

DevOps Project Netflix AWSandAzure



By Ritesh Kumar Singh

Email Address: - riteshkumarsingh9559@gmail.com

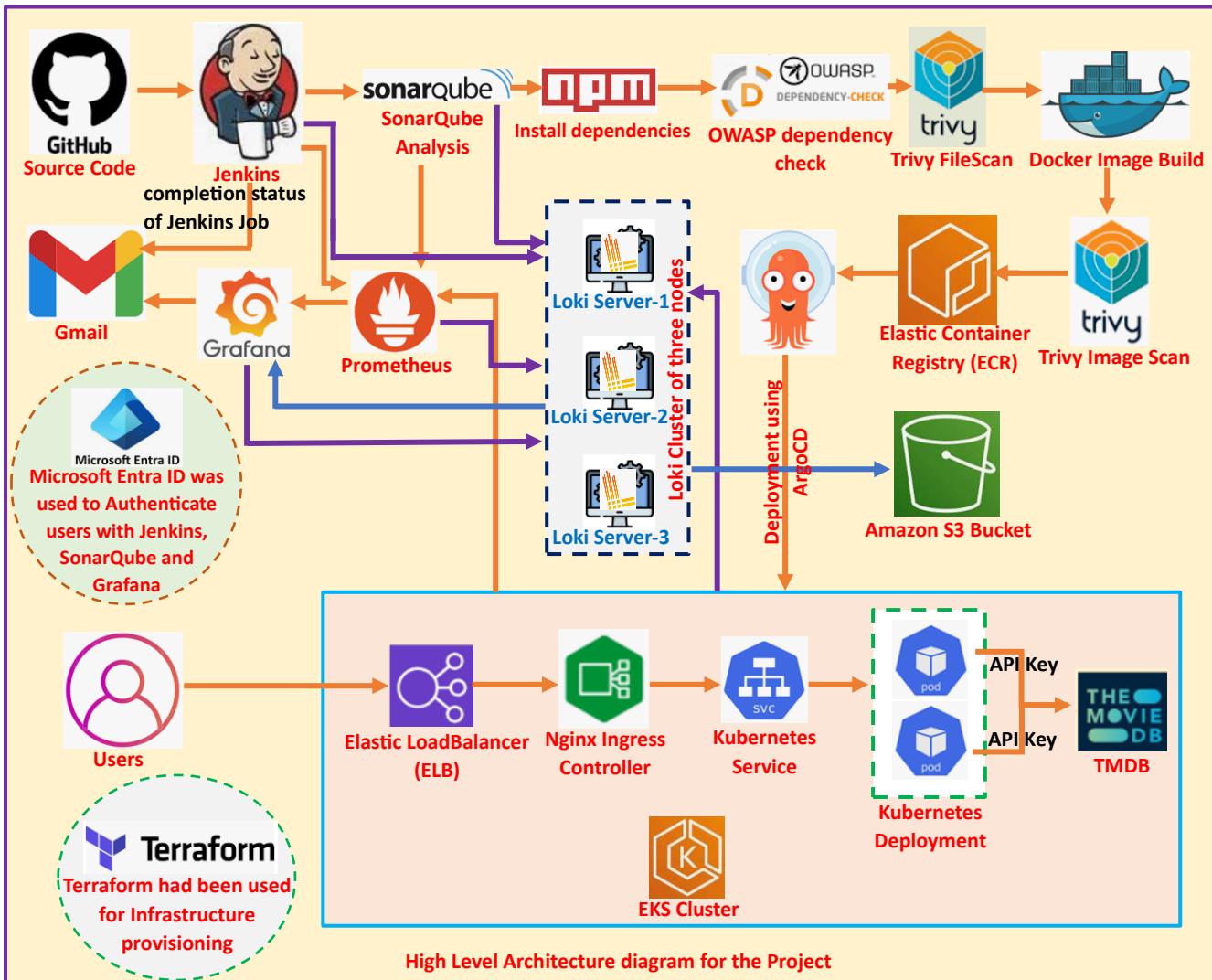
LinkedIn: - <https://www.linkedin.com/in/ritesh-kumar-singh-41113128b/>

GitHub: - <https://github.com/singhrites85>



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

DevOps Project Netflix AWS And Azure



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 NPM was used as the Build Tool as shown in the Architecture diagram above. OWASP dependency check had been performed to scan publicly known vulnerability and Trivy was used for FileScan and Docker Image Scan to identify and address known vulnerability hence enhancing the overall security. 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. For this project the source code was present in the GitHub Repository <https://github.com/singhritesh85/DevSecOps-Project.git>. The Kubernetes Pod will connect with TMDB (The Movie Database) using the API Key (the API Key will be used for Authentication). The successful completion or failure of Jenkins Job will be notified through the notification on Group Email ID. To Authenticate users in Jenkins, SonarQube and Grafana I used Azure Entra ID (formerly known as Azure Active Directory). Grafana will send notification to Group Email ID if an Alert encountered.

To validate the SSL Certificate generated from 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.

The screenshot shows two main parts:

- AWS Certificate Manager (Top):** Displays a newly issued SSL certificate for the domain `singhritesh85.com`. The certificate status is "Issued". It includes fields for Identifier, ARN (arn:aws:acm:us-east-2:0...), and Type (Amazon Issued). Below this, the "Domains (1)" section shows a single CNAME record named `_fc...03.singhritesh85.co...m.` pointing to `_0b...d9.ng...gs.acm-...valiations.aws.`
- Azure DNS Zone (Bottom):** Shows the same CNAME record `_fc...03.singhritesh85.co...m.` with a TTL of 3600 and a Value of `_0b...d9.ng...gs.acm-...valiations.aws.` The left sidebar lists various management options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Settings, DNS Management, Recordsets (selected), DNSSEC, Monitoring, Alerts, Metrics, Automation, Help, and a search bar.

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

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

Add record set

singhritesh85.com

Name

.singhritesh85.com

Type

CNAME – Link your subdomain to another record

Alias record set ⓘ

TTL *

TTL unit

Hours

Alias

SonarQube-1 3.us-east-2.elb.amazon

Add

Cancel

 Give feedback

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

The screenshot shows the AWS Route 53 Recordsets page for the domain `singhritesh85.com`. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Settings, DNS Management, Records (selected), DNSSEC, Monitoring, Alerts, Metrics, Automation, Help, and a search bar.

		SOA	TTL	
<code>_fc-[REDACTED]3</code>	CNAME	3600	<code>0-[REDACTED].gs.</code>	<code>acm-validations.aws.</code>
<code>grafana</code>	CNAME	3600	<code>Grafana-[REDACTED].us-east-2.elb.amazonaws.com</code>	
<code>jenkins-ms</code>	CNAME	3600	<code>jenkins-ms-[REDACTED].us-east-2.elb.amazonaws.com</code>	
<code>sonarqube</code>	CNAME	3600	<code>SonarQube-[REDACTED].us-east-2.elb.amazonaws.com</code>	

A large orange arrow points from the Jenkins entry to the Jenkins Metrics section of the Jenkins UI screenshot below.

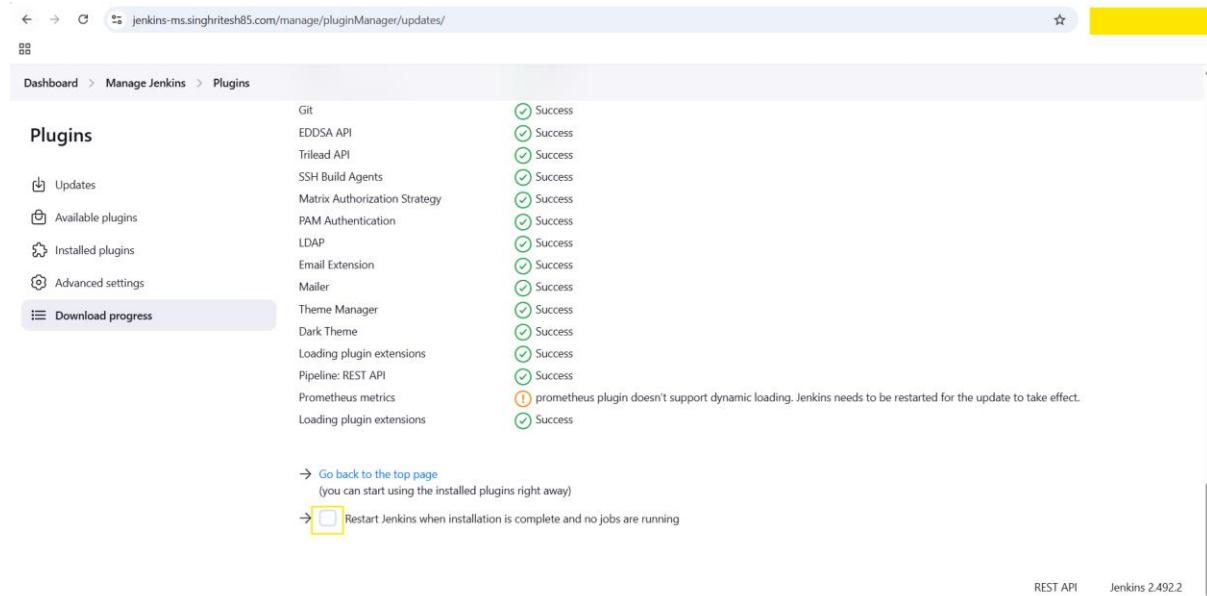
It is possible to monitor Jenkins Job using Prometheus and Grafana and for that I installed **Prometheus Metrics** plugin in Jenkins, after its installation I restarted the Jenkins as shown in the screenshot attached below.

The screenshot shows the Jenkins Manage Jenkins > Plugins page. The left sidebar has links for Dashboard, Manage Jenkins, Plugins, Available plugins (selected), Updates, Installed plugins, Advanced settings, and Download progress. A search bar at the top right contains the text `prometheus metrics`.

Install	Name	Released
<input checked="" type="checkbox"/>	<code>Prometheus metrics</code> [REDACTED]	1 mo 3 days ago

The Jenkins Prometheus Plugin expose an endpoint (default /prometheus) with metrics where a Prometheus Server can scrape.

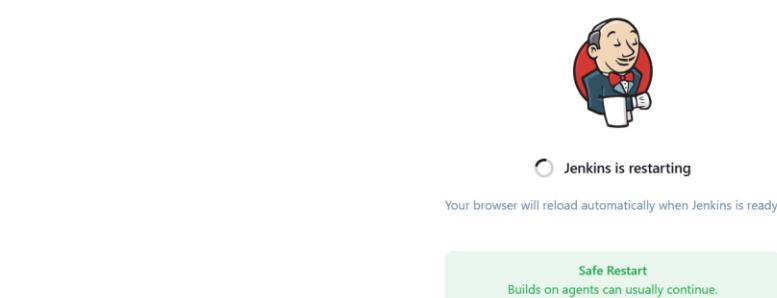
At the bottom right, there are links for REST API and Jenkins 2.492.2.



The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'Plugins' section. A table lists various Jenkins plugins and their installation status:

Plugin	Status
Git	Success
EDDSA API	Success
Trilead API	Success
SSH Build Agents	Success
Matrix Authorization Strategy	Success
PAM Authentication	Success
LDAP	Success
Email Extension	Success
Mailer	Success
Theme Manager	Success
Dark Theme	Success
Loading plugin extensions	Success
Pipeline: REST API	Success
Prometheus metrics	Info: prometheus plugin doesn't support dynamic loading. Jenkins needs to be restarted for the update to take effect.
Loading plugin extensions	Success

Below the table are two links: 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'. The Jenkins logo and version 'Jenkins 2.492.2' are visible at the bottom right.



In Grafana I created two data sources each for prometheus and loki as shown in the screenshot attached below.

grafana.singhritesh85.com/connections/datasources/edit/aegs8fafe603kc

prometheus

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://10.10.4.91:9090

Authentication

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/eegsa2bzsnr4c

loki

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: No Authentication

The screenshot shows the Grafana interface for managing data sources. The URL in the address bar is grafana.singhritesh85.com/connections/datasources/edit/eegsa2bzsnr4c. The page title is "Connections > Data sources > loki". Under the "Alerting" section, there is a button labeled "Manage alert rules in Alerting UI". Below it, the "Queries" section has a "Maximum lines" input set to 1000. The "Derived fields" section includes a "+ Add" button. A prominent green success message box at the bottom left says "Data source successfully connected." followed by "Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view." At the bottom right are "Delete" and "Save & test" buttons.

