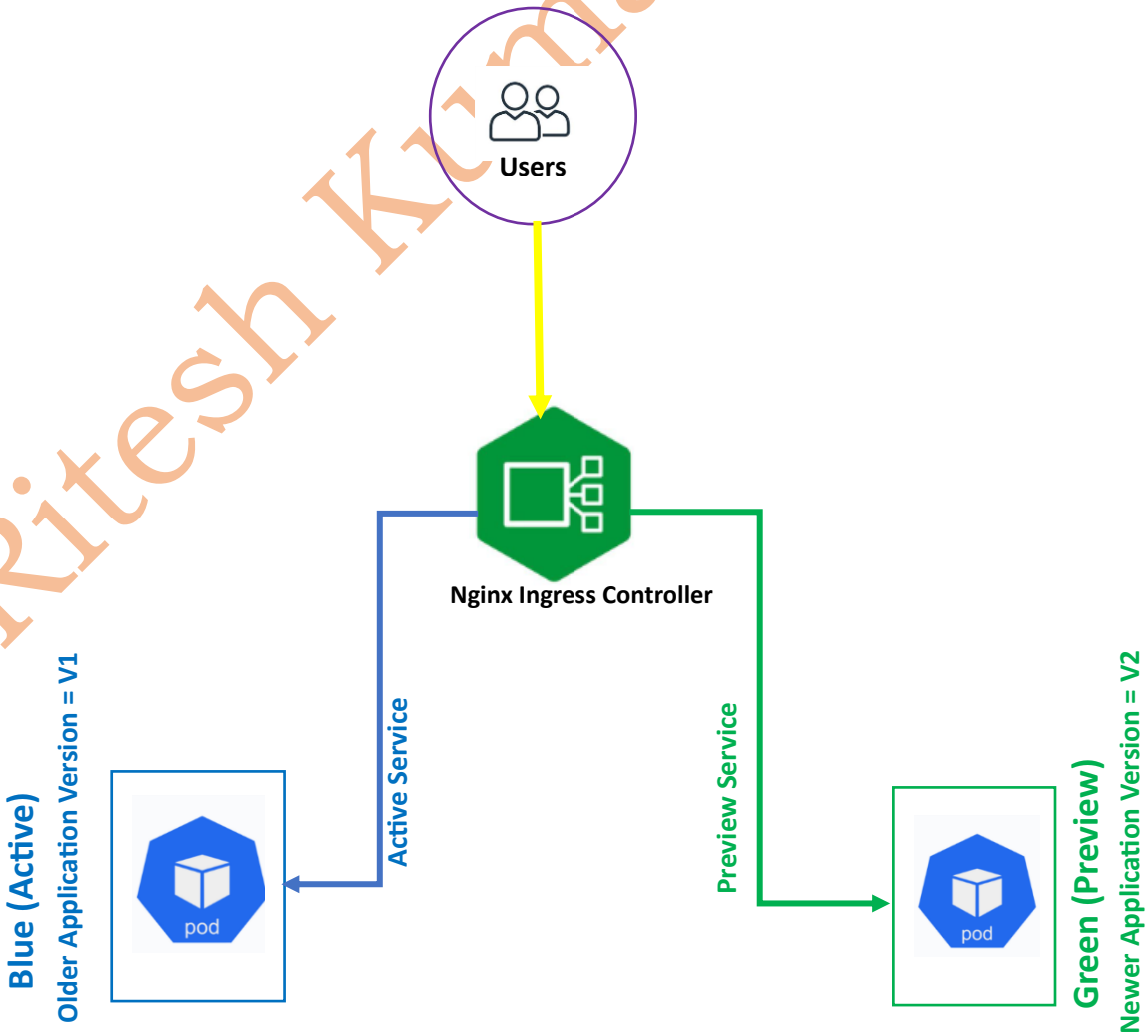

argocd.singhritesh85.com/applications?showFavorites=false&proj=&sync=&autoSync=&health=&namespace=&cluster=&labels=
argo
v2.14.6+fe2a6e9
Applications
Settings
User Info
Documentation
Application filters
Favorites Only
SYNC STATUS
Unknown 0
Synced 2
OutOfSync 0
HEALTH STATUS
Progressing 0
Suspended 0
Healthy 2
Degraded 0
Applications
APPLICATIONS SUMMARY
+ NEW APP
SYNC APPS
REFRESH APPS
Search applications...
Log out
SUMMARY
APPLICATIONS 2
SYNCED 2
HEALTHY 2
CLUSTERS 1
NAMESPACES 2
Sync
Unknown (0)
Synced (2)
OutOfSync (0)
Health
Unknown (0)
Progressing (0)
Suspended (0)
Healthy (2)
Degraded (0)
Missing (0)

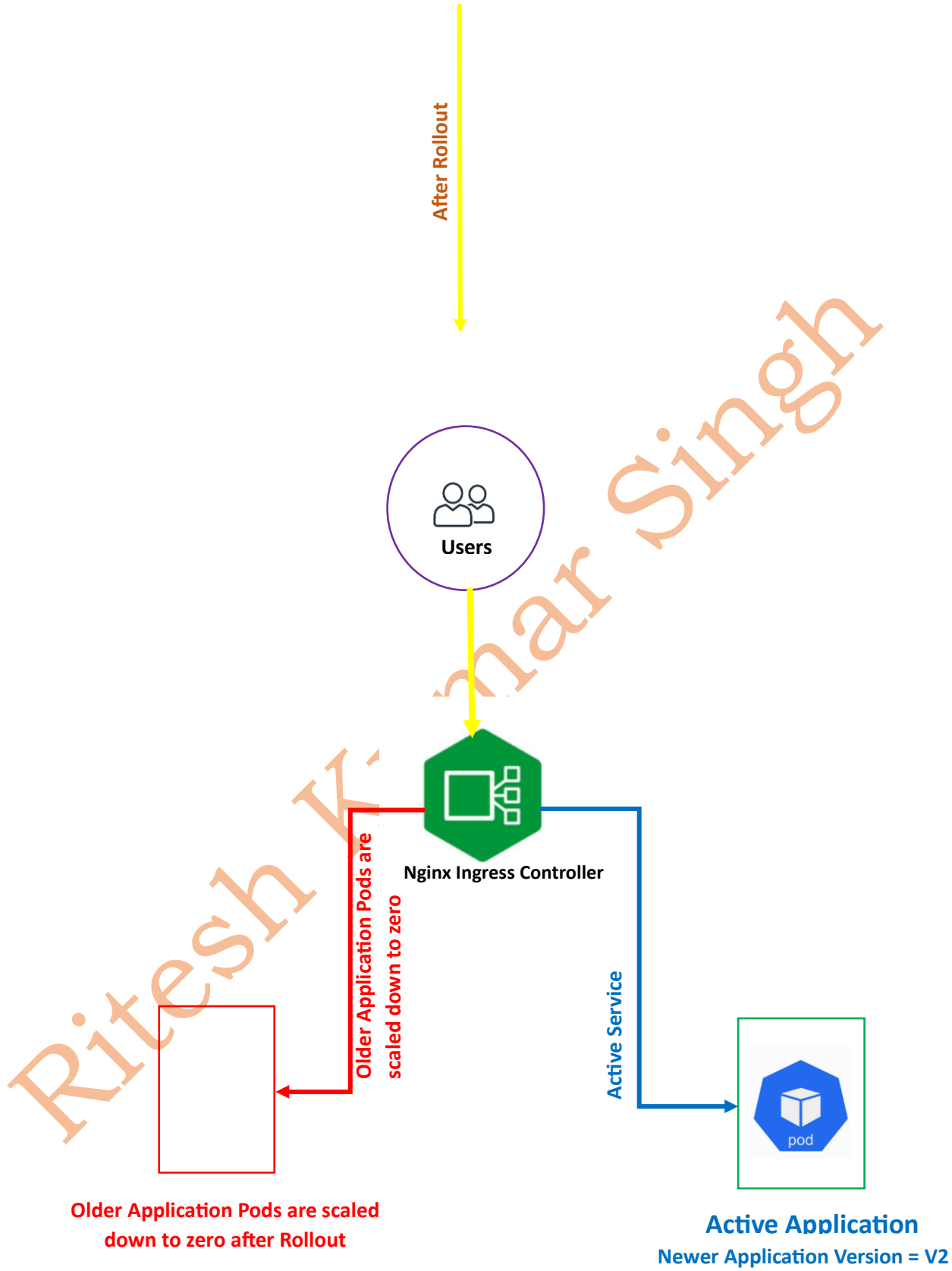

nexus.singhritesh85.com/#browse/browse:maven-snapshot
Sonatype Nexus Repository
OSS 3.68.1-02
Search components
user1
Sign out
Browse
Welcome
Search
Browse
Upload
Browse / maven-snapshot
HTML View
Advanced search...
com
example
bankapp
0.0.1-SNAPSHOT
0.0.1-20250408.070809-1
maven-metadata.xml
maven-metadata.xml.md5
maven-metadata.xml.sha1


sonarqube.singhritesh85.com/projects
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
My Favorites
All
Filters
Quality Gate
Passed
Failed
Reliability (Bugs)
Security (Vulnerabilities)
Security Review (Security Hotspots)
Search by project name or key
1 project(s)
Perspective: Overall Status
Sort by: Name
Create Project
bankapp
Passed
Last analysis: 4 hours ago
Bugs
Vulnerabilities
Hotspots Reviewed
Code Smells
Coverage
Duplications
Lines
0
0
0.0%
11
0.0%
0.0%
471
Java, XML
1 of 1 shown
SonarQube™ technology is powered by SonarSource SA
Community Edition - v9.9.5 (build 90363) - LGPL v3 - Community - Documentation - Plugins - Web API



For a New Release Developer did some change in the Code as per the new Code in Bank App Dashboard instead of Goldencat it will display GoldenDuck. To deploy the new Release of the code I used Blue-Green deployment. In Blue-Green deployment the Blue (Active) represents the Previous version and Green (Preview) represents the newer version of the Application.

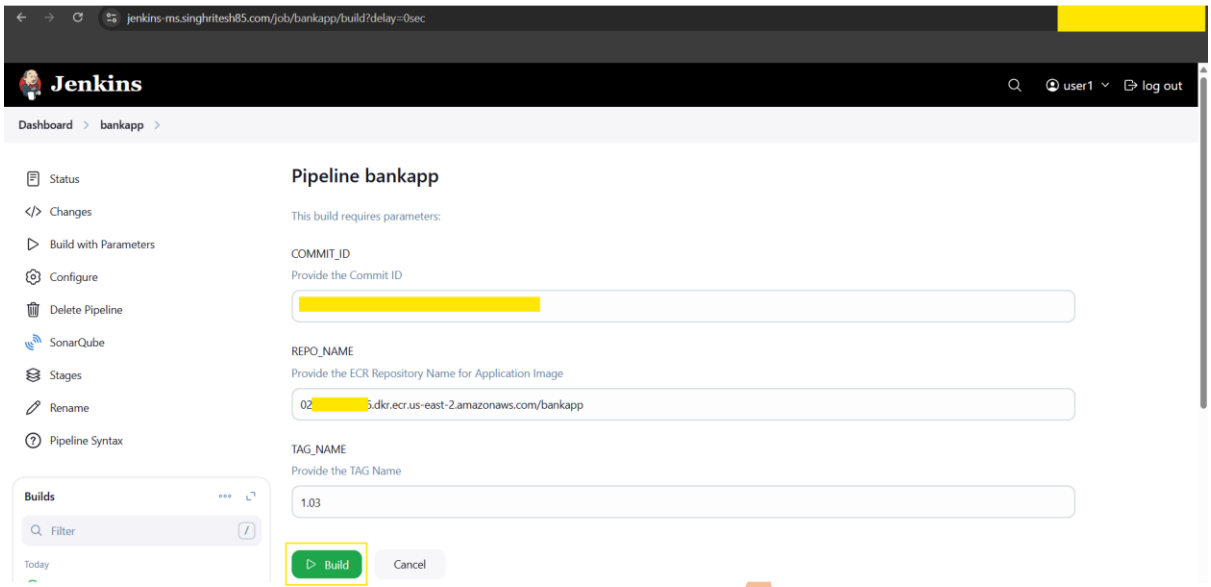




To deploy the newer version of the code I ran the Jenkins Job **bankapp** and changed the Service Type from ClusterIP to LoadBalancer then tested that the Application was working properly or not. Finally,

after testing the behaviour of Bank Application new Release I changed the service type from LoadBalancer to ClusterIP and Promoted the Rollout as discussed below.