The data source for Grafana (Prometheus and Loki) had been configured as shown in the screenshot attached above.

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.

The screenshot shows the Jenkins Plugin Manager interface. The URL in the address bar is jenkins-ms.singhritesh85.com/manage/pluginManager/available. The page title is "Jenkins". The search bar contains "Entra ID". On the left, there are navigation links: "Updates", "Available plugins" (which is selected), "Installed plugins", and "Advanced settings". In the main area, there is a table with columns "Install", "Name", and "Released". One row is highlighted with a yellow border, showing "Install" and "Microsoft Entra ID (previously Azure AD)" under "Name". The "Released" date is "5 days 4 hr ago". The Jenkins logo is visible in the top right corner.

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.

Home > Default Directory | App registrations >

Register an application

Name
The user-facing display name for this application (this can be changed later).

Jenkins-login

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Default Directory only - Single tenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later.

By proceeding, you agree to the Microsoft Platform Policies [?>](#)

Register

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.

The screenshot shows the Jenkins 'Manage Jenkins > Security' configuration page. Under 'Security Realm', 'Azure Active Directory' is selected and highlighted with a yellow box. Other options like 'Client ID', 'Authentication Type (Client Secret selected)', and 'Tenant' are also visible.

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.

The screenshot shows the 'jenkins-login' application's 'Authentication' settings in the Microsoft Azure portal. Under 'Implicit grant and hybrid flows', the 'ID tokens (used for implicit and hybrid flows)' checkbox is selected and highlighted with a yellow box. Other options like 'Access tokens (used for implicit flows)' are also present.

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

[jenkins-login | API permissions](#)

Search Refresh Got feedback?

Overview Quickstart Integration assistant Diagnose and solve problems Manage Branding & properties Authentication Certificates & secrets Token configuration API permissions Expose an API App roles Owners Roles and administrators Manifest

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for Default Directory

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	...

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).

Home > Default Directory | App registrations > jenkins-login

[jenkins-login | API permissions](#)

Search Refresh Got feedback?

Overview Quickstart Integration assistant Diagnose and solve problems Manage Branding & properties Authentication Certificates & secrets Token configuration API permissions Expose an API App roles Owners Roles and administrators Manifest

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for Default Directory

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (5)				
Directory.Read.All	Application	Read directory data	Yes	Granted for Default Dire... ...
Group.Read.All	Application	Read all groups	Yes	Granted for Default Dire... ...
People.Read.All	Application	Read all users' relevant people lists	Yes	Granted for Default Dire... ...
User.Read	Delegated	Sign in and read user profile	No	Granted for Default Dire... ...
User.Read.All	Application	Read all users' full profiles	Yes	Granted for Default Dire... ...

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).

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.

The screenshot shows the Jenkins 'Configure Security' page under 'Manage Jenkins > Security'. It displays a grid of roles and permissions for 'Azure Active Directory Matrix-based security'. The columns represent roles like User/group, Overall, Credentials, Agent, Job, Run, View, SCM, and Metrics. The rows represent actions like View, Read, Delete, Create, Configure, Update, Replay, Discover, Move, and Delete. Specific permissions are granted for 'Authenticated Users' (checked) and 'Anonymous' (unchecked). A search bar and an 'Add' button are also visible.

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.

[Home](#) > [Default Directory](#) | [Custom domain names](#) >

singhritesh85.com

Custom domain name

Delete | Got feedback?

i To use singhritesh85.com with your Microsoft Entra tenant, create a new TXT record with your domain name registrar using the info below.

Record type

TXT

MX

Alias or host name

@

Copied

Destination or points to address

[REDACTED]



TTL

3600



[Share these settings via email](#)

Verification will not succeed until you have configured your domain with your registrar as described above.

Verify

Do the entry for Alias or hostname with Destination in Azure DNS Zone to create a new record set as shown in the screenshot attached below.

Add record set

singhrithesh85.com

Name
@
.singhrithesh85.com

Type
TXT – Text records

TTL *
1

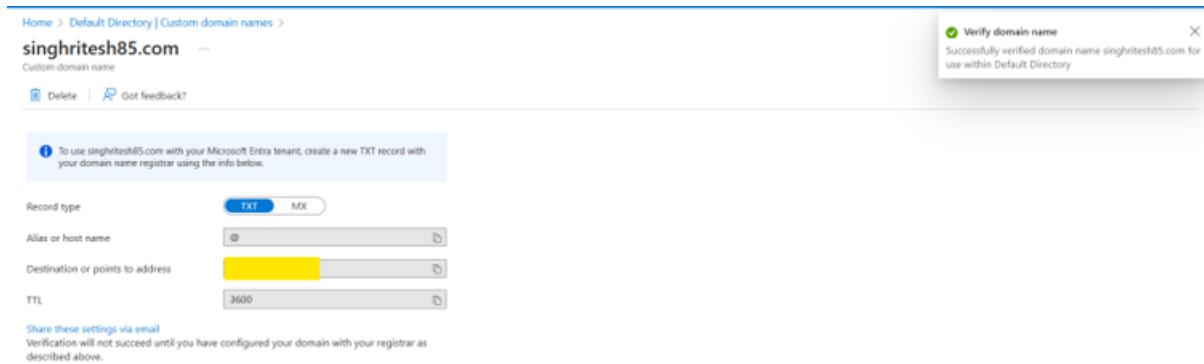
TTL unit
Hours

Value
 XXXXXXXXXX Delete

The quick brown fox jumps over the lazy dog.

Add Cancel Give feedback

Then verified the 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.

Home > Default Directory | Custom domain names >

singhritesh85.com

Custom domain name

Make primary Delete

Type	Custom
Status	Verified
Federated	No
Primary domain	No
In use	No

Name	Status	Federated	Primary
singhritesh85.com	✓ Verified		✓

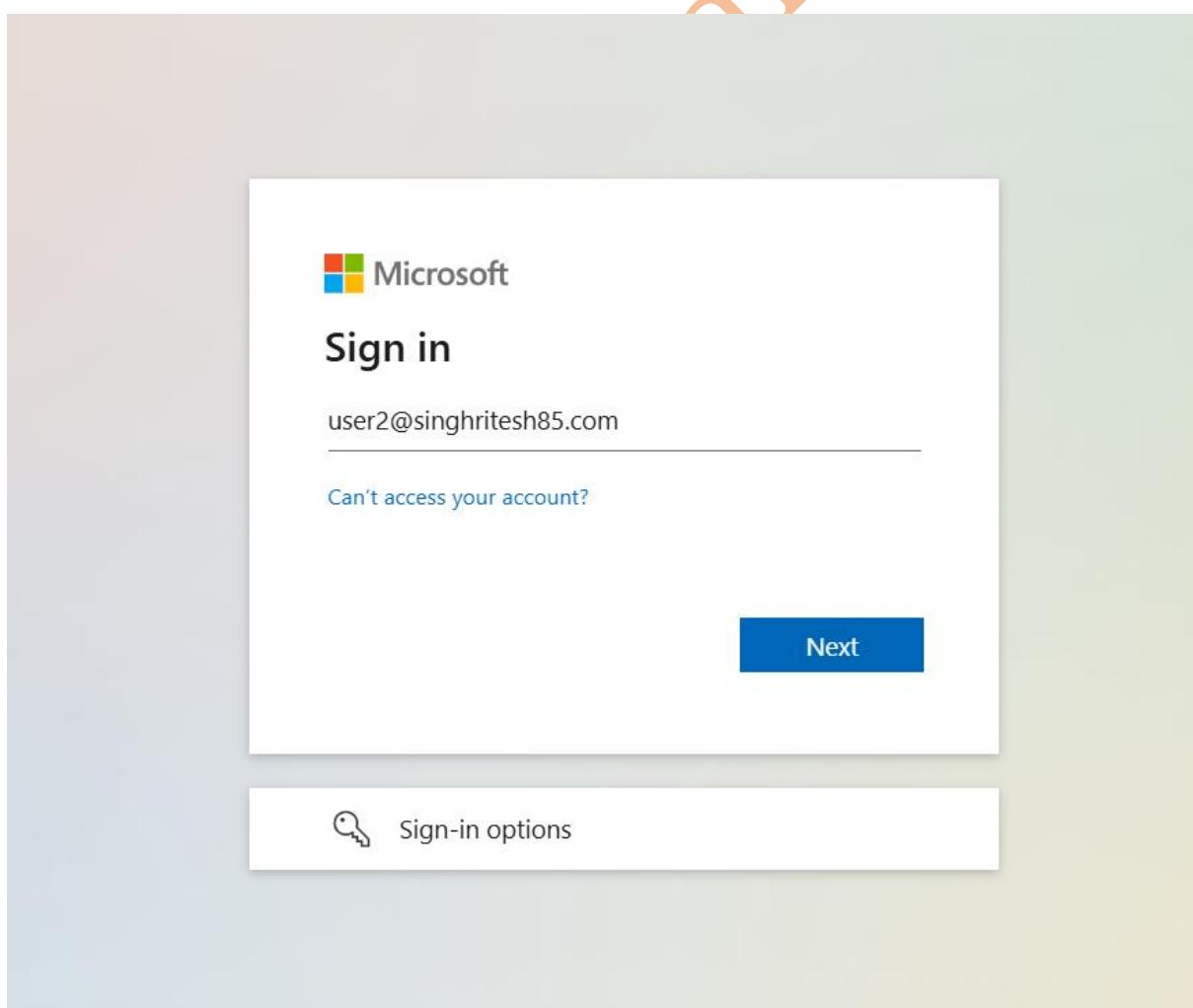
Then created two member users user1 and user2 as shown in the screenshot attached below.

<input type="checkbox"/>	 User1	user1@singhritesh85.com		Member
<input type="checkbox"/>	 User2	user2@singhritesh85.com		Member

For the very first time I logged-in with a user then I provided user1 as administrator privileges and user2 as restricted access and remove the Authenticated user overall administrator access which I provided before first time logged-in as shown in the screenshot attached below.

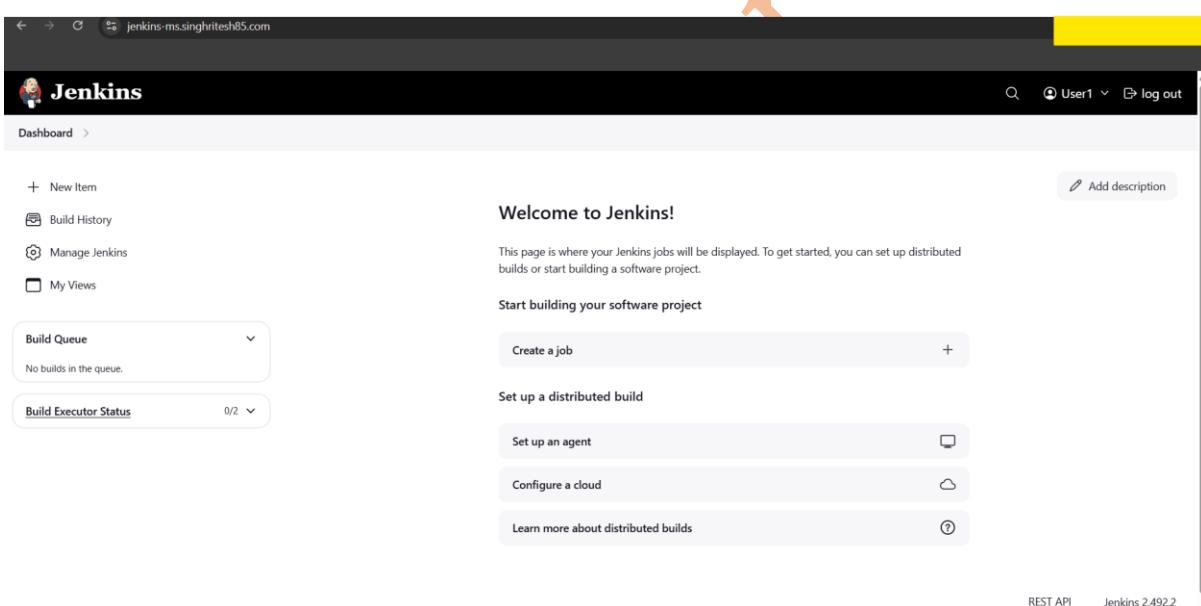
User/group	Overall	Credentials	Agent	Job	Run	View	SCM	Metrics
Anonymous								
Authenticated Users								
User2	<input checked="" type="checkbox"/>							
User1	<input checked="" type="checkbox"/>							
Azure User/group to add								
<input type="text"/> Search for a name <input type="button"/> Add								

Then I logged-in with user2 and checked the access which user2 got as shown in the screenshot attached below.