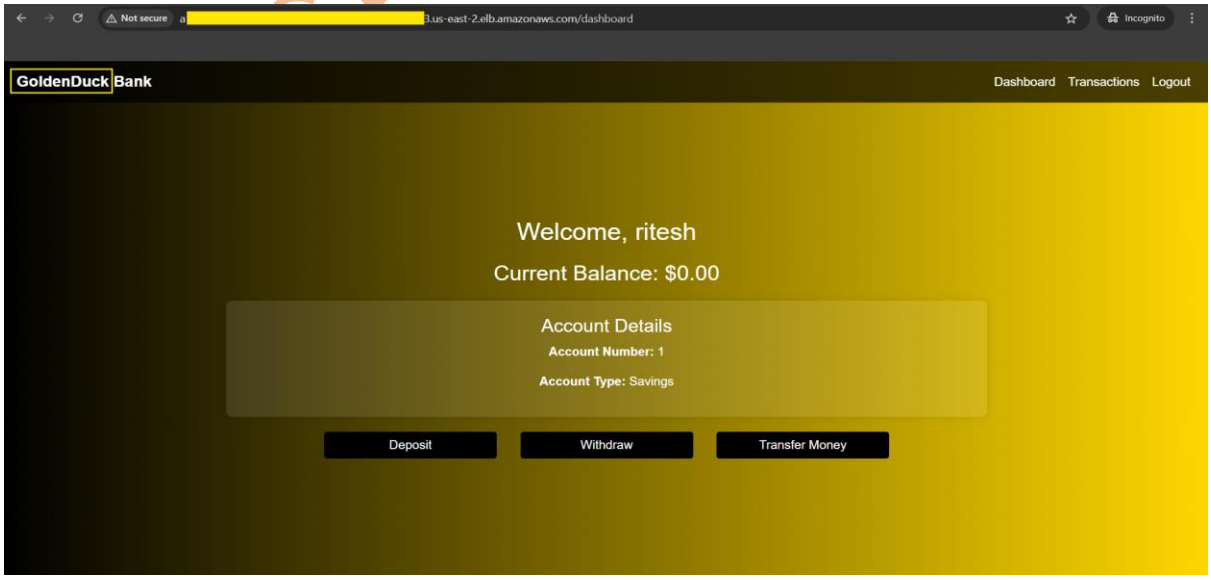
The string parameters which I used to run the Jenkins Job **bankapp** is as shown below.




```
[jenkins@ ~]$ kubectl get pods -n bankapp --context=bankapp
NAME                                READY   STATUS    RESTARTS   AGE
bankapp-folo-5                  g   1/1     Running   0          32s
bankapp-folo-7                  xb  1/1     Running   0          93m
```

Pod for newer release

```
[jenkins@ ~]$ kubectl get svc -n bankapp --context=bankapp
NAME                    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
bankapp-folo-active     ClusterIP   172.    .22     <none>        80/TCP    95m
bankapp-folo-preview    ClusterIP   172.    174     <none>        80/TCP    95m
[jenkins@ ~]$ kubectl edit svc bankapp-folo-preview -n bankapp --context=bankapp
service/bankapp-folo-preview edited
[jenkins@ ~]$ kubectl get svc -n bankapp --context=bankapp
NAME                    TYPE           CLUSTER-IP      EXTERNAL-IP                                                    PORT(S)        AGE
bankapp-folo-active     ClusterIP      172.    .22     <none>                                                         80/TCP         96m
bankapp-folo-preview    LoadBalancer   172.    174     a                                    8.us-east-2.elb.amazonaws.com   80:30782/TCP   96m
```



After testing the newer version of Application (Green) I found it was working properly. Finally promoted the rollout to make the changes as Active which is shown in the screenshot attached below.

kubectl get ro -n <namespace>

kubectl argo rollouts get rollout <rollout-name> -n <namespace>

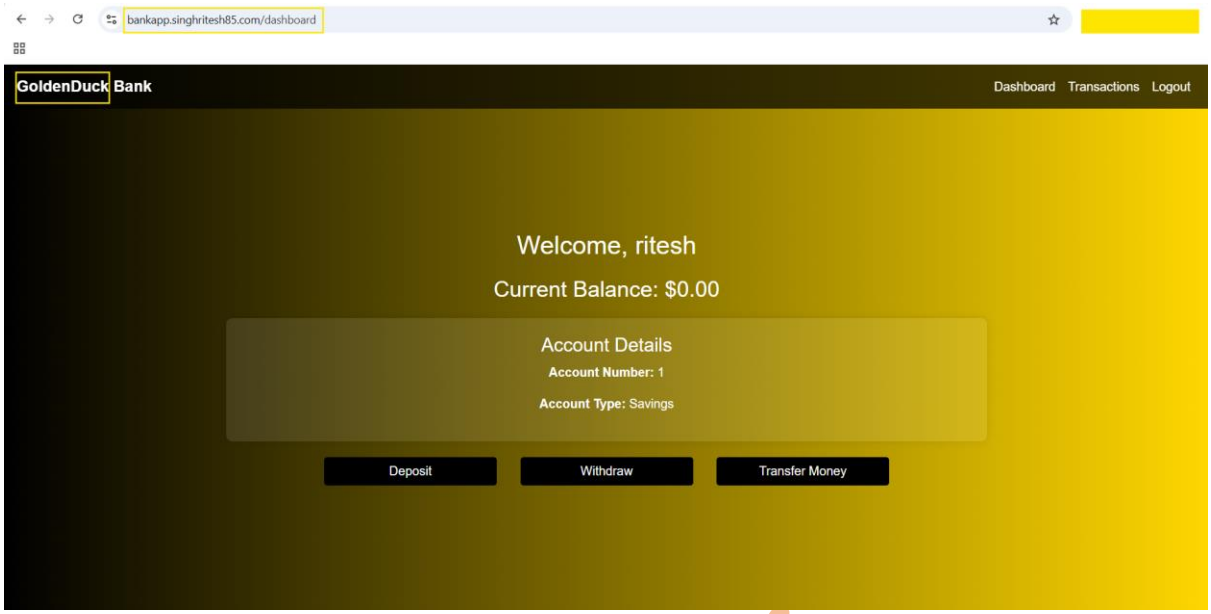
kubectl argo rollouts promote <rollout-name> -n <namespace>

```
[jenkins@                ~]$ kubectl config get-contexts
CURRENT   NAME      CLUSTER                                                AUTHINFO          NAMESPACE
*         bankapp   arn:aws:eks:us-east-2:02        6:cluster/eks-demo-cluster-dev   jenkins-bankapp
          mysql     arn:aws:eks:us-east-2:02        6:cluster/eks-demo-cluster-dev   jenkins-mysql
```

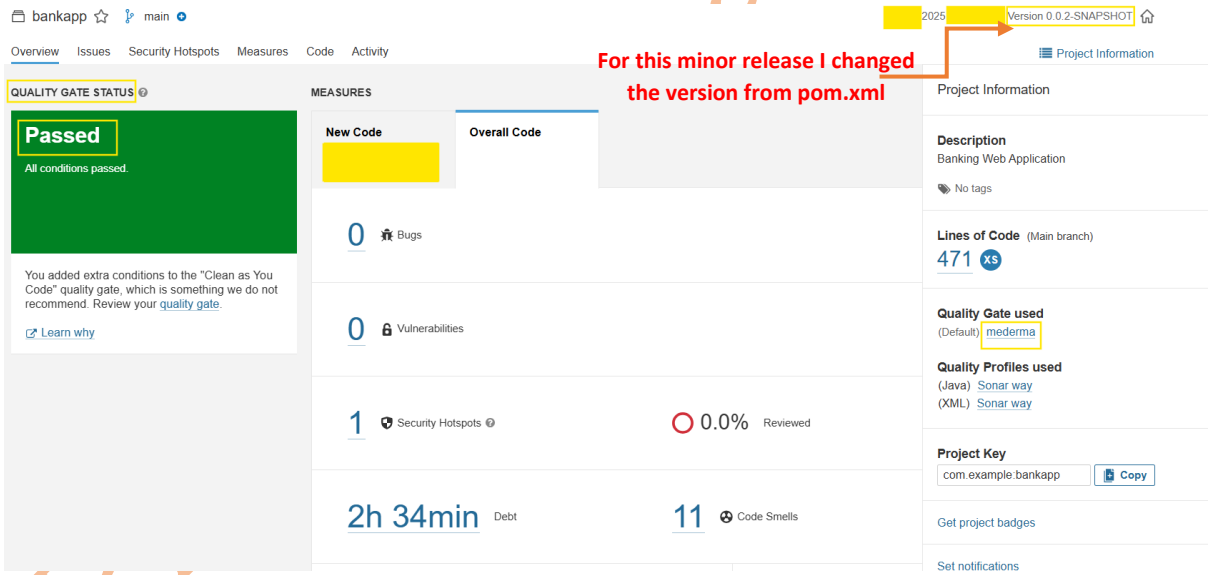
```
[jenkins@                ~]$ kubectl get ro -n bankapp
NAME           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
bankapp-folo   1         2         1            1           113m
[jenkins@                ~]$ kubectl argo rollouts get rollout bankapp-folo -n bankapp
Name:            bankapp-folo
Namespace:       bankapp
Status:          ॥ Paused
Message:         BlueGreenPause
Strategy:        BlueGreen
Images:          02        6.dkr.ecr.us-east-2.amazonaws.com/bankapp:1.02 (stable, active)
                 02        6.dkr.ecr.us-east-2.amazonaws.com/bankapp:1.05 (preview)
Replicas:
  Desired:       1
  Current:       2
  Updated:       1
  Ready:         1
  Available:     1

NAME                                    KIND        STATUS        AGE   INFO
⟳ bankapp-folo                          Rollout     ॥ Paused      113m
├──# revision:3
│  └──⧉ bankapp-folo-6        8         ReplicaSet  ✔Healthy     10m   preview
│     └──□ bankapp-folo-6        8-w    x  Pod      ✔Running     10m   ready:1/1
├──# revision:2
│  └──⧉ bankapp-folo-5        8         ReplicaSet  • ScaledDown  20m
└──# revision:1
   └──⧉ bankapp-folo-7        c         ReplicaSet  ✔Healthy     113m  stable,active
      └──□ bankapp-folo-7        c-g    b  Pod      ✔Running     113m  ready:1/1
[jenkins@                ~]$ kubectl argo rollouts promote bankapp-folo -n bankapp
rollout 'bankapp-folo' promoted
```

```
[jenkins@                ~]$ kubectl get all -n bankapp
NAME                              READY   STATUS    RESTARTS   AGE
pod/bankapp-folo-                 1/1     Running   0          

NAME                              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/bankapp-folo-active       ClusterIP   172.            <none>        80/TCP    
service/bankapp-folo-preview      ClusterIP   172.            <none>        80/TCP    

NAME                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/bankapp-folo-               1         1         1       
replicaset.apps/bankapp-folo-               0         0         0       
replicaset.apps/bankapp-folo-               0         0         0       
```

Finally, the newer release came out which was ready for others to use as shown in the screenshot attached below.



The SonarQube UI after this release (considering this as a minor release) is as shown in the screenshot attached below.



Installation of node-exporter and promtail had been done using the helm chart in the EKS Cluster as written below.

helm repo add prometheus-community https://*prometheus-community.github.io/helm-charts*

kubectl create ns node-exporter

helm install my-prometheus-node-exporter prometheus-community/prometheus-node-exporter --version 4.37.1 --set service.type=LoadBalancer -n node-exporter

```
[root@                ~]# helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
"prometheus-community" has been added to your repositories
[root@                ~]# kubectl create ns node-exporter
namespace/node-exporter created
[root@                ~]# helm install my-prometheus-node-exporter prometheus-community/prometheus-node-exporter --version 4.37.1 --set service.type=LoadBalanc
er -n node-exporter
NAME: my-prometheus-node-exporter
LAST DEPLOYED:    Apr          2025
NAMESPACE: node-exporter
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:
     NOTE: It may take a few minutes for the LoadBalancer IP to be available.
           You can watch the status of by running 'kubectl get svc -w my-prometheus-node-exporter'
  export SERVICE_IP=$(kubectl get svc --namespace node-exporter my-prometheus-node-exporter -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
  echo http://$SERVICE_IP:9100
[root@                ~]#
```

Below screenshot shows the Kubernetes Service which was created for node-exporter using the helm chart.

```
[root@                ~]# kubectl get svc -n node-exporter
NAME                          TYPE           CLUSTER-IP      EXTERNAL-IP                                                              PORT(S)          AGE
my-prometheus-node-exporter   LoadBalancer   172.      .63   a                                      5.us-east-2.elb.amazonaws.com   9100:31063/TCP   2m55s
```

The updated prometheus configuration file **/etc/prometheus/prometheus.yml** is as shown in the screenshot attached below.

```
- job_name: "EKS"
  static_configs:
    - targets: ["a                                      5.us-east-2.elb.amazonaws.com:9100"]
```

Then restarted the prometheus service as shown in the screenshot attached below.

```
[root@                ~]# systemctl restart prometheus.service
[root@                ~]# systemctl status prometheus.service
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: disabled)
   Active: active (running) since      2025                ; 9s ago
 Main PID: 2998 (prometheus)
```

I installed the promtail using the helm chart as written below. First, I cloned the helm chart present in GitHub Repo.

git clone https://github.com/singhritesh85/helm-chart-promtail.git

After cloning helm chart from GitHub, I updated the values.yaml file of promtail helm chart with Loki Servers Private IP Addresses as shown in the screenshot attached below.

kubectl create ns promtail && helm upgrade --install promtail ./helm-chart-promtail -f ./helm-chart-promtail/values.yaml -n promtail

kubectl get pods -n promtail --watch

```
# -- The log level of the Promtail server
# Must be reference in `config.file` to configure `server.log_level`
# See default config in `values.yaml`
logLevel: info
# -- The log format of the Promtail server
# Must be reference in `config.file` to configure `server.log_format`
# Valid formats: `logfmt, json`
# See default config in `values.yaml`
logFormat: logfmt
# -- The port of the Promtail server
# Must be reference in `config.file` to configure `server.http_listen_port`
# See default config in `values.yaml`
serverPort: 3101
# -- The config of clients of the Promtail server
# Must be reference in `config.file` to configure `clients`
# @default -- See `values.yaml`
clients:
  - url: http://10.[redacted].165:3100/loki/api/v1/push
  - url: http://10.[redacted].14:3100/loki/api/v1/push
  - url: http://10.[redacted].195:3100/loki/api/v1/push

# -- Configures where Promtail will save it's positions file, to resume reading after restarts.
# Must be referenced in `config.file` to configure `positions`
positions:
  filename: /run/promtail/positions.yaml
# -- The config to enable tracing
enableTracing: false
# -- A section of reusable snippets that can be reference in `config.file`.
# Custom snippets may be added in order to reduce redundancy.
```

```
[root@[redacted] ~]# git clone https://github.com/singhritesh85/helm-chart-promtail.git
```

```
[root@[redacted] ~]# kubectl create ns promtail && helm upgrade --install promtail ./helm-chart-promtail -f ./helm-chart-promtail/values.yaml -n promtail
```

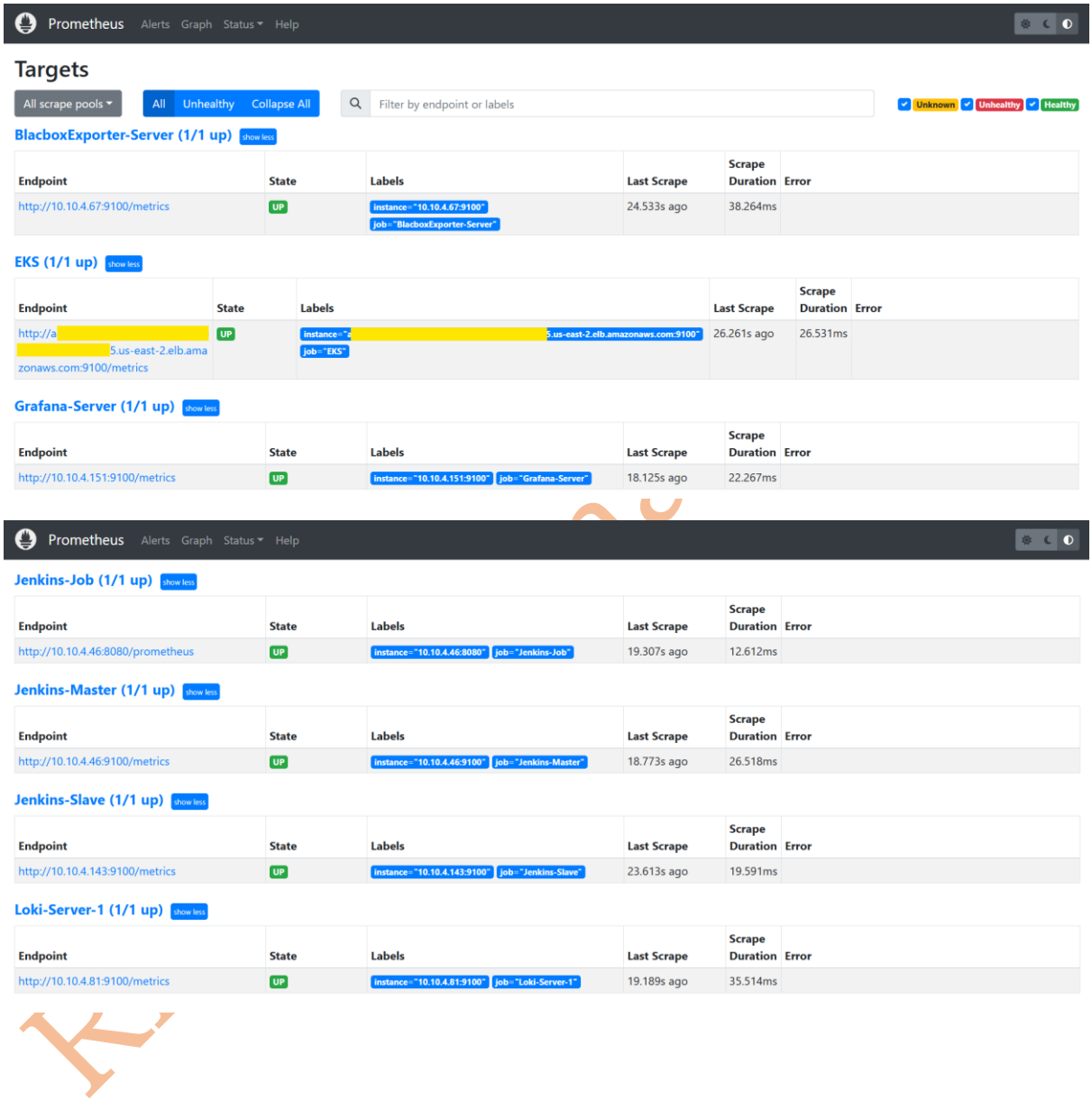
```
[root@[redacted] ~]# kubectl get pods -n promtail --watch
NAME                READY   STATUS    RESTARTS   AGE
promtail-c[redacted]r   1/1     Running   0          98s
promtail-x[redacted]d   1/1     Running   0          98s
```

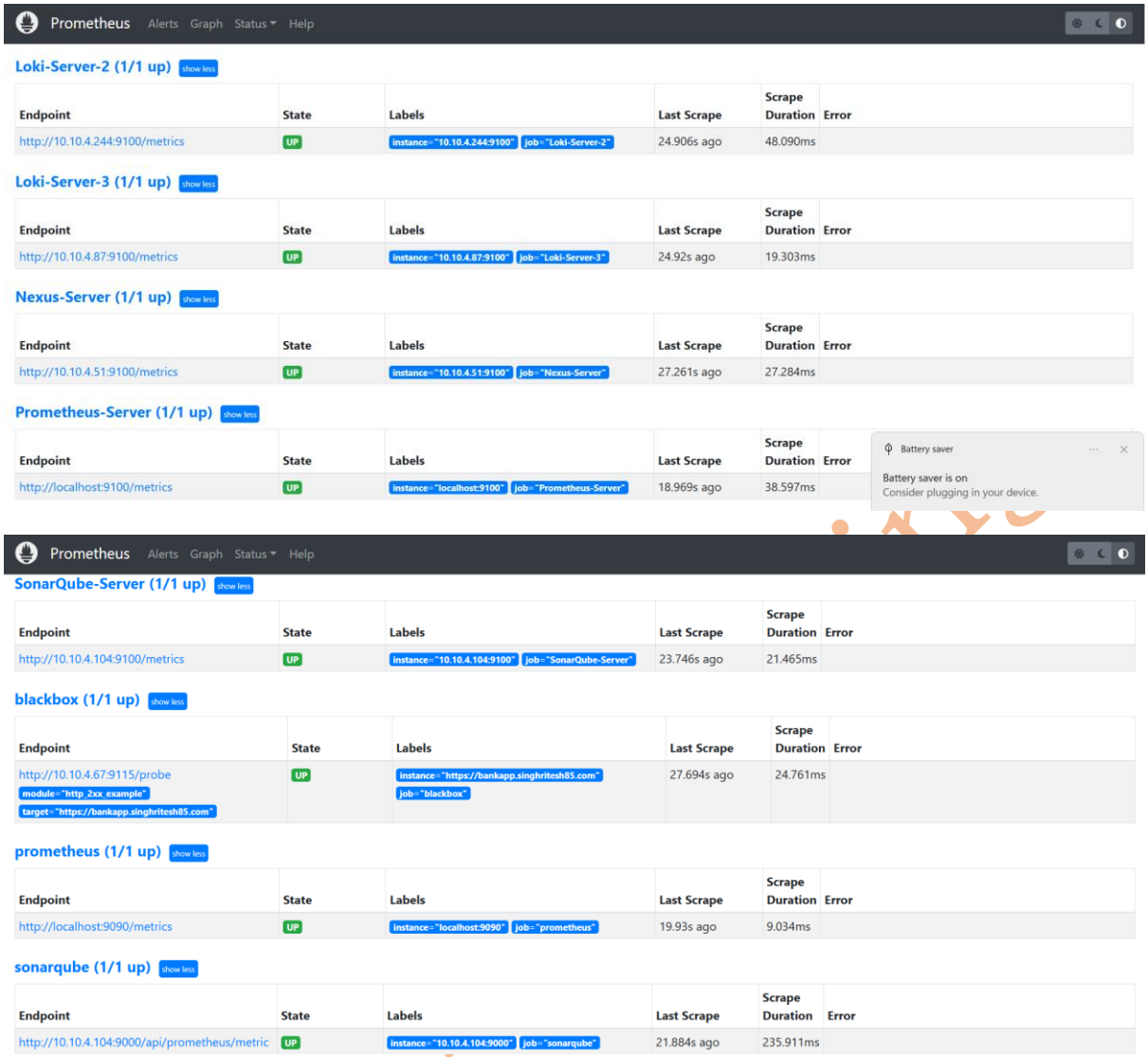
Monitoring Using Prometheus and Grafana and Log Aggregation using Loki

For Monitoring Tool I had used Prometheus and Grafana. To monitor SonarQube I had used SonarQube-Prometheus-Exporter which was installed using terraform at the path **/opt/sonarqube/extensions/plugins**. It was downloaded from the link https://github.com/dmeiners88/sonarqube-prometheus-exporter/releases/download/v1.0.0-SNAPSHOT-2018-07-04/sonar-prometheus-exporter-1.0.0-SNAPSHOT.jar. These steps had been covered in the terraform **user_data_sonarqube.sh**. It is basically a bootstrap script for SonarQube Server. For Monitoring Jenkins, you need to install the plugin **Prometheus metrics** and then restart Jenkins, these steps already been discussed at the starting. The configuration for prometheus had already been done in the terraform. I had taken sonarqube username and password as **admin** and