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



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.

SonarQube Marketplace screenshot showing the 'Azure Active Directory (AAD) Authentication Plug-in for SonarQube' plugin highlighted with a yellow box.

Plugin Name	Description	Version	Details	Action
1C (BSL) Community Plugin <small>LANGUAGES</small>	Code Analyzer for 1C (BSL)	1.16.2	Support SQ 25.1+ ... Homepage Issue Tracker Licensed under GNU LGPL v3 Developed by 1c-syntax	Install
AEM Rules for SonarQube <small>EXTERNAL ANALYSERS</small>	Adds rules for AEM Java development	1.6	SonarQube 8.9 LTS compatibility release due to underlying Java plugin API changes ... Homepage Issue Tracker Licensed under The Apache Software License Developed by Wunderman Thompson Technology	Install
Ansible Lint <small>EXTERNAL ANALYZERS</small>	Analyze Ansible playbooks	2.5.1	Support for SonarQube 9.2 ... Homepage Issue Tracker Licensed under Apache License, Version 2.0	Install
Apigee <small>EXTERNAL ANALYZERS</small>	Adds XML, rules test Apigee proxies.	3.0.2	Support for SQ 9.7+ ... Homepage Issue Tracker Licensed under Apache License, Version 2.0	Install
Azure Active Directory (AAD) Authentication Plug-in for SonarQube <small>INTEGRATION</small>	Allows the use of Azure Active Directory as an authentication source for SonarQube	1.3.2	Updates commons-text to fix CVE-2022-42889 ... Homepage Issue Tracker Licensed under The MIT License (MIT) Developed by ALM DevOps Rangers	Install
CVS <small>INTEGRATION</small>	Provides SCM CVS integration	1.1.1	Fix classnotfound error ... Homepage Issue Tracker Licensed under GNU LGPL 3	Install

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

SonarQube Marketplace screenshot after restarting the server. The 'Azure Active Directory (AAD) Authentication Plug-in for SonarQube' plugin's 'Install' button is now labeled 'Install Pending'.

Plugin Name	Description	Version	Details	Action
1C (BSL) Community Plugin <small>LANGUAGES</small>	Code Analyzer for 1C (BSL)	1.16.2	Support SQ 25.1+ ... Homepage Issue Tracker Licensed under GNU LGPL v3 Developed by 1c-syntax	Install
AEM Rules for SonarQube <small>EXTERNAL ANALYZERS</small>	Adds rules for AEM Java development	1.6	SonarQube 8.9 LTS compatibility release due to underlying Java plugin API changes ... Homepage Issue Tracker Licensed under The Apache Software License Developed by Wunderman Thompson Technology	Install
Ansible Lint <small>EXTERNAL ANALYZERS</small>	Analyze Ansible playbooks	2.5.1	Support for SonarQube 9.2 ... Homepage Issue Tracker Licensed under Apache License, Version 2.0	Install
Apigee <small>EXTERNAL ANALYZERS</small>	Adds XML, rules test Apigee proxies.	3.0.2	Support for SQ 9.7+ ... Homepage Issue Tracker Licensed under Apache License, Version 2.0	Install
Azure Active Directory (AAD) Authentication Plug-in for SonarQube <small>INTEGRATION</small>	Allows the use of Azure Active Directory as an authentication source for SonarQube	1.3.2	Updates commons-text to fix CVE-2022-42889 ... Homepage Issue Tracker Licensed under The MIT License (MIT) Developed by ALM DevOps Rangers	Install Pending
CVS <small>INTEGRATION</small>	Provides SCM CVS integration	1.1.1	Fix classnotfound error ... Homepage Issue Tracker Licensed under GNU LGPL 3	Install

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

SonarQube Administration Settings - General Settings

Server base URL: https://sonarqube.singhritesh85.com/

Issues

Default Assignee: New issues will be assigned to this user each time it is not possible to determine the user who is the author of the issue.

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

Home > Default Directory | App registrations >

Register an application ...

Supported account types:

- Accounts in this organizational directory only (Default Directory only - Single tenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

Help me choose...

Redirect URI (optional): https://sonarqube.singhritesh85.com/oauth2/callback/aad

By proceeding, you agree to the Microsoft Platform Policies [Learn more](#)

Register

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

Search | Overview | Delete | Endpoints | Preview features

Essentials

Display name	: SonarQube	Client credentials	: 0_certificate_1_secret
Application (client) ID	: [REDACTED]	Redirect URIs	: 1_web_0_spas_0_public_client
Object ID	: [REDACTED]	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: [REDACTED]	Managed application in L...	: SonarQube

Supported account types: My organization only

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

Get Started Documentation

Build your application with the Microsoft identity platform

The Microsoft identity platform is an authentication service, open-source libraries, and application management tools. You can create modern, standards-based authentication solutions, access and protect APIs, and add sign-in for your users and customers. [Learn more](#)

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.

The screenshots show the Sonarqube Administration interface with the 'General Settings' tab selected. The left sidebar lists various configuration categories. In the main area, under 'General Settings', there are sections for 'Enabled', 'Client ID', 'Client Secret', 'Tenant ID', and 'Allow users to sign-up'. The 'Enabled' section has a yellow border around it, indicating it's the current focus. The 'Client ID' and 'Client Secret' sections also have yellow borders. The 'Tenant ID' and 'Allow users to sign-up' sections do not have yellow borders.

Enabled:

- Enable Azure AD users to login.
- Value is ignored if client ID and secret are not defined.

Client ID:

Client ID provided by Azure AD when registering the application. Key: sonar.auth.aad.clientId.secured

Client Secret:

Client key provided by Azure AD when registering the application. Key: sonar.auth.aad.clientSecret.secured

Tenant ID:

Azure AD Tenant ID. Key: sonar.auth.aad.tenantId

Allow users to sign-up:

Allow new users to authenticate. When set to 'False', only existing users will be able to authenticate to the system.

sonarqube.singhritesh85.com/admin/settings?category=aad

Administration

General Settings

Edit global settings for this SonarQube instance.

Find in Settings

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.

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)

Reset Default: Unique

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

sonarqube.singhritesh85.com/sessions/new?return_to=%2F

Log in to SonarQube

Log in with Microsoft

More options

SonarQube™ technology is powered by SonarSource SA
[LGPL v3](#) - [Community](#) - [Documentation](#) - [Plugins](#)

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

The screenshot shows the SonarQube Administration interface under the Global Permissions section. It lists users and groups with their assigned permissions across four categories: Administer System, Administer Analysis, Execute Analysis, and Create.

	Administer System	Administer Analysis	Execute Analysis	Create
sonar-administrators System administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input checked="" type="checkbox"/> Projects
sonar-users Every authenticated user automatically belongs to this group	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Projects
Ritesh ritesh	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input checked="" type="checkbox"/> Projects
User1 user1 user1@singhritesh85.com	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input checked="" type="checkbox"/> Projects
User2 user2 user2@singhritesh85.com	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input type="checkbox"/> Projects
Anyone DEPRECATED Anybody who browses the application belongs to this group. If authentication is not enforced, assigned permissions also apply to non-authenticated users.	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input type="checkbox"/> Projects
Administrator admin	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input type="checkbox"/> Projects

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.

The screenshot shows the Azure App Registration - Register an application page. A new application named "Grafana-login" has been registered. Key configuration details include:

- Supported account types:** Accounts in this organizational directory only (Default Directory only - Single tenant).
- Redirect URI (optional):** https://grafana.singhritesh85.com/login/azuread
- By proceeding, you agree to the Microsoft Platform Policies**
- Register** button

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

Home > Grafana-login

Grafana-login | Certificates & secrets

Search Got feedback?

Overview Quickstart Integration assistant Diagnose and solve problems Manage Branding & properties Authentication Certificates & secrets Token configuration API permissions Expose an API App roles Owners Roles and administrators Manifest Support + Troubleshooting

Got a second to give us some feedback? →

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) Client secrets (1) Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

New client secret

Description	Expires	Value	Secret ID
demo	2025		

Client ID

Now, login into the Grafana for the first time 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.

grafana.singhritesh85.com/admin/authentication

Home > Administration > Authentication

Authentication

Manage your auth settings and configure single sign-on. Find out more in our documentation.

GitHub OAuth	Not enabled
GitLab OAuth	Not enabled
Google OAuth	Not enabled
Generic OAuth	Not enabled
Azure AD OAuth	Not enabled
Okta OAuth	Not enabled

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**).

The image consists of three vertically stacked screenshots of the Grafana Admin UI, specifically the 'Authentication > Azure AD' configuration page.

Screenshot 1: General settings

- Display name:** Will be displayed on the login page as "Sign in with ...". Helpful if you use more than one identity providers or SSO protocols.
- Azure Entra ID:** Client authentication method used to authenticate to the token endpoint. Set to "Client secret".
- Client Id:** The client Id of your OAuth2 app. (redacted)
- Client secret:** The client secret of your OAuth2 app. (redacted)
- FIC managed identity client id:** The managed identity client id of the federated identity credential of your OAuth2 app. (redacted)
- FIC audience:** The audience of the federated identity credential of your OAuth2 app. (redacted)

Screenshot 2: Extra security measures

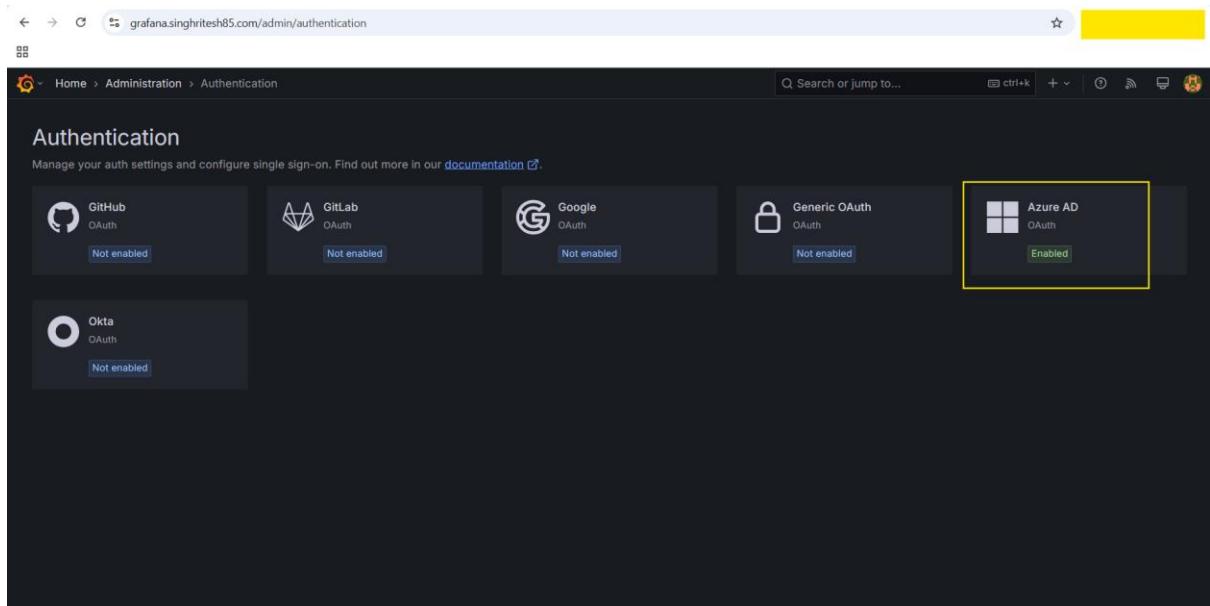
- Scopes:** openid, email, profile
- Auth URL:** https://login.microsoftonline.com/[REDACTED]/oauth2/v2.0
- Token URL:** https://login.microsoftonline.com/[REDACTED]/oauth2/v2.0
- Allow sign up:** Enabled (radio button)
- Auto login:** Enabled (radio button)
- Sign out redirect URL:** The URL to redirect the user to after signing out from Grafana. (redacted)