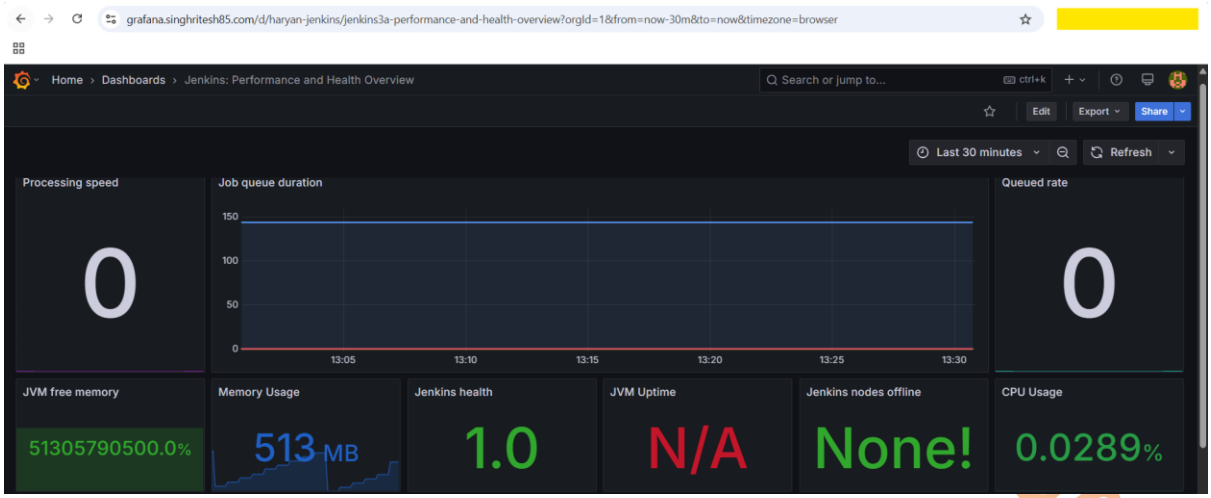
Admin123 respectively, you can choose as of your own choice and update the terraform script accordingly (Prometheus needs username and password to extract the metrics from SonarQube). I had provided the terraform script with this GitHub Repository. I had already integrated Prometheus and Loki as a data source for Grafana which was already discussed above.

Here I checked the prometheus console and I found all the Targets was UP as shown in the screenshot attached below.

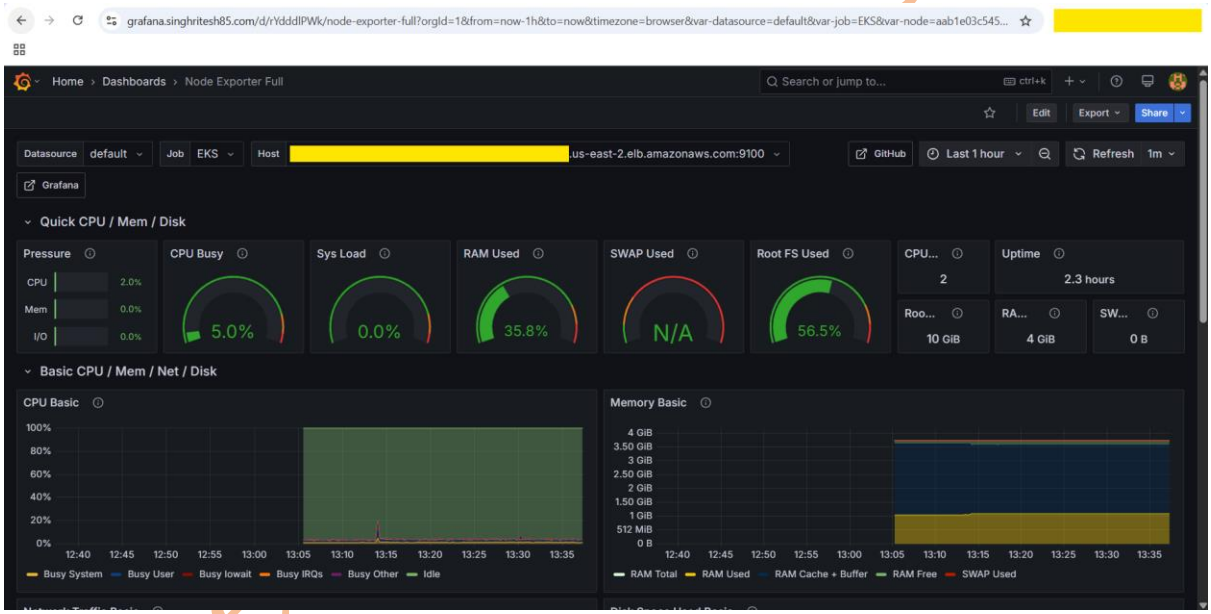




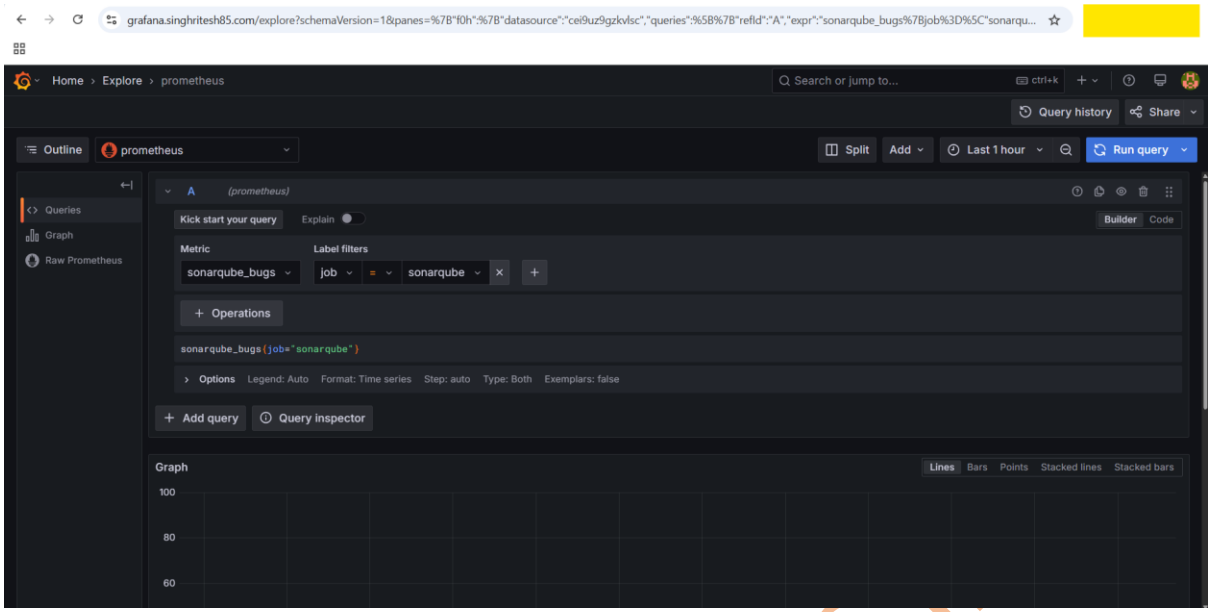
For Monitoring Jenkins Job using Prometheus I created the Grafana Dashboard using the Grafana ID **9964**.

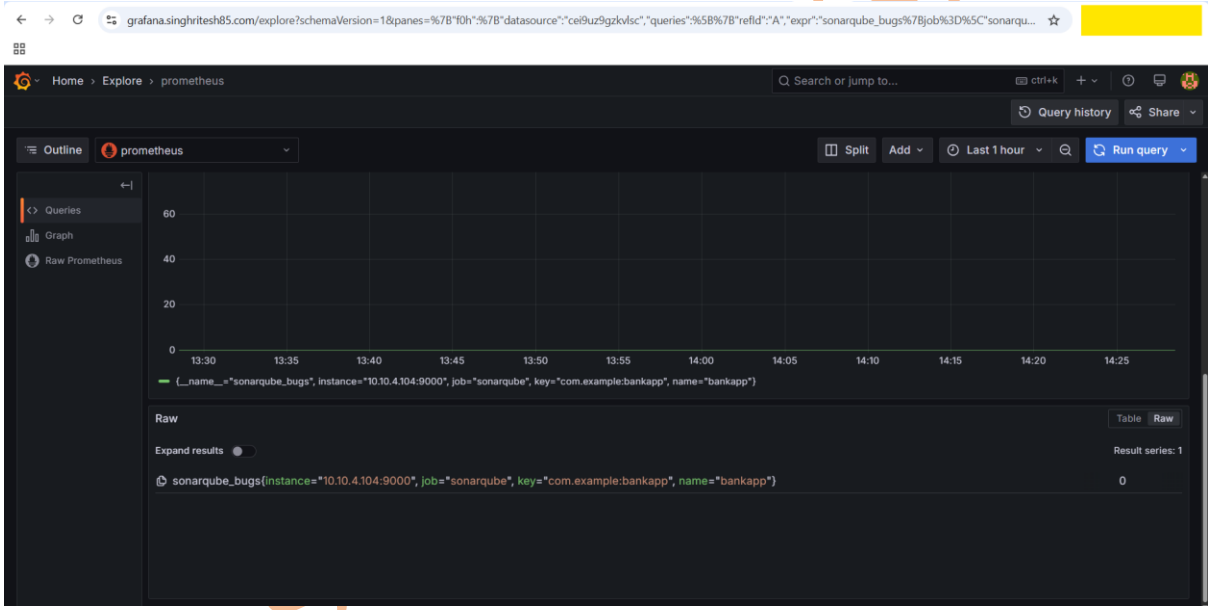


For Monitoring all the Servers and EKS Cluster health using the Node Exporter I used Grafana ID **1860**.

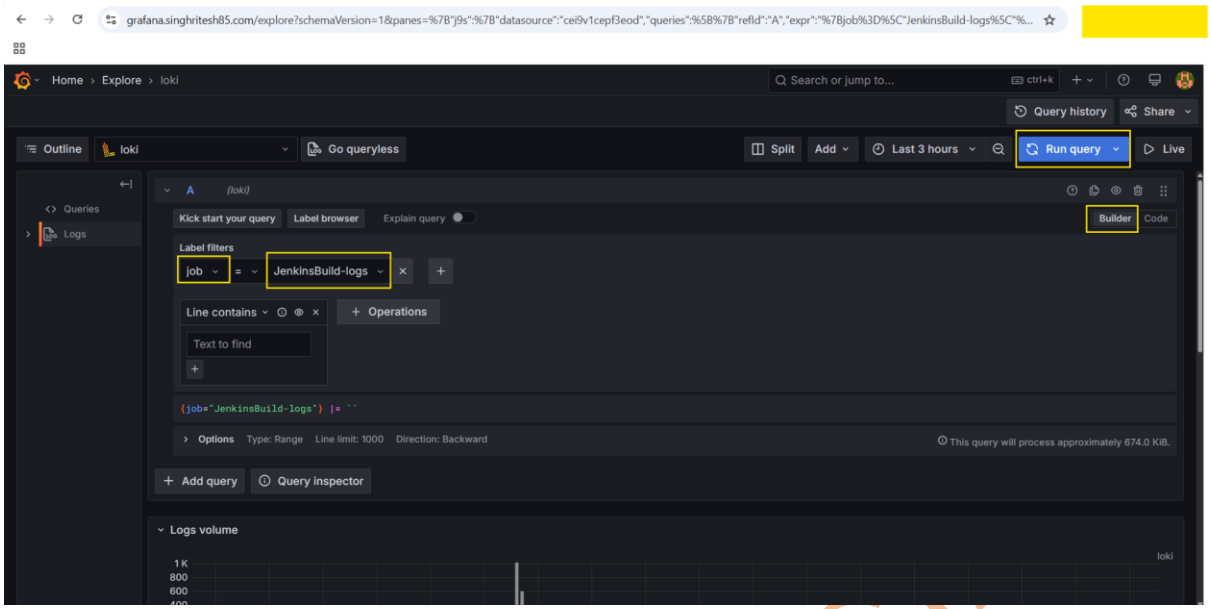


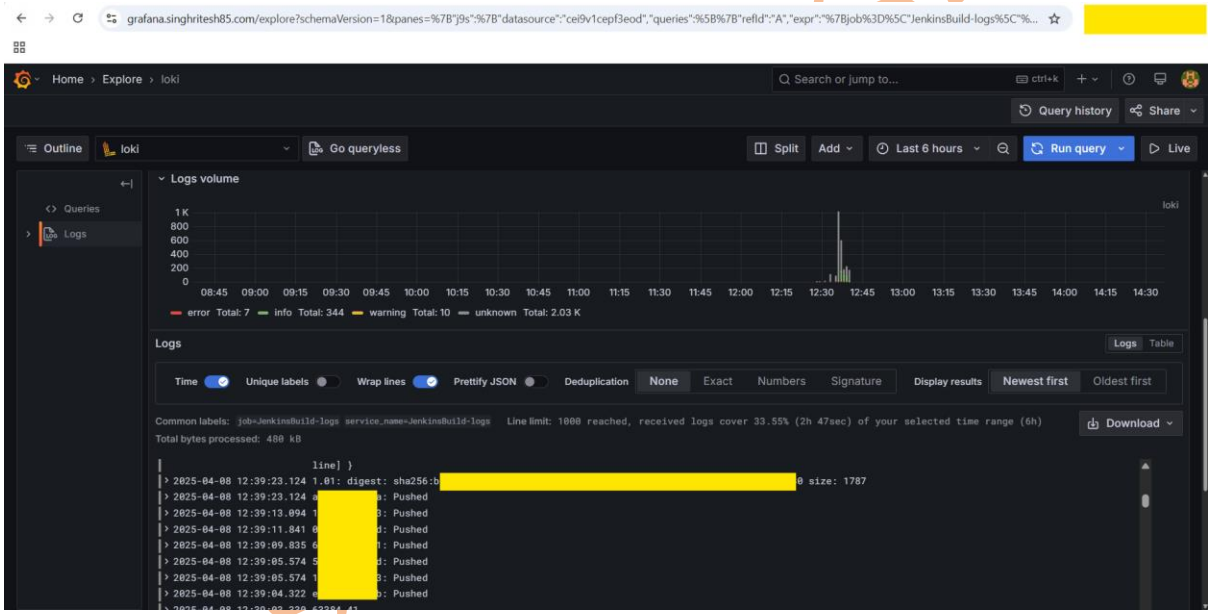
Grafana Metrics for SonarQube I started exploring as shown in the screenshot attached below.





Logs using Loki through Grafana I started exploring as shown in the screenshot attached below.





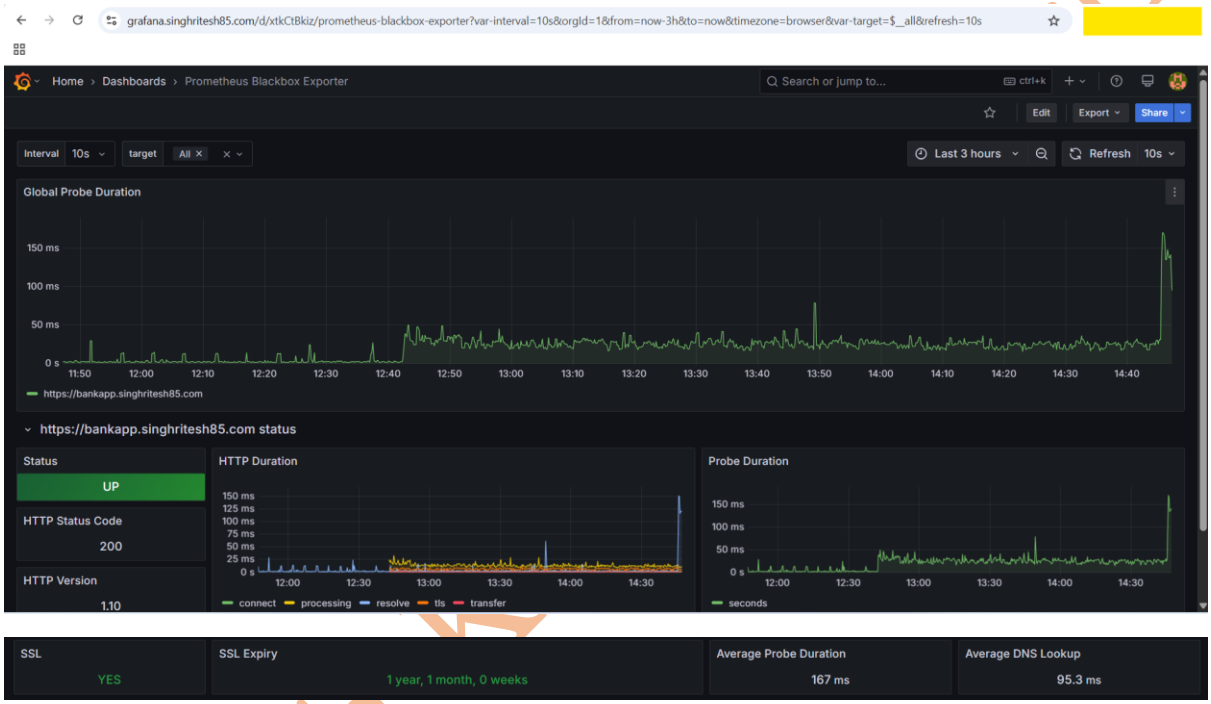
To achieve synthetic monitoring using Prometheus Blackbox Exporter I updated the **/etc/resolv.conf** file for Blackbox Exporter Server as shown in the screenshot attached below. I had used Google's Public DNS Server which is shown in the attached screenshot below.

```
[root@                    ~]# cat /etc/resolv.conf
; generated by /usr/sbin/dhclient-script
search us-east-2.compute.internal
options timeout:2 attempts:5
nameserver 8.8.8.8      #10.10.0.2
```

Finally, I was able to perform the synthetics monitoring on the Bankapp Application URL as shown in the screenshot attached below. Application URL **https://bankapp.singhritesh85.com** had been monitored using blackbox exporter.

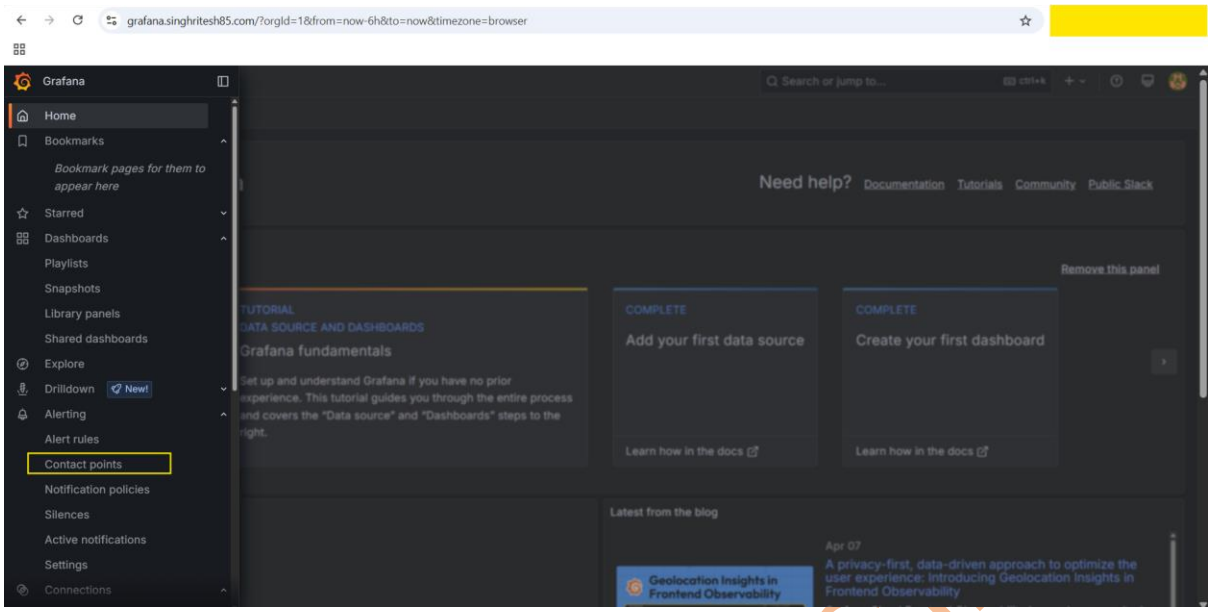
I had installed Blackbox Exporter on a different server and not on the Prometheus Server. **The module name** is monitor_website.yml present of the blackbox exporter server at the path (/opt/blackbox_exporter_linux_amd64/monitor_website.yml). Prometheus blackbox operator is used for endpoint monitoring (Synthetic Monitoring) across the protocol http, https, TCP and ICMP. In this project I am monitoring the Application URL **https://bankapp.singhritesh85.com** with the help of Prometheus Blackbox-Exporter. Prometheus blackbox exporter will send the metrics to Prometheus. For this project Prometheus acts as a DataSource for Grafana and send metrics to Grafana which we can see with the help of Charts and Graphs.

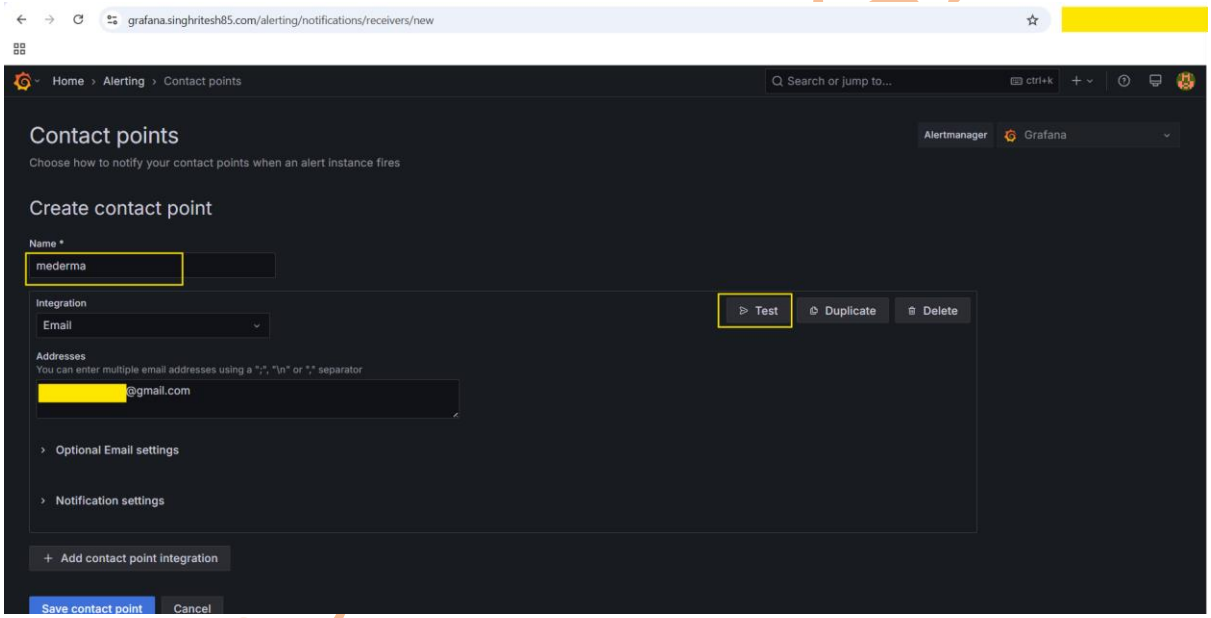
To create the Grafana Dashboard for Application URL Monitoring using blackbox exporter I had used the Grafana ID **7587** and below is the created Dashboard.