Screenshot 3: User mapping

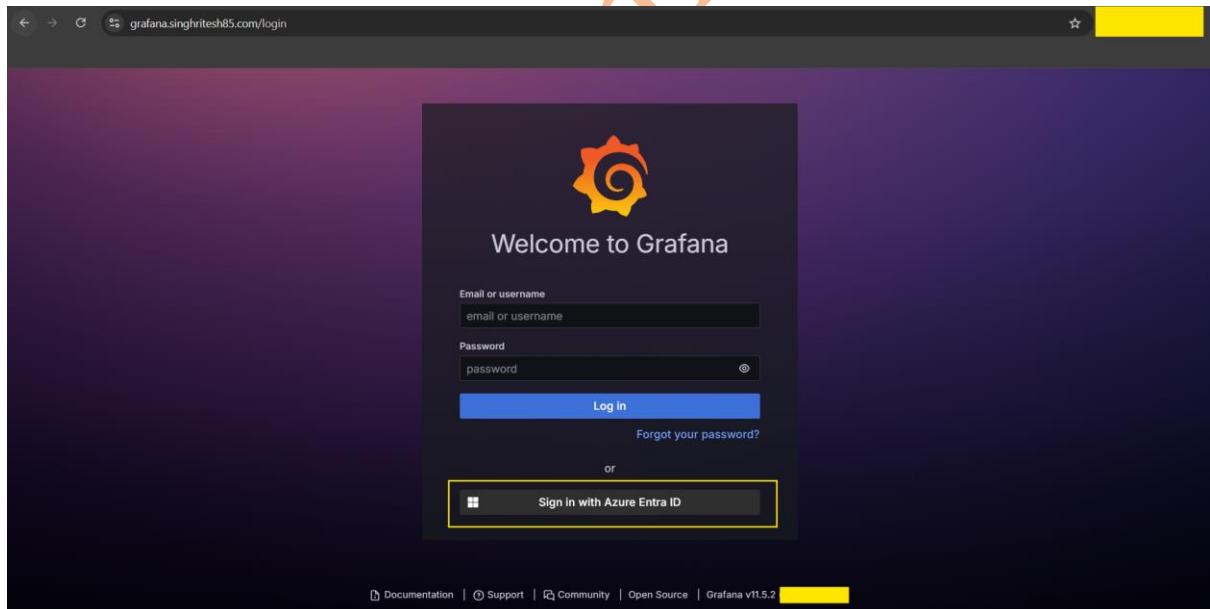
- Role attribute strict mode:** If enabled, denies user login if the Grafana role cannot be extracted using Role attribute path. (radio button)
- Organization mapping:** List of <GroupId><OrgIdOrName><Role> mappings. Enter mappings (my-team:1:viewer...) and press Enter to add.
- Allow assign Grafana admin:** If enabled, it will automatically sync the Grafana server administrator role. (radio button)
- Skip organization role sync:** Prevent synchronizing users' organization roles from your IdP. (radio button)

Buttons at the bottom: Disable, Save, Discard, : (dropdown menu)

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



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.



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@yellow ~]# 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
;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

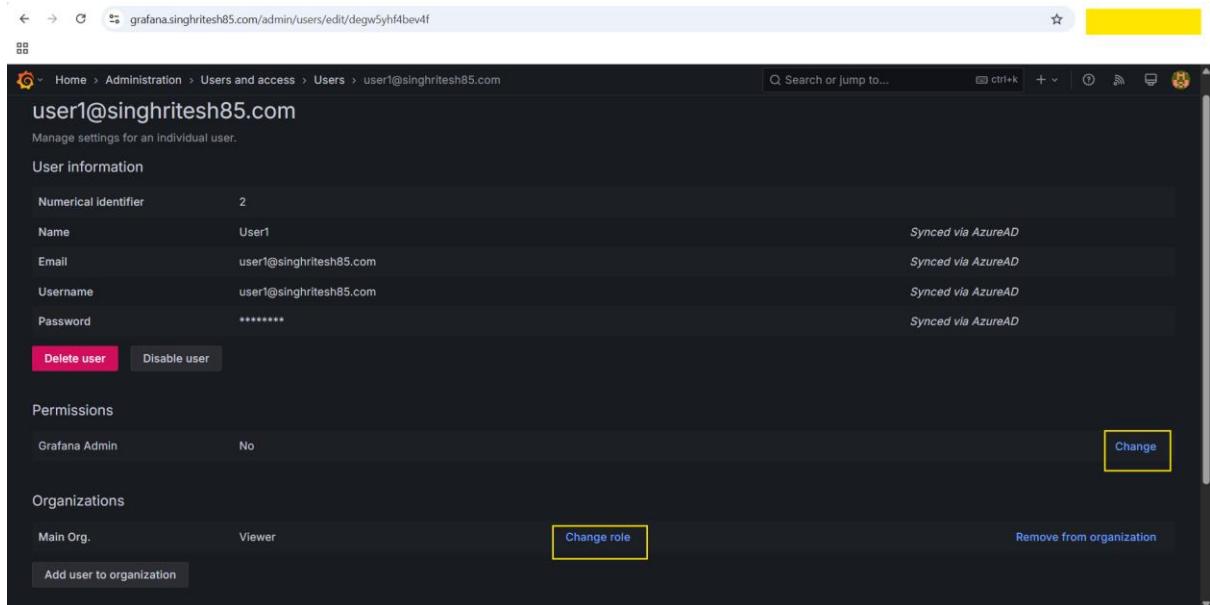
# The full public facing url you use in browser, used for redirects and emails
# If you use reverse proxy and sub path specify full url (with sub path)
;root_url = %(protocol)s://%(domain)s:%(http_port)s/

root_url = "https://grafana.singhrithesh85.com"
# Serve Grafana from subpath specified in `root_url` setting. By default it is set to `false` for compatibility reasons
;root_path = "/"

[root@[REDACTED] ~]# systemctl restart grafana-server.service
[root@[REDACTED] ~]# 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 Tue 2025-[REDACTED]
```

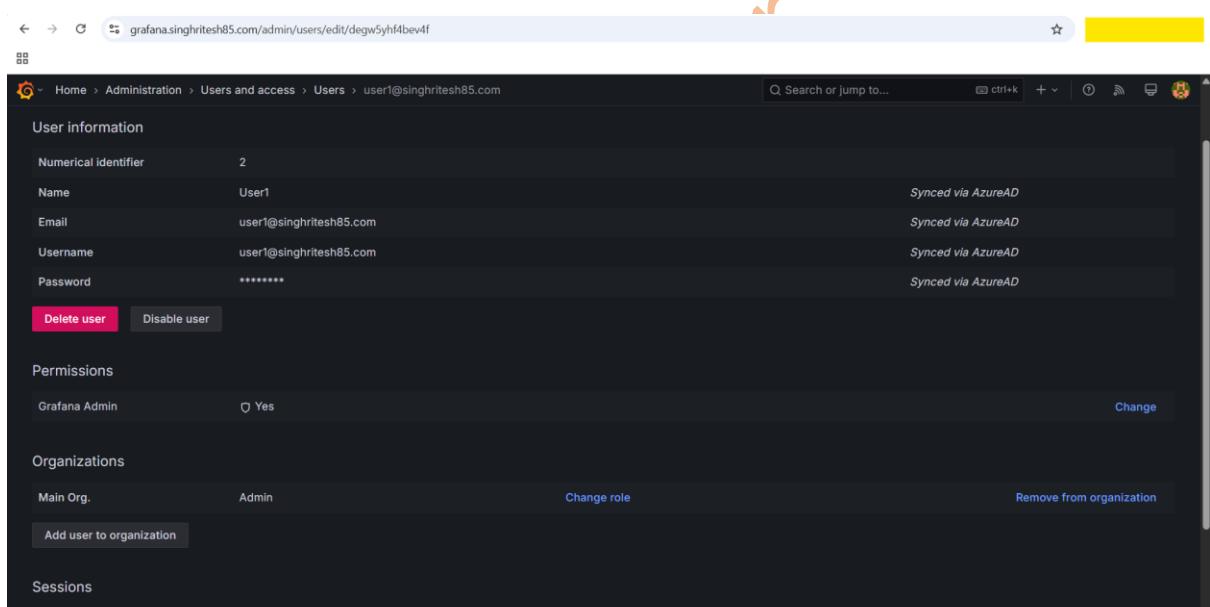
For the first time when a user logged-in, they had viewer access which Grafana Administrator can change as per the requirement as shown in the screenshot attached below.

Login	Email	Name	Last active	Origin
admin	admin@localhost		3 minutes	
user1@singhrithesh85.com	user1@singhrithesh85.com	User1	2 minutes	AzureAD
user2@singhrithesh85.com	user2@singhrithesh85.com	User2	< 1 minute	AzureAD

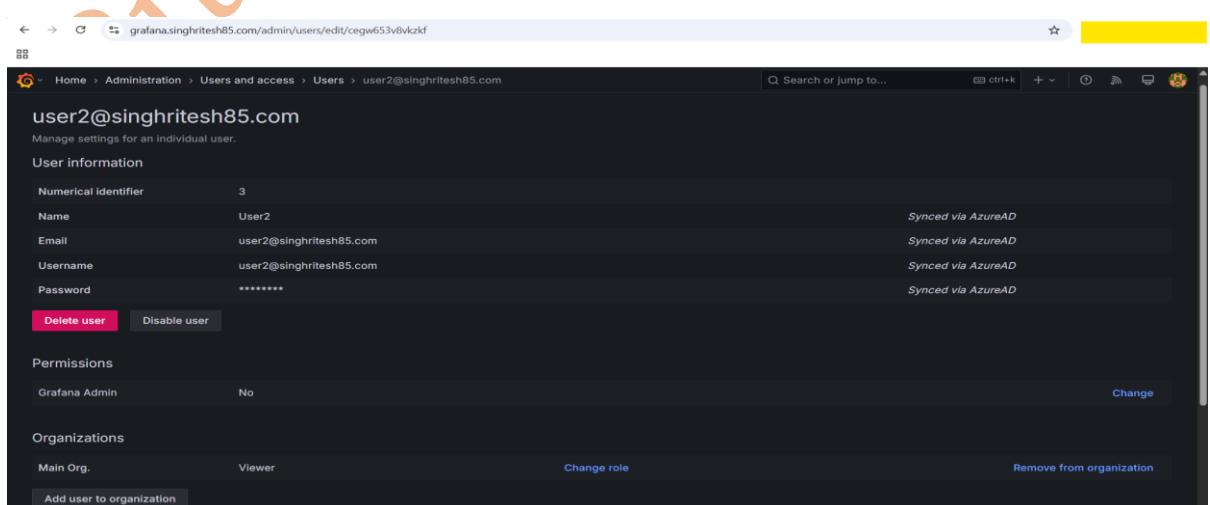


The screenshot shows the Grafana user settings page for user1@singhritesh85.com. In the 'Permissions' section, under 'Grafana Admin', the 'Yes' radio button is selected. A yellow box highlights the 'Change' button next to the 'Grafana Admin' field.

As stated earlier I will provide Admin access to user1 and viewer access to user2 as shown in the screenshot attached below.



The screenshot shows the Grafana user settings page for user1@singhritesh85.com. In the 'Permissions' section, under 'Grafana Admin', the 'Yes' radio button is selected. A yellow box highlights the 'Change' button next to the 'Grafana Admin' field.



The screenshot shows the Grafana user settings page for user2@singhritesh85.com. In the 'Permissions' section, under 'Grafana Admin', the 'No' radio button is selected. A yellow box highlights the 'Change' button next to the 'Grafana Admin' field.

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

Installation of Nginx Ingress Controller in EKS Cluster