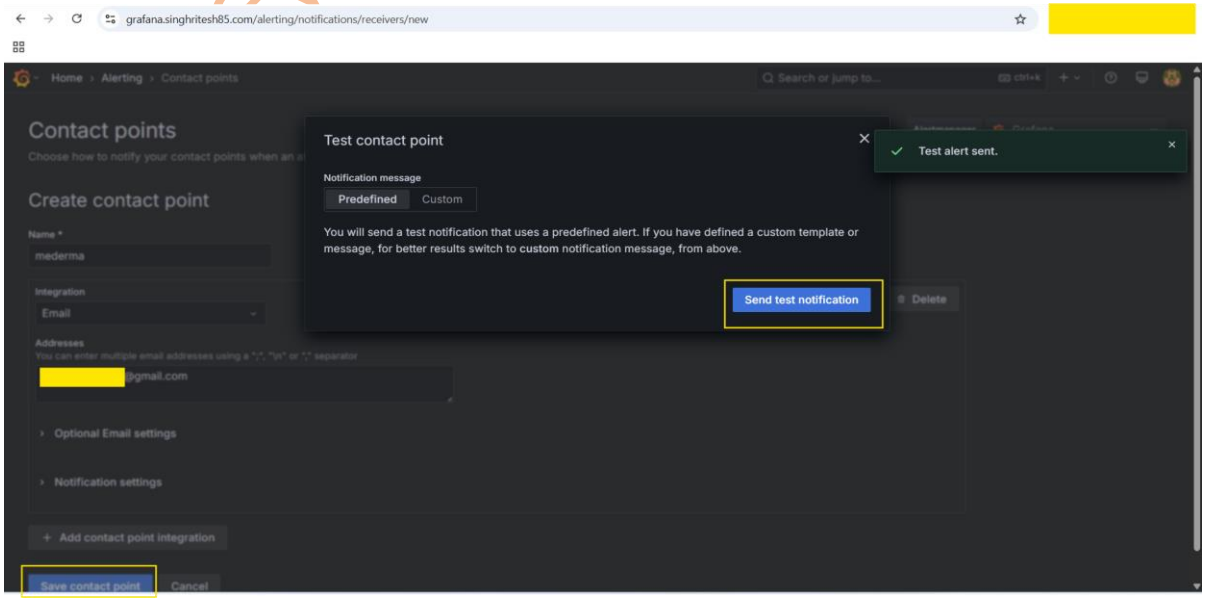


Configuration of Alerts in Grafana

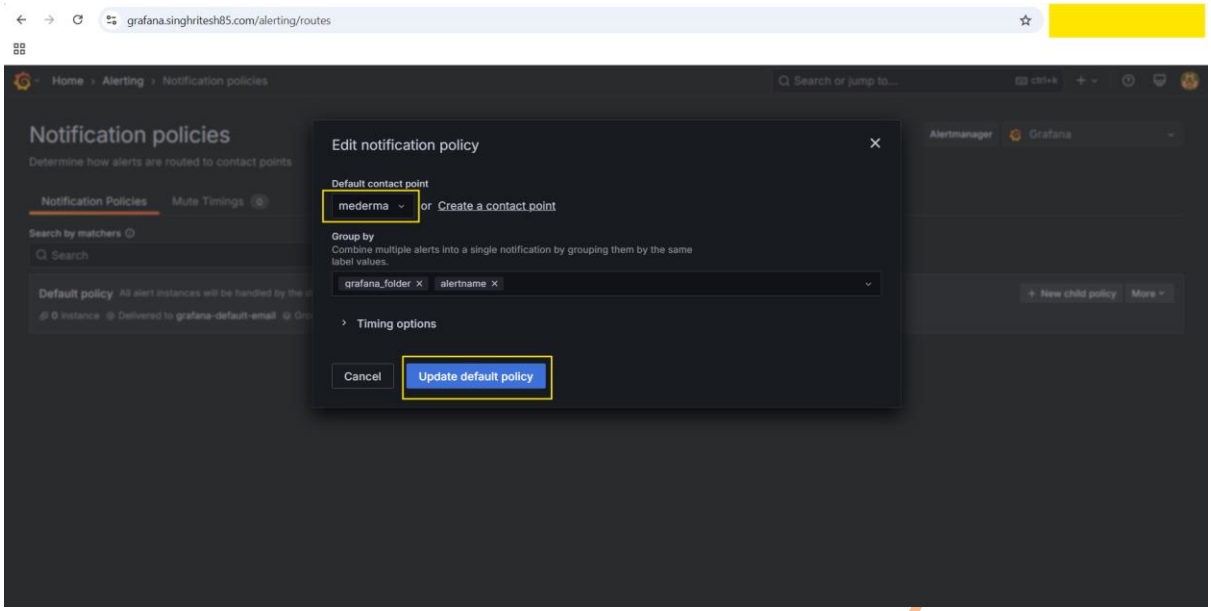
To configure Alerts in Grafana, first I created **contact points** with the Email ID and changed smtp settings in the configuration file **/etc/grafana/grafana.ini** of Grafana which I already discussed above. Here I had configured the contact points in Grafana UI as shown in the screenshot attached below.


grafana.singhritesh85.com/?orgId=1&from=now-6h&to=now&timezone=browser
Grafana
Home
Bookmarks
Bookmark pages for them to appear here
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
New!
Alerting
Alert rules
Contact points
Notification policies
Silences
Active notifications
Settings
Connections
Need help?
Documentation
Tutorials
Community
Public Slack
Remove this panel
Add your first data source
Create your first dashboard
Learn how in the docs
Learn how in the docs
Latest from the blog
Apr 07
A privacy-first, data-driven approach to optimize the user experience: Introducing Geolocation Insights in Frontend Observability

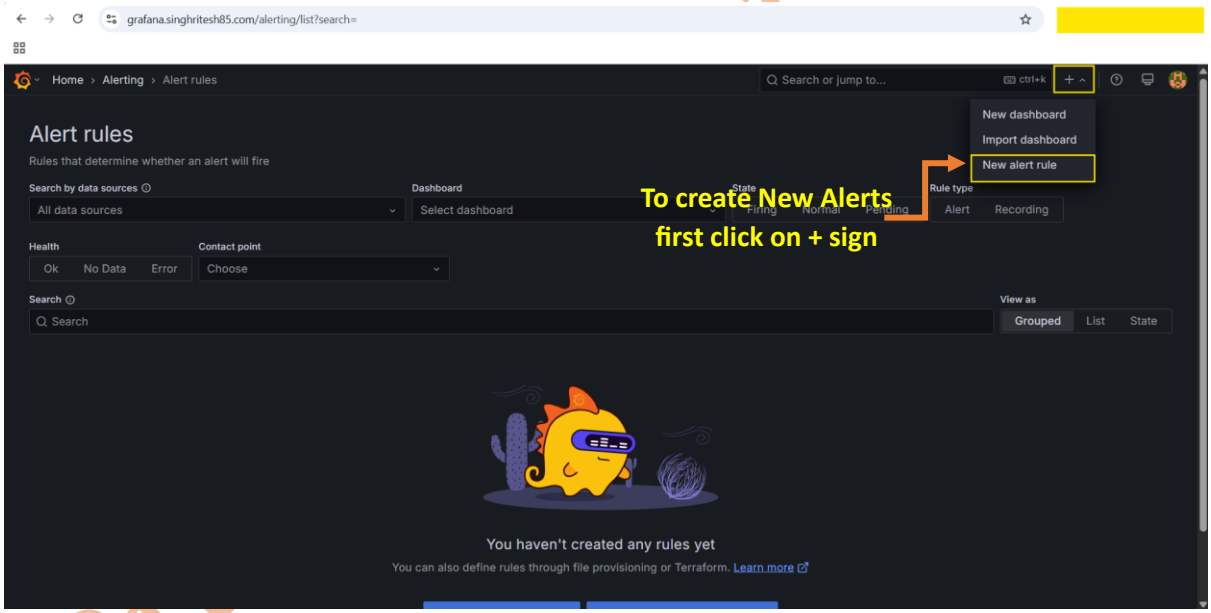

grafana.singhritesh85.com/alerting/notifications/receivers/new
Home
Alerting
Contact points
Contact points
Choose how to notify your contact points when an alert instance fires
Alertmanager
Grafana
Create contact point
Name *
mederma
Integration
Email
Test
Duplicate
Delete
Addresses
You can enter multiple email addresses using a ";", "\n" or "," separator
@gmail.com
Optional Email settings
Notification settings
Add contact point integration
Save contact point
Cancel

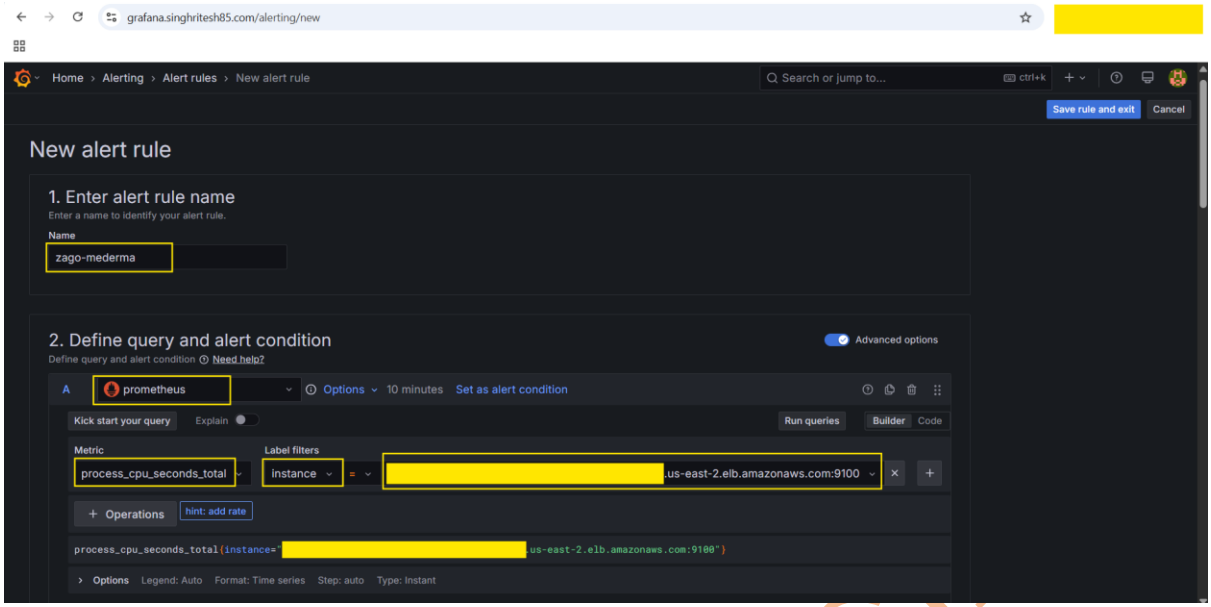

grafana.singhritesh85.com/alerting/notifications/receivers/new
Contact points
Create contact point
Test contact point
Notification message
Predefined
Custom
You will send a test notification that uses a predefined alert. If you have defined a custom template or message, for better results switch to custom notification message, from above.
Send test notification
Test alert sent.
Optional Email settings
Notification settings
Add contact point integration
Save contact point
Cancel

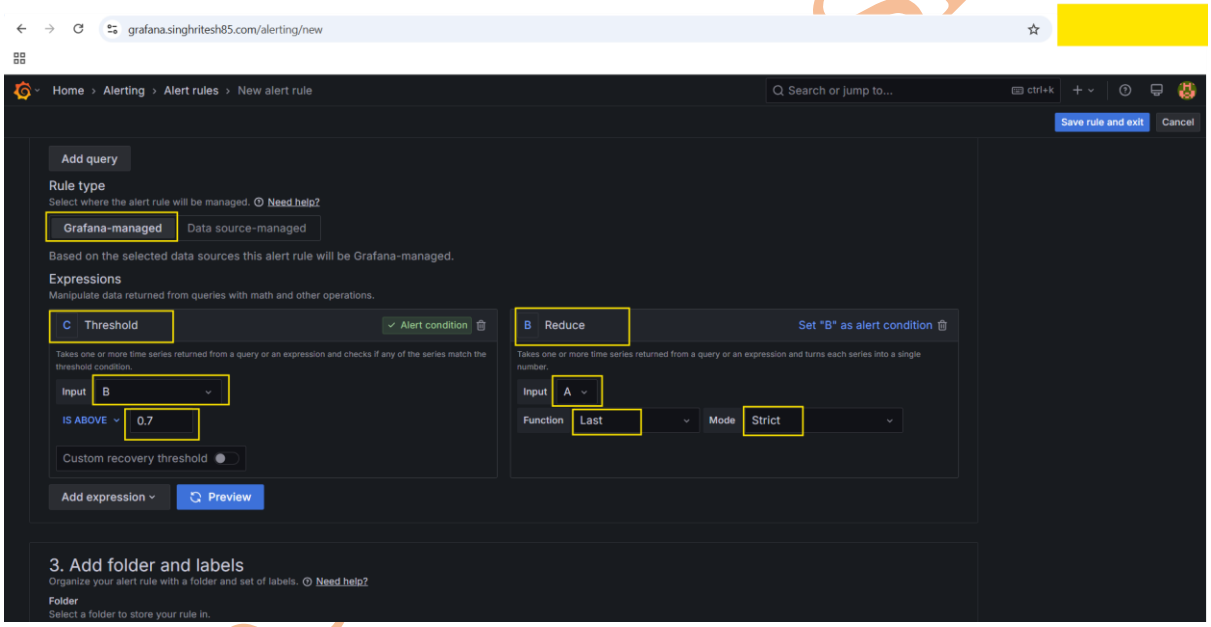
The Default **Notification Policy** had been changed as shown in the screenshot attached below.