```
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-2:02XXXXXXXXXX6:certificate/XXXXXXX-XXXX-XXXX-XXXX-XXXX-XXXXXX --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"
```

```
[jenkins@███████████ ~]$ kubectl create ns ingress-nginx
```

```
[jenkins@███████████ ~]$ helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
```

```
[jenkins@███████████ ~]$ helm repo update
```

```
[jenkins@ip-10-10-4-202 ~]$ 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-2:02XXXXXXXXXX6:certificate/XXXXXXX-XXXX-XXXX-XXXX-XXXXXX --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-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"
```

NAME	READY	STATUS	RESTARTS	AGE
ingress-nginx-controller-6	b	1/1	Running	0 2m31s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
ingress-nginx-controller	LoadBalancer	172. ███████████ .165.a ███████████	1.us-east-2.elb.amazonaws.com	80:3 ███████████ /TCP,443:TCP
ingress-nginx-controller-admission	ClusterIP	172. ███████████ .190	<none>	443/TCP

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
```

NAME	STATUS	ROLES	AGE	VERSION
ip-10- ███████████ -121.us-east-2.compute.internal	Ready	<none>	87m	v1.30.4-eks-a ███████████ 9
ip-10- ███████████ -153.us-east-2.compute.internal	Ready	<none>	87m	v1.30.4-eks-a ███████████ 9


```
[jenkins@███████████ ~]$ kubectl create namespace argocd
namespace/argocd created
[jenkins@███████████ ~]$ 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.

```
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
```

```
[jenkins@yellow ~]$ 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
[jenkins@yellow ~]$ kubectl apply -f argocd-ingress-rule.yaml
kubecingress.networking.k8s.io/minimal-ingress created
[jenkins@yellow ~]$ kubectl get ing -n argocd
NAME      CLASS   HOSTS           ADDRESS      PORTS   AGE
minimal-ingress  nginx  argocd.singhritesh85.com  80       11s
[jenkins@yellow ~]$ kubectl get ing -n argocd --watch
NAME      CLASS   HOSTS           ADDRESS      PORTS   AGE
minimal-ingress  nginx  argocd.singhritesh85.com  80       16s
minimal-ingress  nginx  argocd.singhritesh85.com  a[REDACTED] 1.us-east-2.elb.amazonaws.com  80       33s
```

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.

singhritesh85.com | Records

Add record set

Name: argocd

Type: CNAME – Link your subdomain to another record

Alias record set: No

TTL: 1

TTL unit: Hours

Alias: 0 [REDACTED] 1.us-east-2.elb.amazonaws.com

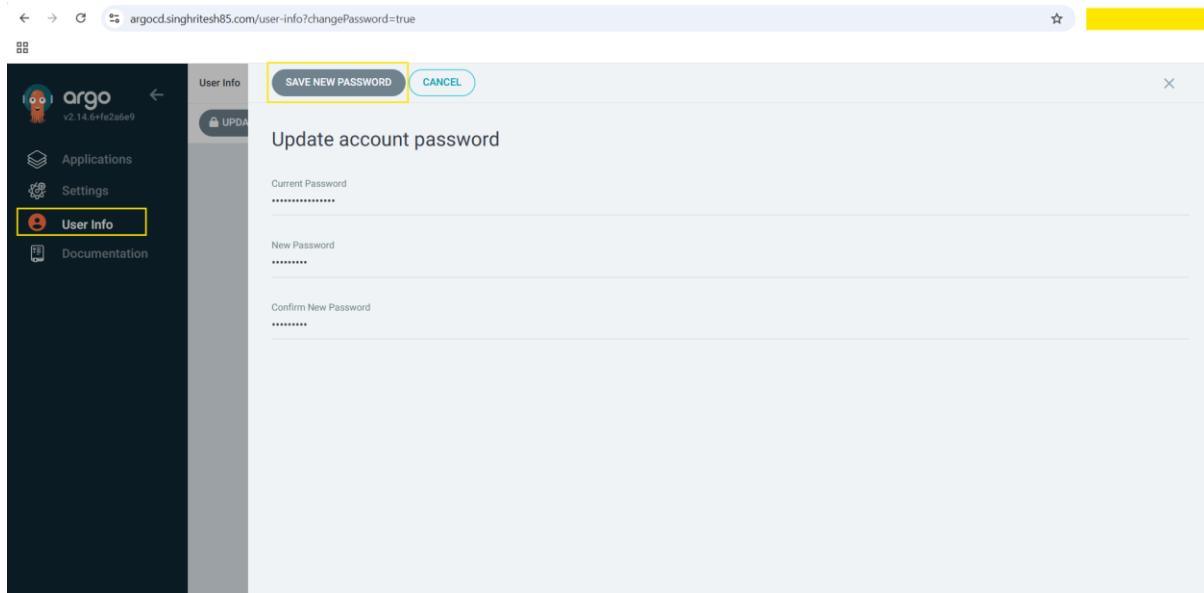
Add

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

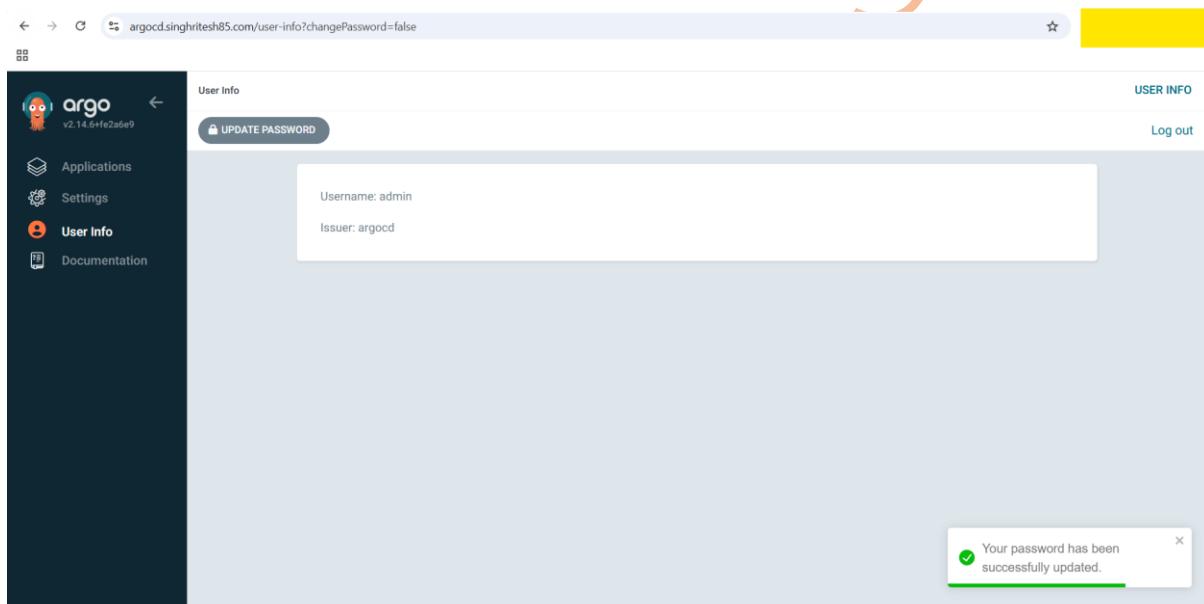
```
kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath='{.data.password}' | base64 -d
```

```
[jenkins@yellow ~]$ kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath='{.data.password}' | base64 -d
U[REDACTED]d[jenkins@ip-10-10-4-202 ~]$
```

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



Now the Password was successfully updated as verified by the screenshot attached below.



Installation of ArgoCD CLI is 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
```

```
[jenkins@yellow ~]$ curl -sSL -o argocd-linux-amd64 https://github.com/argoproj/argo-cd/releases/latest/download/argocd-linux-amd64
[jenkins@yellow ~]$ sudo install -m 555 argocd-linux-amd64 /usr/local/bin/argocd
[jenkins@yellow ~]$ rm argocd-linux-amd64
```

TMDB API Key can be used as shown below while building the Docker Image.

The screenshot shows the TMDB API settings page. On the left, there's a sidebar with 'Settings' selected. Under 'API', the 'API Key' section is highlighted with a yellow box. To the right, there's a main content area with sections for Overview, Details, Sessions, Stats, and Regenerate Key. Below these are Documentation, Support, and API Details sections. The 'API Key' section contains a text input field where the API key is displayed as a redacted string. A 'Logout' button is at the bottom right of the sidebar.

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.

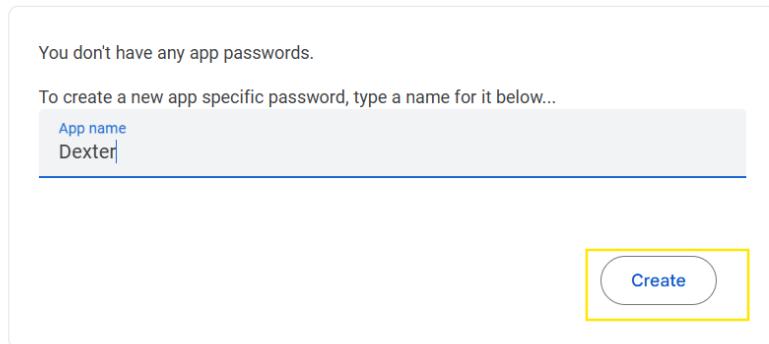
The screenshot shows the Google Account security page. The 'Security' tab is selected in the sidebar. A search bar at the top has 'app password' typed into it. Below the search bar, the results list includes 'App passwords' which is highlighted with a yellow box. Other options like 'Password Manager', 'Web & App Activity', and 'Sign in with app passwords' are also listed. The main content area shows 'Recent security activity' with a note 'No security activity or alerts in the last 28 days'. Below that is a 'How you sign in to Google' section with a note '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](#)



singh

← App passwords

App passwords help you sign into your Google Account on older apps and services that don't support 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](#)

Your app password
Dexter

To create a new app password, type a name for it below...

App name

Generated app password

Your app password for your device

[REDACTED]

How to use it

Go to the settings for your Google Account in the application or device you are trying to set up. Replace your password with the 16-character password shown above.

Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.

Done

I had deleted this App Password after completion of this project, this App password does not exist anymore.

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

The screenshots show the Jenkins 'Manage Jenkins > System' configuration page. The top screenshot displays the 'E-mail Notification' section, where the SMTP server is set to 'smtp.gmail.com' and the default user e-mail suffix is '@gmail.com'. Authentication is enabled with a User Name and Password. A warning message states: 'For security when using authentication it is recommended to enable either TLS or SSL'. The bottom screenshot shows the 'Advanced' section, where 'Use SSL' and 'Use TLS' are checked, and the SMTP Port is set to 465. A message indicates that an email was successfully sent.

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.

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

```
#####
# SMTP / Emailing #####
[smtp]
enabled = true
host = smtp.gmail.com:587
user = [REDACTED]@gmail.com
# If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"""
password = [REDACTED]
;cert_file =
;key_file =
skip_verify = true
from_address = [REDACTED]@gmail.com
from_name = Netflix Clone
# 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
```

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

```
[root@[REDACTED] ~]# systemctl restart grafana-server.service
[root@[REDACTED] ~]# 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 Tue 2025-
```

Creation of Alerts in Grafana I will discuss later here I will discuss first integration of Jenkins with SonarQube then will create the Jenkins Job and deploy the Netflix 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-Netflix-Clone-Aws.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.

The screenshot shows the SonarQube interface at the URL sonarqube.singhritesh85.com/account/security. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, Profile, Security, Notifications, and Projects. The main content area is titled 'Tokens'. It displays a message about using tokens for security instead of user logins. A 'Generate Tokens' section allows creating a new token with a name ('SonarQube'), type ('Global'), and expiration ('Never'). A success message indicates a new token has been created, with a 'Copy' button next to its value. A table lists the generated token: Name (SonarQube), Type (Global), Project (empty), Last use (Never), Created (2025-01-12), and Expiration (Never). A 'Revoke' button is present for the token.