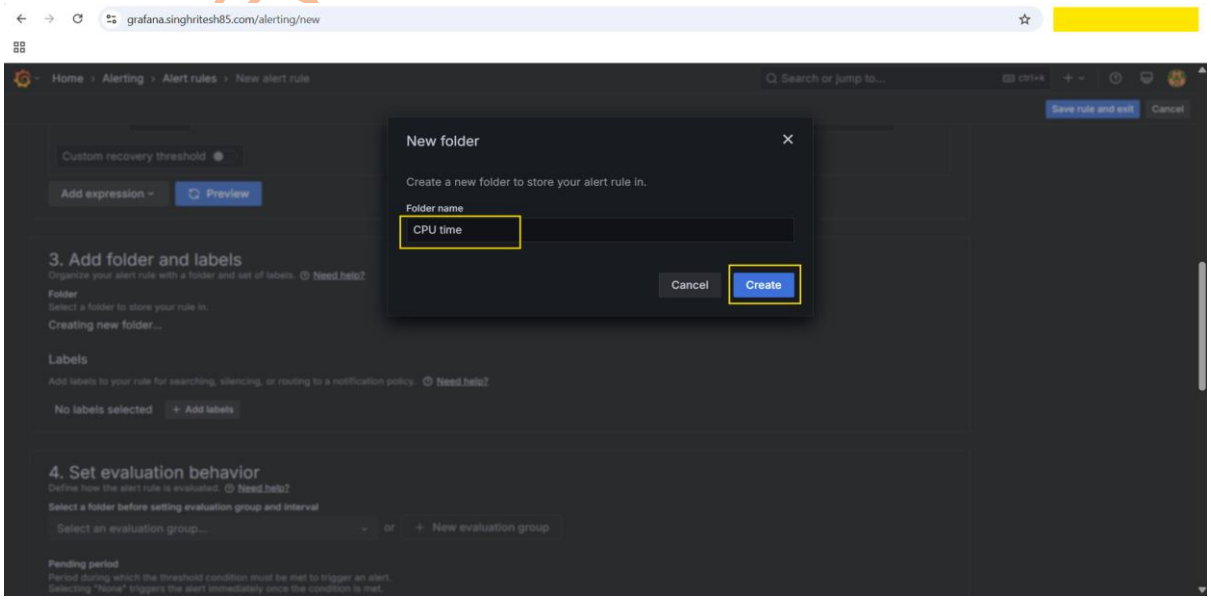


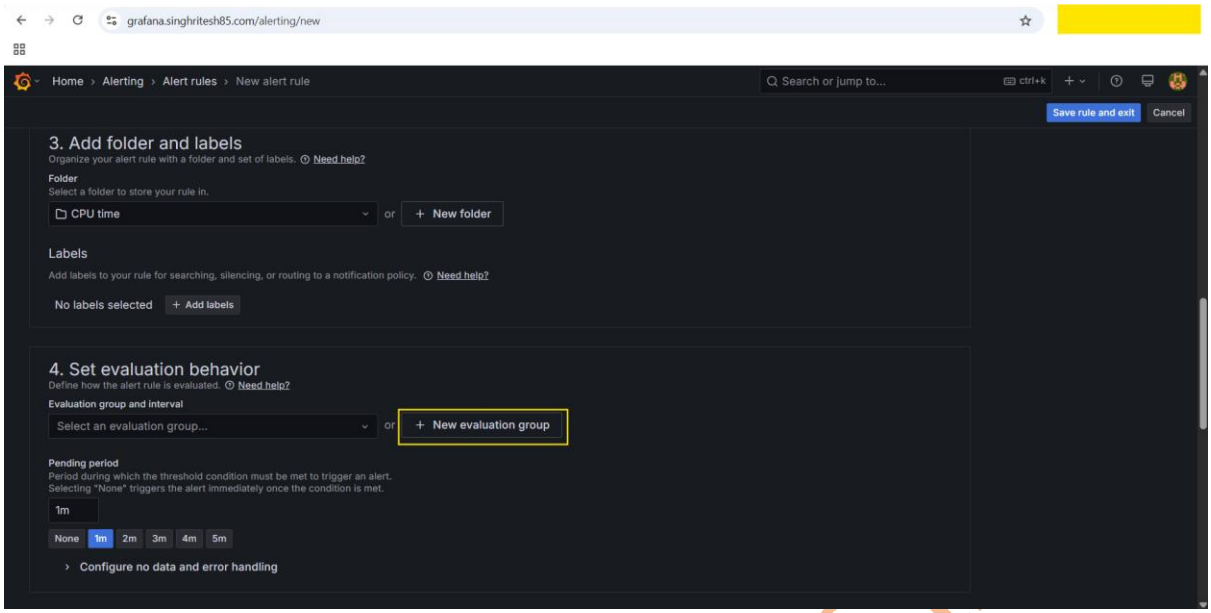
Configure **Alert Rule** as shown in the screenshot attached below.

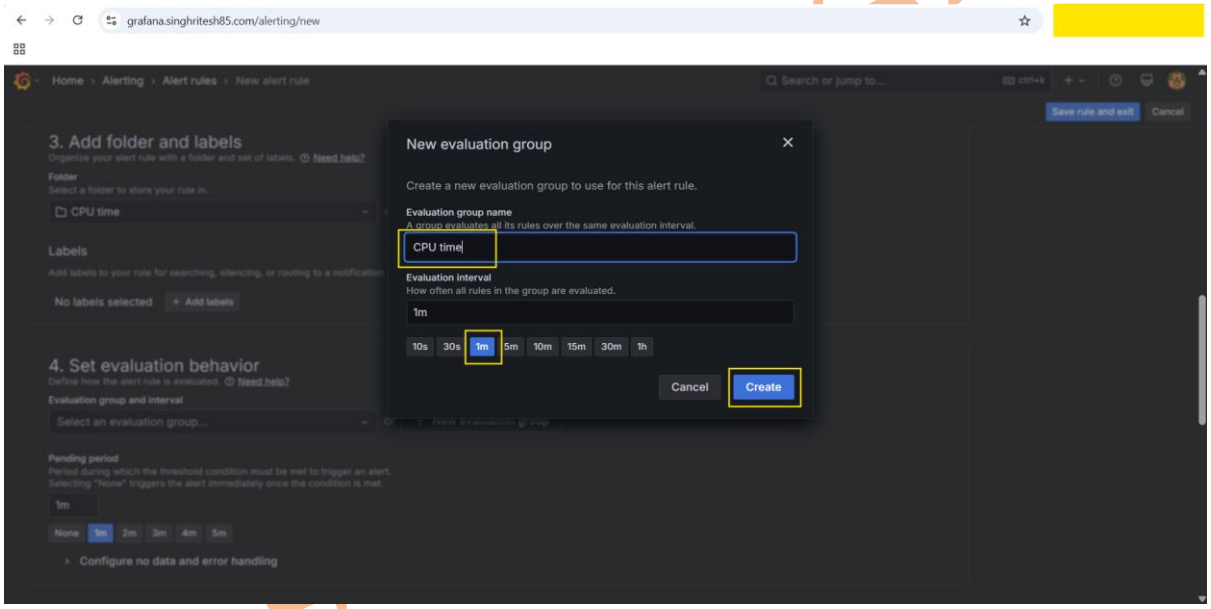


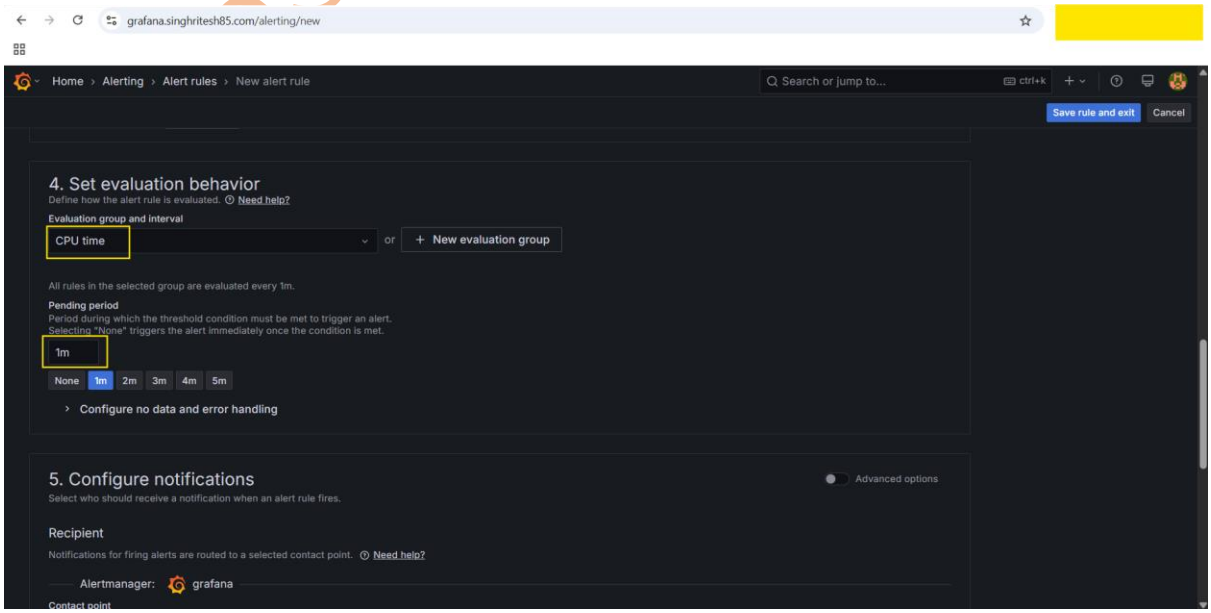

grafana.singhritesh85.com/alerting/new
Home › Alerting › Alert rules › New alert rule
Search or jump to...
Save rule and exit
Cancel
New alert rule
1. Enter alert rule name
Enter a name to identify your alert rule.
Name
zago-mederma
2. Define query and alert condition
Define query and alert condition Need help?
Advanced options
A
prometheus
Options
10 minutes
Set as alert condition
Kick start your query
Explain
Run queries
Builder
Code
Metric
process_cpu_seconds_total
Label filters
instance
us-east-2.elb.amazonaws.com:9100
Operations
hint: add rate
process_cpu_seconds_total{instance="
us-east-2.elb.amazonaws.com:9100"}
Options
Legend: Auto
Format: Time series
Step: auto
Type: Instant

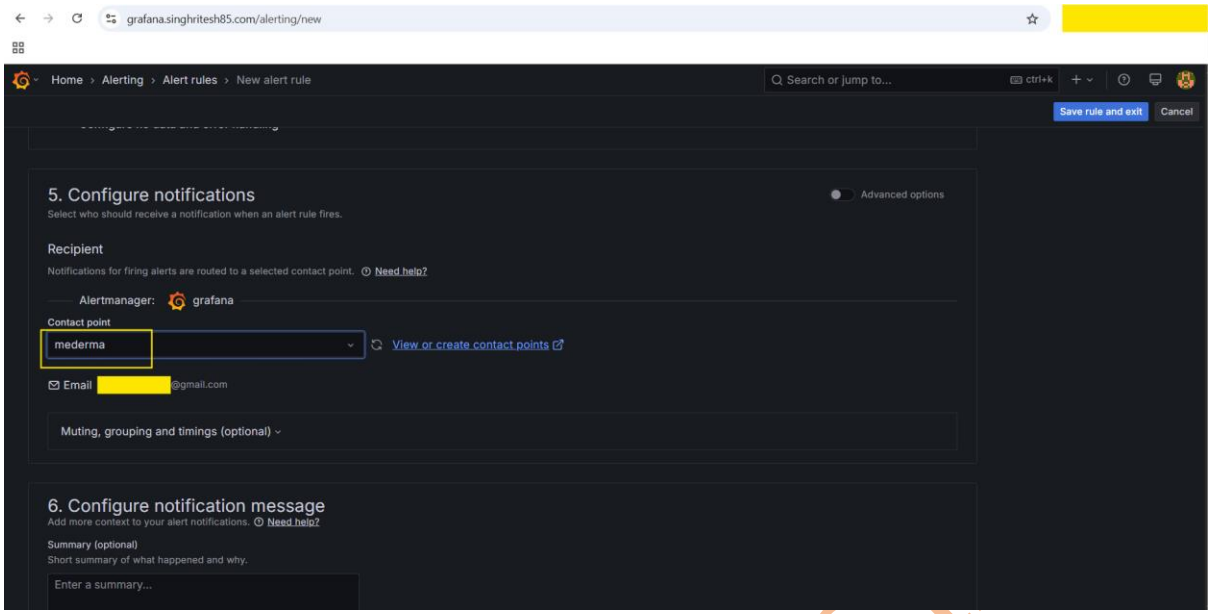

grafana.singhritesh85.com/alerting/new
Home › Alerting › Alert rules › New alert rule
Search or jump to...
Save rule and exit
Cancel
Add query
Rule type
Select where the alert rule will be managed. Need help?
Grafana-managed
Data source-managed
Based on the selected data sources this alert rule will be Grafana-managed.
Expressions
Manipulate data returned from queries with math and other operations.
C Threshold
Alert condition
Takes one or more time series returned from a query or an expression and checks if any of the series match the threshold condition.
Input
B
IS ABOVE
0.7
Custom recovery threshold
B Reduce
Set "B" as alert condition
Takes one or more time series returned from a query or an expression and turns each series into a single number.
Input
A
Function
Last
Mode
Strict
Add expression
Preview
3. Add folder and labels
Organize your alert rule with a folder and set of labels. Need help?
Folder
Select a folder to store your rule in.


grafana.singhritesh85.com/alerting/new
New folder
Create a new folder to store your alert rule in.
Folder name
CPU time
Cancel
Create
Custom recovery threshold
Add expression
Preview
3. Add folder and labels
Folder
Select a folder to store your rule in.
Creating new folder...
Labels
No labels selected
Add labels
4. Set evaluation behavior
Select a folder before setting evaluation group and interval
Select an evaluation group...
or
New evaluation group
Pending period

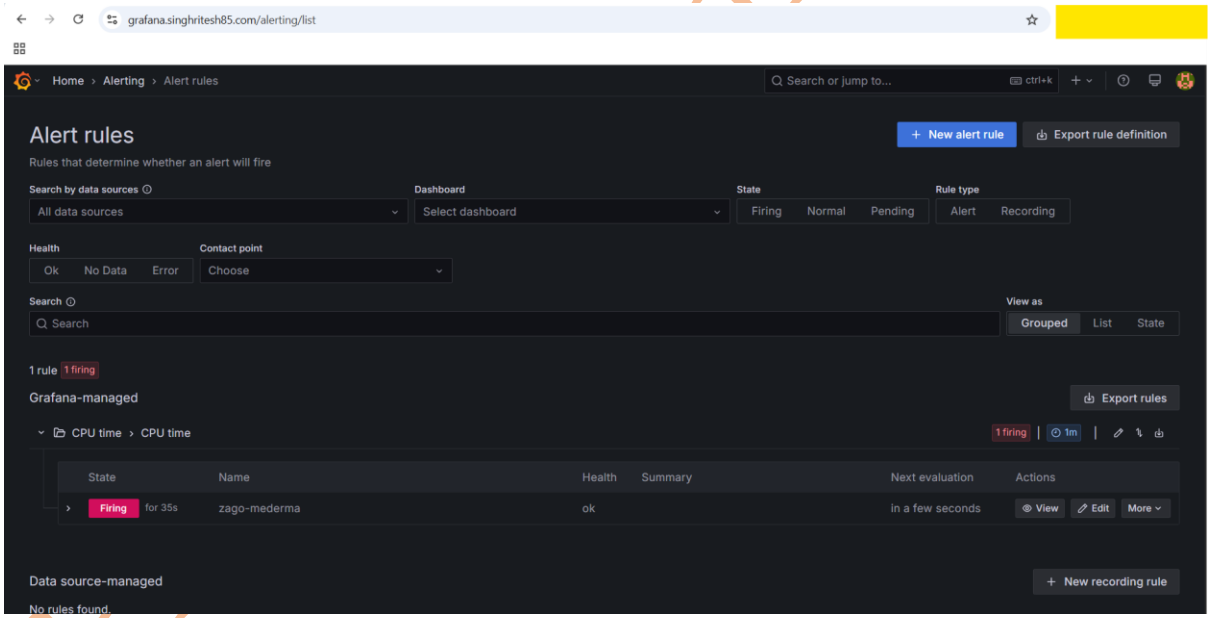

grafana.singhritesh85.com/alerting/new
Home › Alerting › Alert rules › New alert rule
Search or jump to...
Save rule and exit
Cancel
3. Add folder and labels
Organize your alert rule with a folder and set of labels. Need help?
Folder
Select a folder to store your rule in.
CPU time
or
+ New folder
Labels
Add labels to your rule for searching, silencing, or routing to a notification policy. Need help?
No labels selected
+ Add labels
4. Set evaluation behavior
Define how the alert rule is evaluated. Need help?
Evaluation group and interval
Select an evaluation group...
or
+ New evaluation group
Pending period
Period during which the threshold condition must be met to trigger an alert.
Selecting "None" triggers the alert immediately once the condition is met.
1m
None
1m
2m
3m
4m
5m
Configure no data and error handling


grafana.singhritesh85.com/alerting/new
New evaluation group
Create a new evaluation group to use for this alert rule.
Evaluation group name
A group evaluates all its rules over the same evaluation interval.
CPU time
Evaluation interval
How often all rules in the group are evaluated.
1m
10s
30s
1m
5m
10m
15m
30m
1h
Cancel
Create


grafana.singhritesh85.com/alerting/new
Home › Alerting › Alert rules › New alert rule
Search or jump to...
Save rule and exit
Cancel
4. Set evaluation behavior
Define how the alert rule is evaluated. Need help?
Evaluation group and interval
CPU time
or
+ New evaluation group
All rules in the selected group are evaluated every 1m.
Pending period
Period during which the threshold condition must be met to trigger an alert.
Selecting "None" triggers the alert immediately once the condition is met.
1m
None
1m
2m
3m
4m
5m
Configure no data and error handling
5. Configure notifications
Select who should receive a notification when an alert rule fires.
Advanced options
Recipient
Notifications for firing alerts are routed to a selected contact point. Need help?
Alertmanager:
grafana
Contact point

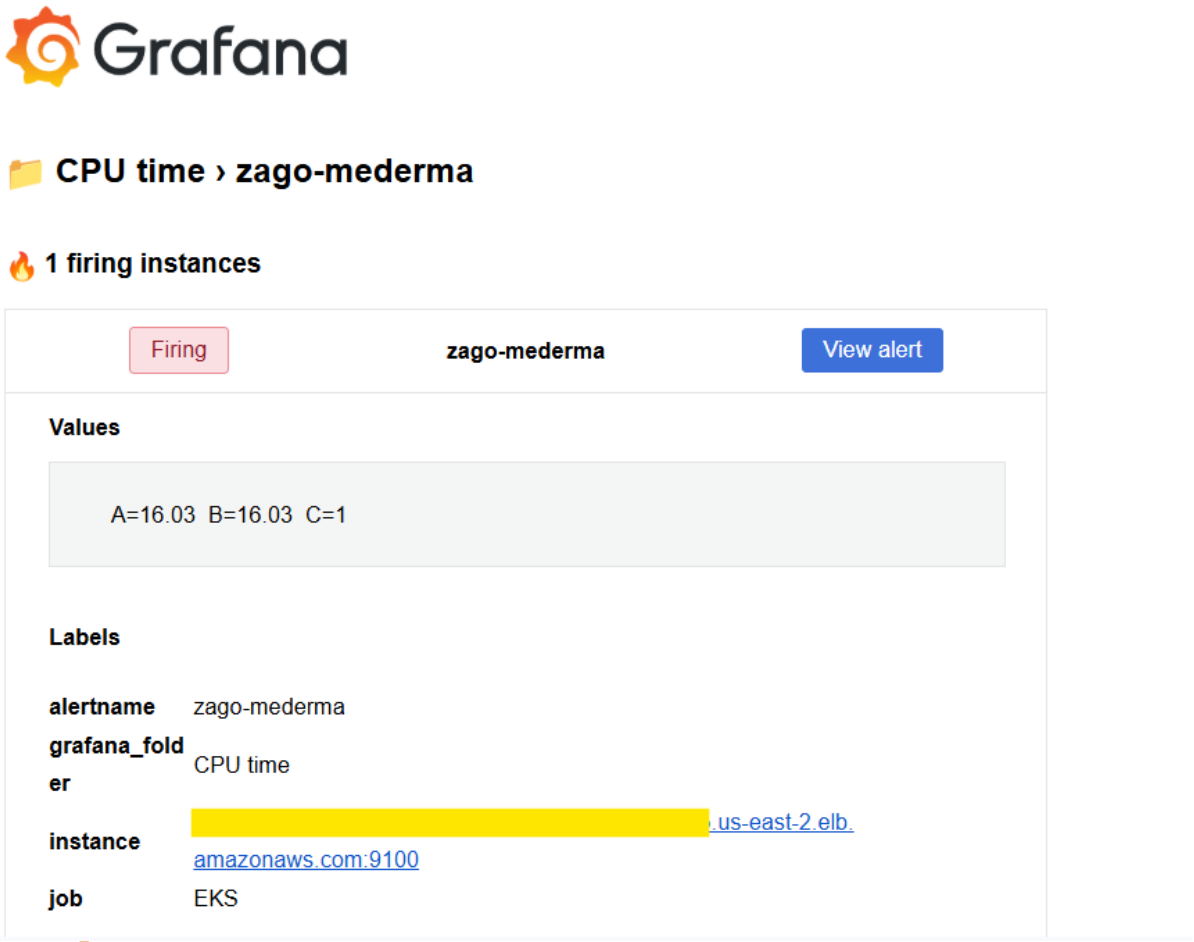


If the Alert Rule is in firing state after condition crosses the threshold condition, then Grafana console screenshot will be showing the same as shown in the screenshot attached below.



An Email was sent to the Email ID as shown in the screenshot attached below.


[FIRING:1] zago-mederma CPU time (.us-east-2.elb.amazonaws.com:9100 EKS)
Inbox ×
Grafana Alert for BankApp <@gmail.com>
to me
3:44 PM (1 minute ago)
Grafana
CPU time › zago-mederma
1 firing instances
Firing
zago-mederma
View alert
Values
A=16.03 B=16.03 C=1


Grafana
CPU time › zago-mederma
1 firing instances
Firing
zago-mederma
View alert
Values
A=16.03 B=16.03 C=1
Labels
alertname zago-mederma
grafana_folder CPU time
instance .us-east-2.elb.amazonaws.com:9100
job EKS

Source Code: - https://github.com/singhritesh85/Bank-App.git

GitHub Repo: - https://github.com/singhritesh85/DevOps-Project-BankApplication-BlueGreen-Deployment-MultiCloud.git

Helm Chart: - https://github.com/singhritesh85/helm-repo-for-Blue-Green-Deployment.git

https://github.com/singhritesh85/helm-repo-for-bitnami.git

Terraform Script: - https://github.com/singhritesh85/DevOps-Project-BankApplication-BlueGreen-Deployment-MultiCloud.git

Reference: - https://github.com/Goldencat98/Bank-App.git