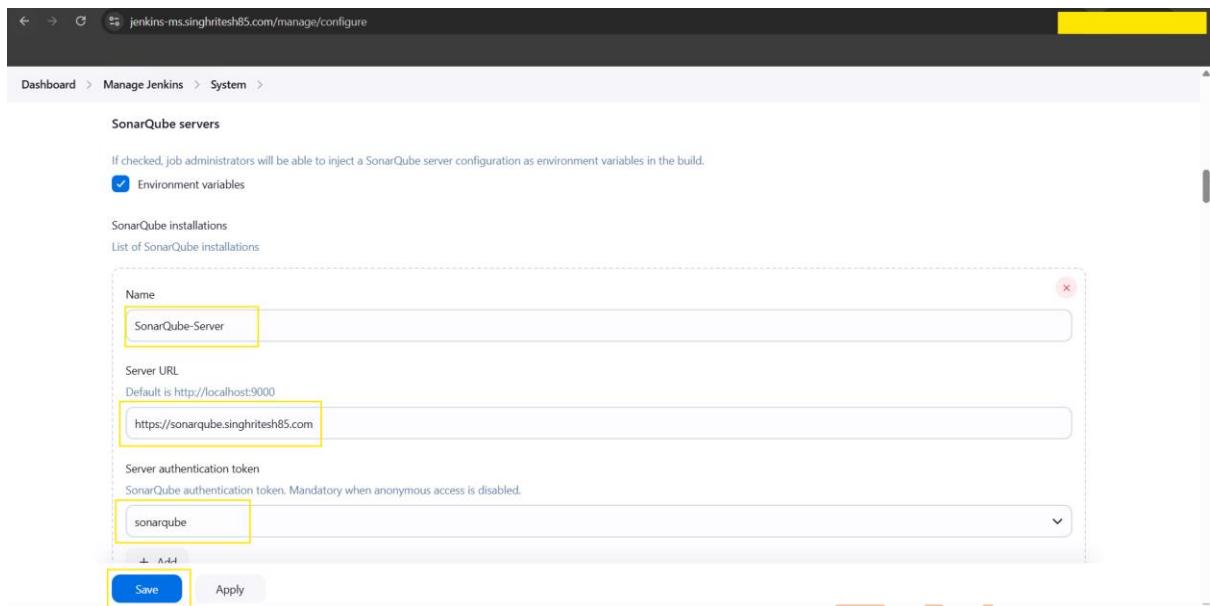
Now go to Jenkins Manage Jenkins > System > Plugins > Available Plugins and install the **SonarQube-Scanner** Plugin as shown in the screenshot attached below.

The screenshot shows the Jenkins Plugin Manager interface. In the search bar at the top right, 'sonarqube scanner' is typed. Below the search bar, there is a button labeled 'Install' with a yellow border. To the right of the search bar, there is a user icon for 'User1' and a 'log out' link. The main area is titled 'Plugins' and contains a sidebar with links for 'Updates', 'Available plugins' (which is selected), 'Installed plugins', and 'Advanced settings'. The 'Available plugins' section lists the 'SonarQube Scanner' plugin by SonarSource, version 2.18, released 1 month 26 days ago. The plugin description states: 'This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.' At the bottom right of the page, it says 'REST API' and 'Jenkins 2.492.2'.

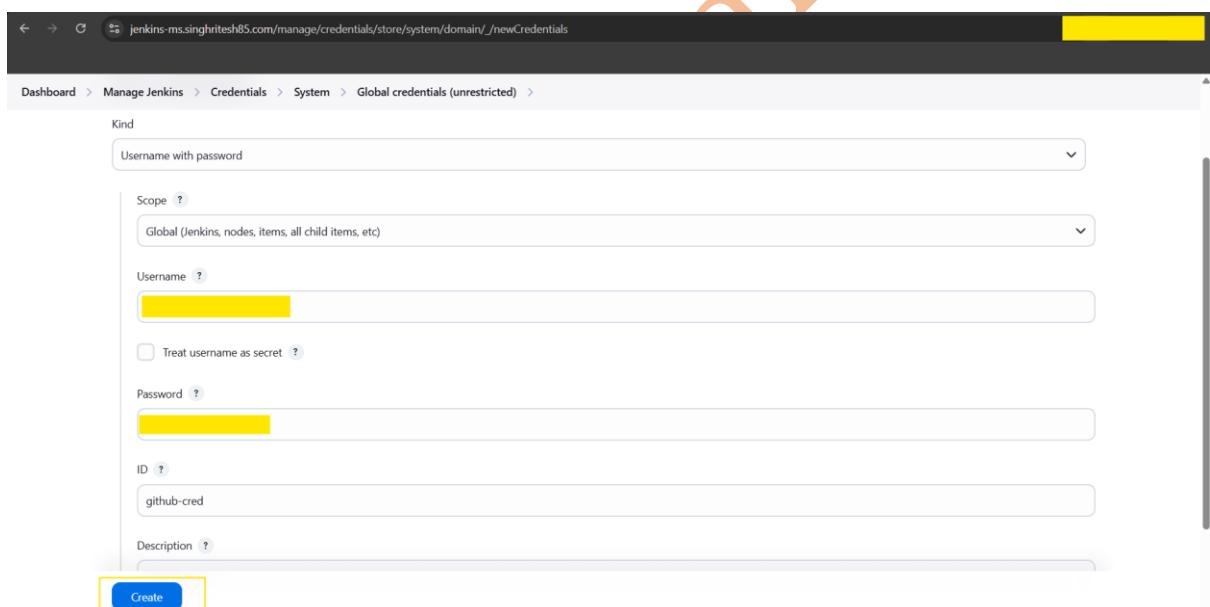
Do not restart the Jenkins after installation of **SonarQube-Scanner Plugin**. Then create a Jenkins Credential of kind Secret text with the SonarQube Secret Token which I created earlier as shown in the screenshot attached below.

The screenshot shows the 'New credentials' configuration screen. The 'Kind' dropdown is set to 'Secret text'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc.)'. The 'Secret' field contains a yellowed-out value. The 'ID' field is set to 'sonarqube'. The 'Description' field is also set to 'sonarqube'. At the bottom left, there is a blue 'Create' button with a yellow border. The URL in the browser bar is 'Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >'.

No go to Jenkins Manage Jenkins > System and search for SonarQube and do the configuration as shown in the screenshot attached below.



I had created the credential named as github-cred in 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.

The screenshot shows the Jenkins 'New credentials' page. The 'Kind' dropdown is set to 'Username with password'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains a yellow placeholder. A checkbox 'Treat username as secret' is present. The 'Password' field contains a yellow placeholder. The 'ID' field is set to 'jenkins-cred'. A blue 'Create' button is at the bottom.

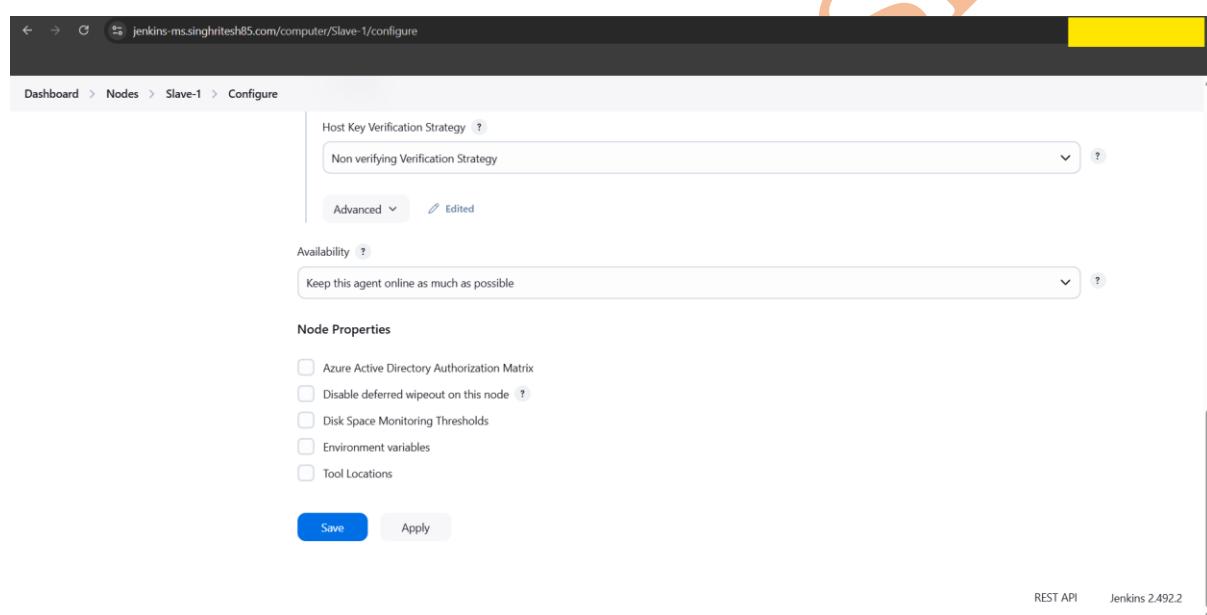
I had integrated a Slave node in Jenkins with the name of **Slave-1**. To do so go to Manage Jenkins > Nodes > New Node as shown in the screenshot attached below.

The screenshot shows the Jenkins 'New node' page. The 'Node name' field is set to 'Slave-1'. The 'Type' section shows 'Permanent Agent' selected. A description explains it adds a plain, permanent agent to Jenkins. A blue 'Create' button is at the bottom.

The screenshot shows the Jenkins 'Configure' page for the 'Slave-1' node. The 'Name' field is set to 'Slave-1'. The 'Description' field contains 'This is a Slave Node.' The 'Build Executor Status' section shows 'Slave-1' with '(offline)' status. The 'Number of executors' field is set to '2'. The 'Remote root directory' field is set to '/home/jenkins'. A 'Save' and 'Apply' button are at the bottom.



The screenshot shows the Jenkins configuration page for 'Slave-1'. In the 'Host' section, there is a yellow box highlighting the 'Private IP Address of Jenkins Slave Node' field, which contains a placeholder IP address. In the 'Credentials' section, there is another yellow box highlighting the dropdown menu where the service account was selected.



The screenshot shows the Jenkins configuration page for 'Slave-1'. In the 'Availability' section, the dropdown menu is set to 'Keep this agent online as much as possible'. In the 'Node Properties' section, several checkboxes are listed: 'Azure Active Directory Authorization Matrix', 'Disable deferred wipeout on this node', 'Disk Space Monitoring Thresholds', 'Environment variables', and 'Tool Locations'. The 'Tool Locations' checkbox is highlighted with a yellow box.

Now the Slave-1 is in Online mode and 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 namespace **netflix** but does not have access for the entire EKS cluster. That means deployment user jenkins access was restricted to the namespace **netflix** in the EKS Cluster.

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

Rite

```
cat sa-role-rolebinding.yaml

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

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: netflix
  annotations:
    kubernetes.io/service-account.name: jenkins
[root@... ~]# kubectl apply -f secrets.yaml
secret/mysecretname created
[root@... ~]# kubectl get secrets -n netflix
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: netflix
  annotations:
    kubernetes.io/service-account.name: jenkins
```

```
[root@... ~]# kubectl describe secret mysecretname -n netflix
Name:         mysecretname
Namespace:    netflix
Labels:      <none>
Annotations: kubernetes.io/service-account.name: jenkins
              kubernetes.io/service-account.uid: 6
Type:        kubernetes.io/service-account-token

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

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@██████████ ~]$ cat ~/.kube/config
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data:
  server: https://██████████.us-east-2.eks.amazonaws.com
  name: arn:aws:eks:us-east-2:██████████:cluster/eks-demo-cluster-dev
contexts:
- context:
  cluster: arn:aws:eks:us-east-2:██████████:cluster/eks-demo-cluster-dev
  user: jenkins
  name: dexter
  current-context: dexter
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 netflix namespace 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:netflix:jenkins" cannot list resource "nodes" in API group "" at the cluster scope
[jenkins@██████████ ~]$ kubectl get pods -n netflix
No resources found in netflix namespace.
[jenkins@██████████ ~]$
```

I had executed the Jenkins Job as shown in the screenshot attached below.

```

pipeline{
    agent{
        node{
            label "Slave-1"
            customWorkspace "/home/jenkins/mydemo"
        }
    }
    environment{
        JAVA_HOME="/usr/lib/jvm/java-17-amazon-corretto.x86_64"
        PATH="$PATH:$JAVA_HOME/bin:/opt/sonar-scanner/bin:/opt/dependency-
check/bin:/opt/node-v16.0.0/bin"
    }
    parameters {
        string(name: 'COMMIT_ID', defaultValue: "", description: 'Provide the Commit ID')
        string(name: 'REPO_NAME', defaultValue: "", description: 'Provide ECR Repository URI')
        string(name: 'TAG_NAME', defaultValue: "", description: 'Provide a tag name for Docker Image')
        string(name: 'REPLICA_COUNT', defaultValue: "", description: 'Provide the number of Pods to be
created')
    }
    stages{
        stage("clone-code"){
            steps{
                cleanWs()
                checkout scmGit(branches: [[name: "${COMMIT_ID}"]], extensions: [], userRemoteConfigs:
[[credentialsId: 'github-cred', url: 'https://github.com/singhritesh85/DevSecOps-Project.git']])
            }
        }
        stage("SonarAnalysis"){
            steps {
                withSonarQubeEnv('SonarQube-Server') {
                    sh 'sonar-scanner -Dsonar.projectKey=netflix-clone -Dsonar.projectName=netflix-clone'
                }
            }
        }
    }
}

```

```

    }

}

stage("Quality Gate") {
    steps {
        timeout(time: 1, unit: 'HOURS') {
            waitForQualityGate abortPipeline: true
        }
    }
}

stage("Install Dependencies"){
    steps {
        sh 'npm install'
    }
}

stage("OWASP Dependency Check"){
    steps{
        sh 'dependency-check.sh --disableYarnAudit --disableNodeAudit --scan . --out .'
    }
}

stage("Trivy Scan files"){
    steps{
        sh 'trivy fs . > /home/jenkins/trivy-filescan.txt'
    }
}

stage("Docker-Image"){
    steps{
        sh 'docker build --build-arg TMDB_V3_API_KEY=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -t myimage:1.06 . --no-cache'
        sh 'docker tag myimage:1.06 ${REPO_NAME}:${TAG_NAME}'
        sh 'trivy image --exit-code 0 --severity MEDIUM,HIGH ${REPO_NAME}:${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 'yes|argocd login argocd.singhritesh85.com --username admin --password Admin@123'

sh 'argocd login argocd.singhritesh85.com --username admin --password Admin@123 --skip-test-tls --grpc-web'

sh 'argocd app create netflix-clone --project default --repo https://github.com/singhritesh85/helm-repo-for-netflix-clone.git --path ./folo --dest-namespace netflix --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=${REPLICA_COUNT} --upsert'

sh 'argocd app sync netflix-clone'

}
}

}

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'
}

}
}

}

```

To run the Jenkins Job, we need to provide some parameters (String Parameters in Jenkins) as shown in the screenshot attached below.

The screenshot shows the Jenkins Parameters screen for a job named 'test'. The left sidebar lists various Jenkins features like Status, Changes, Console Output, and Pipeline Overview. The 'Parameters' section is selected. It contains four input fields:

- COMMIT_ID**: A text input field containing a yellowed-out commit ID.
- REPO_NAME**: A text input field containing '02 [REDACTED] 6.dkr.ecr.us-east-2.amazonaws.com/netflix'.
- TAG_NAME**: A text input field containing '1.01'.
- REPLICA_COUNT**: A text input field containing '2'.

Before Running the Jenkins Job change 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@ [REDACTED] ~]# 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
```

After successful execution of Jenkins Job the screenshot for ArgoCD is as shown below.

The screenshot shows the ArgoCD Applications summary page. The left sidebar has links for Applications, Settings, User Info, Documentation, and Application filters. The main area displays application status metrics:

SUMMARY		Sync		Health	
APPLICATIONS	1	Unknown (0)	Synced (1)	Unknown (0)	Progressing (0)
SYNCED	1	Synced (1)		Suspended (0)	
HEALTHY	1			Healthy (1)	
CLUSTERS	1			Degraded (0)	
NAMESPACES	1			Missing (0)	

After successfully running the Jenkins Job, Pods had been created and the SonarQube UI are as shown in the screenshot attached below.

```
[jenkins@... ~]$ kubectl get pods -n netflix
NAME                      READY   STATUS    RESTARTS   AGE
netflix-clone-folo-5g     1/1     Running   0          32s
netflix-clone-folo-52     1/1     Running   0          32s
```

The screenshot shows the SonarQube interface with the 'Projects' tab selected. A search bar at the top right contains the placeholder 'Search for projects...'. Below it, a 'Create Project' button is visible. The main area displays a single project named 'netflix-clone' with a 'Passed' status. To the left, there are filters for Quality Gate (Passed, Failed), Reliability (Bugs: A rating 1, B rating 0, C rating 0, D rating 0, E rating 0), and Security (Vulnerabilities: A rating 1, B rating 0, C rating 0, D rating 0, E rating 0). At the bottom, a note says 'Last analysis: 57 minutes ago'.

To access the Netflix Clone Application, I created the ingress rule as discussed below.

```
[jenkins@... ~]$ cat ingress-rule.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netflix-ingress
  namespace: netflix
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  ingressClassName: nginx
  rules:
  - host: netflix-clone.singhritesh85.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: netflix-clone-folo
            port:
              number: 80
[jenkins@... ~]$ kubectl apply -f ingress-rule.yaml
ingress.networking.k8s.io/netflix-ingress created
[jenkins@... ~]$ kubectl get ing -n netflix
NAME      CLASS   HOSTS           ADDRESS          PORTS   AGE
netflix-ingress   nginx   netflix-clone.singhritesh85.com   a[REDACTED] 1.us-east-2.elb.amazonaws.com   80      12s
```

After Successful execution of Jenkins Job, I received two email one regarding a Jenkins Job had been executed and another Jenkins job had been executed successfully as shown in the screenshot attached below.

Jenkins Job test has been executed Inbox x

[\[REDACTED\]@gmail.com](#)

to me ▾

A Jenkins Job with Job Name test has been executed

Reply

Forward



Jenkins Job test has been Sucessfully Executed Inbox x

[\[REDACTED\]@gmail.com](#)

to me ▾

A Jenkins Job with Job Name test and Build Number=9 has been executed Successfully, Please Open the URL <https://jenkins-ms.singhritesh85.com/job/test/9/> and click on Console Output to see the Log. The Result of execution is SUCCESS

mar-

Reply

Forward



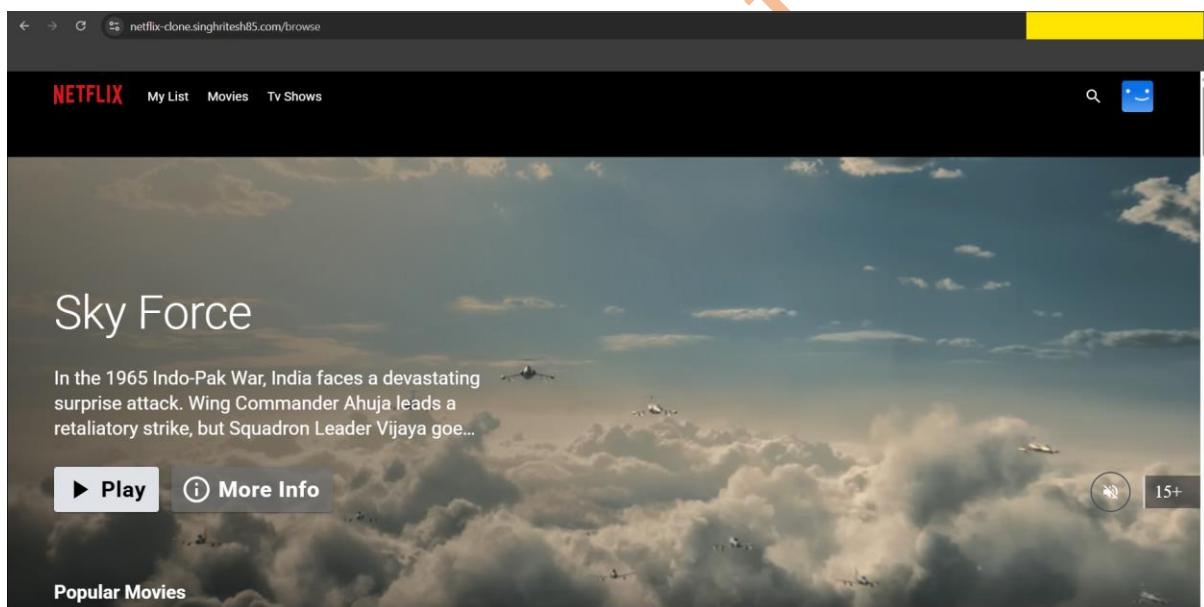
```
cat ingress-rule.yaml

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netflix-ingress
  namespace: netflix
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  ingressClassName: nginx
  rules:
  - host: netflix-clone.singhritesh85.com
    http:
      paths:
      - path: /
        pathType: Prefix
      backend:
        service:
          name: netflix-clone-folo
          port:
            number: 80
```

To access the Netflix clone Application through URL I did the entry of HOST of ingress corresponding to the DNS Name in Azure DNS Zone to create the record set of Type CNAME is as shown in the screenshot attached below.

The screenshot shows the AWS Route 53 'Records' page for the domain 'singhritesh85.com'. A modal window titled 'Add record set' is open, showing the configuration for a new CNAME record named 'netflix-clone'. The target is set to 'acm-validations.aws.' with a TTL of 1 second. The 'Alias' dropdown is set to 'No'. The main table lists several existing records with various TTL values.

Finally, I accessed the Netflix Clone application through the URL 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
```

```
[root@ ~]# kubectl create ns node-exporter
```

```
[root@ ~]# helm install my-prometheus-node-exporter prometheus-community/prometheus-node-exporter --version 4.37.1 --set service.type=LoadBalancer -n node-exporter
```

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   172.17.0.142   a.us-east-2.elb.amazonaws.com   9100:30804/TCP   2m19s
```

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

```
- job_name: "EKS"
  static_configs:
    - targets: ["a.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 Tue 2025-
```

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.████████:3100/loki/api/v1/push
  - url: http://10.████████:3100/loki/api/v1/push
  - url: http://10.████████: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@████████ ~]# git clone https://github.com/singhritesh85/helm-chart-promtail.git
[root@████████ ~]# kubectl create ns promtail && helm upgrade --install promtail ./helm-chart-promtail -f ./helm-chart-promtail/values.yaml -n promtail
[root@████████ ~]# kubectl get pods -n promtail --watch
NAME        READY   STATUS    RESTARTS   AGE
promtail-1  1/1     Running   0          2m11s
promtail-2  1/1     Running   0          2m11s

```

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

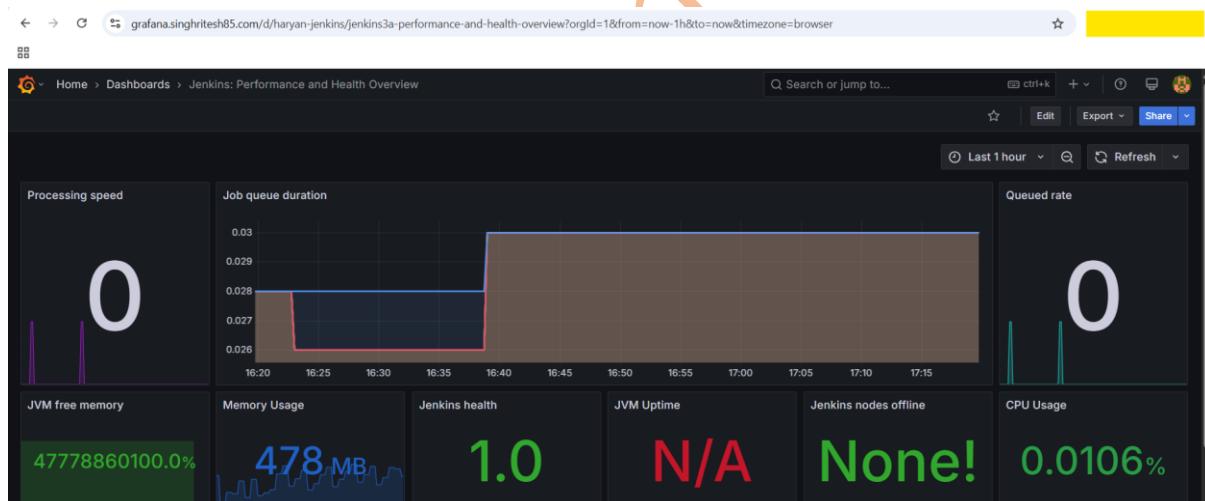
had provided the terraform script with this GitHub Repository. I had already integrated Prometheus and Loki as a data source for Grafana which was already discussed above.

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

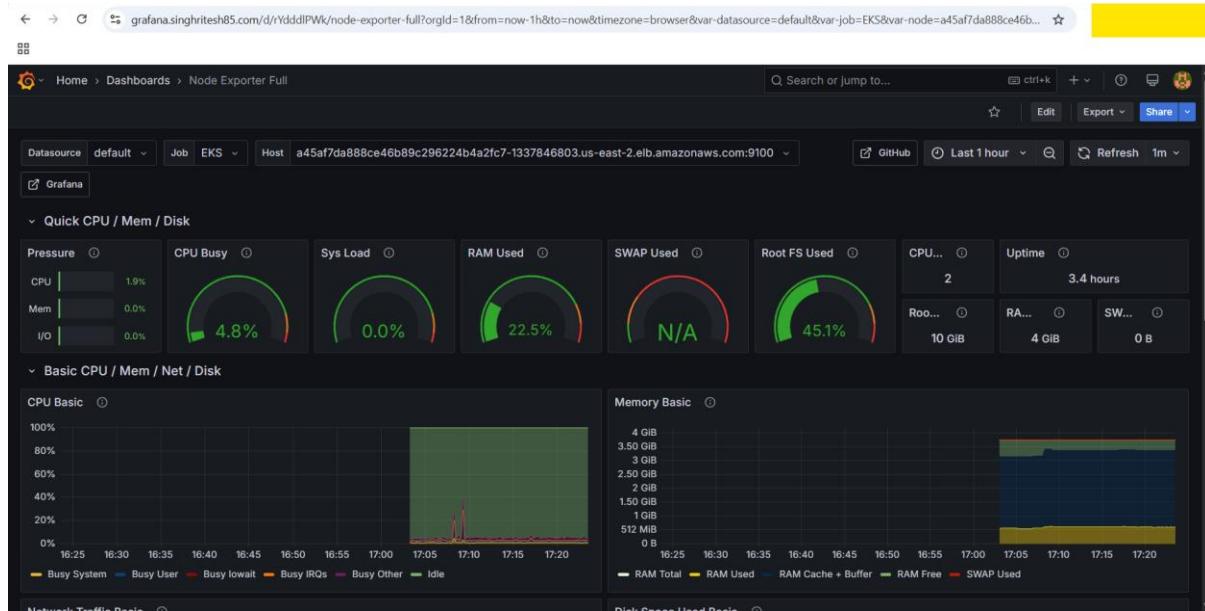
The screenshot shows the Prometheus Targets page with three sections:

- BlacboxExporter-Server (1/1 up)**: Shows one target at `http://10.10.4.101:9100/metrics` with state **UP**, instance `instance="10.10.4.101:9100"`, and job `job="BlacboxExporter-Server"`. Last scraped 1m 26s ago, scrape duration 14.454ms.
- EKS (1/1 up)**: Shows one target at `http://a45af7da888ce46b89c296224b4a2fc7-1337846803.us-east-2.elb.amazonaws.com:9100/metrics` with state **UP**, instance `instance="a45af7da888ce46b89c296224b4a2fc7-1337846803.us-east-2.elb.amazonaws.com:9100"`, and job `job="EKS"`. Last scraped 1m 16s ago, scrape duration 24.779ms.
- Grafana-Server (1/1 up)**: Shows one target at `http://10.10.4.169:9100/metrics` with state **UP**, instance `instance="10.10.4.169:9100"`, and job `job="Grafana-Server"`. Last scraped 1m 15s ago, scrape duration 13.000ms.

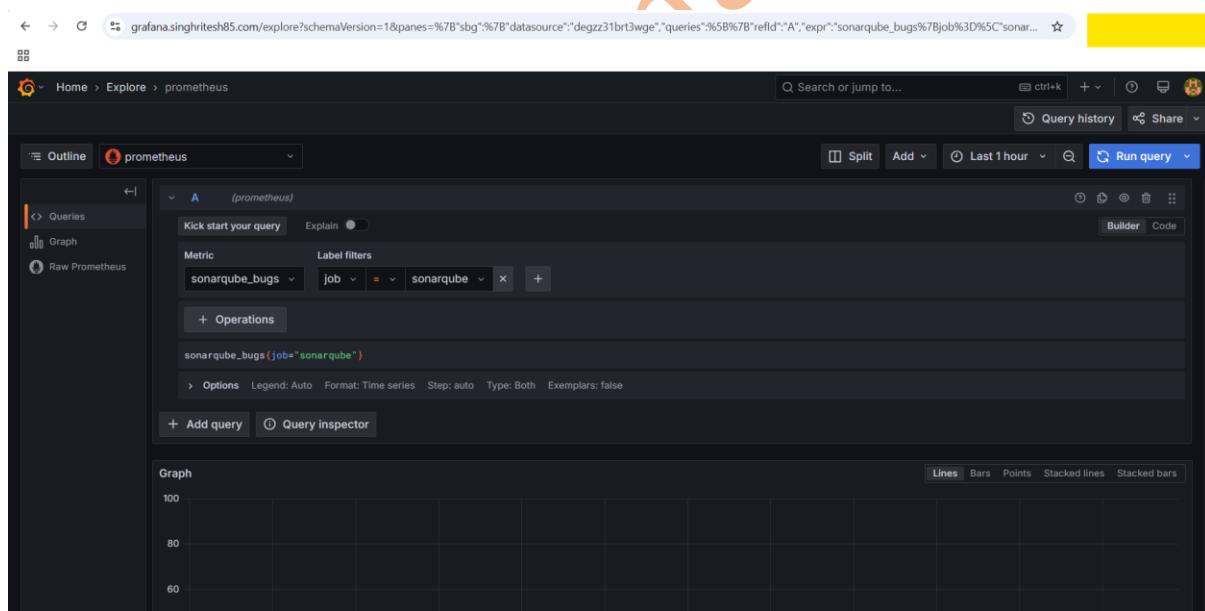
For Monitoring Jenkins Job using Prometheus I created the Grafana Dashboard using the Grafana ID **9964**.

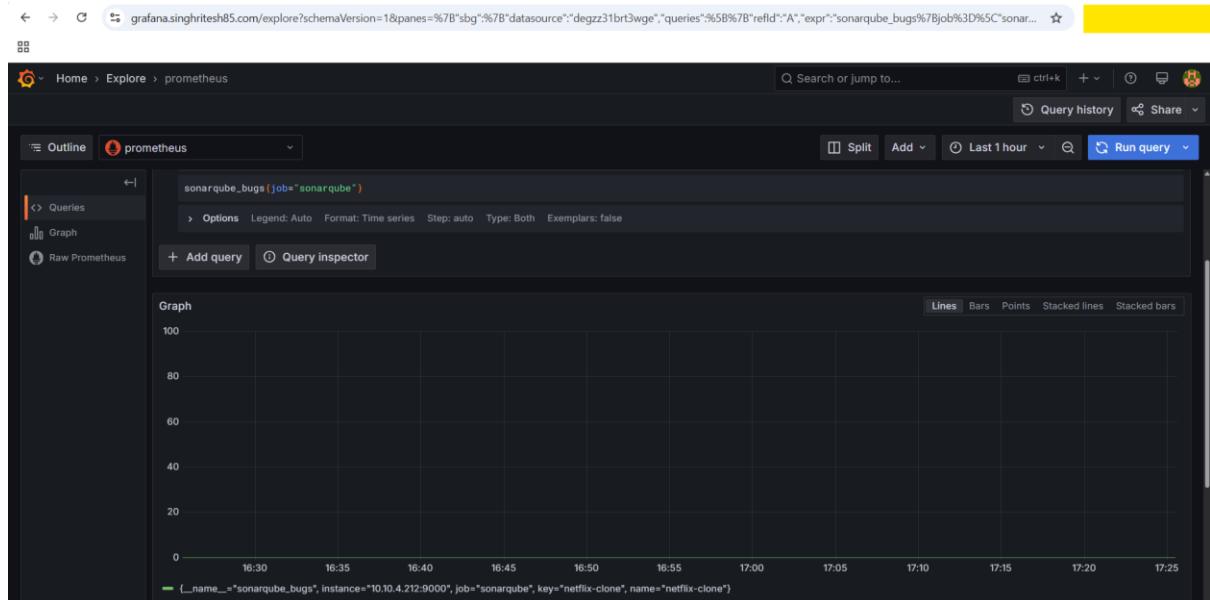


For Monitoring all the Servers and EKS Cluster health using the Node Exporter I used Grafana ID **1860**.

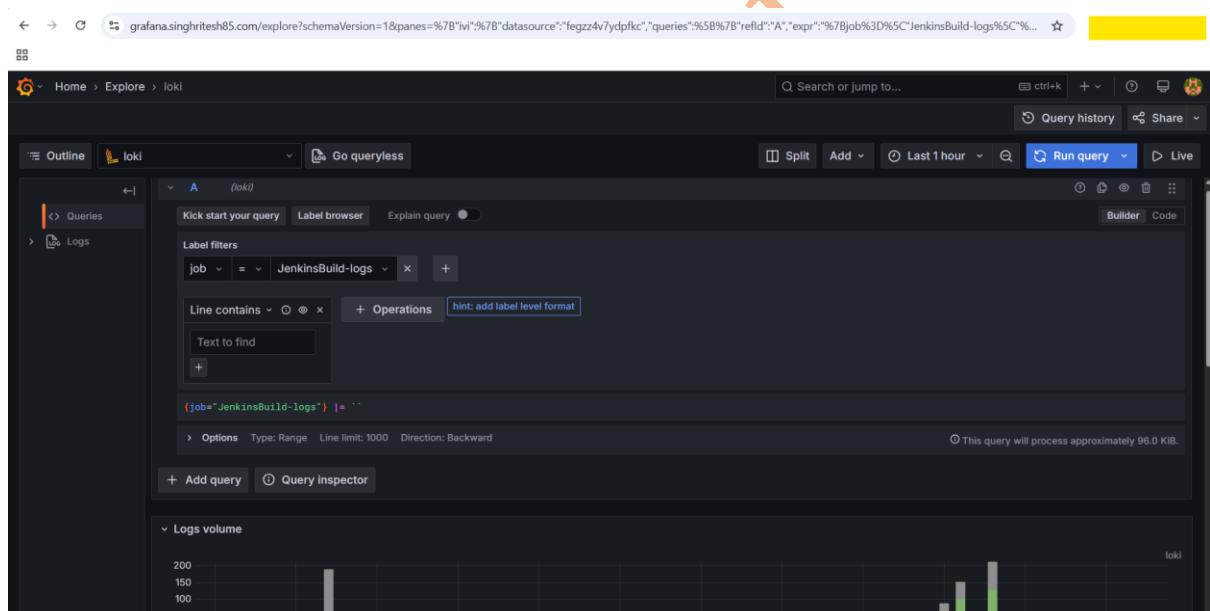


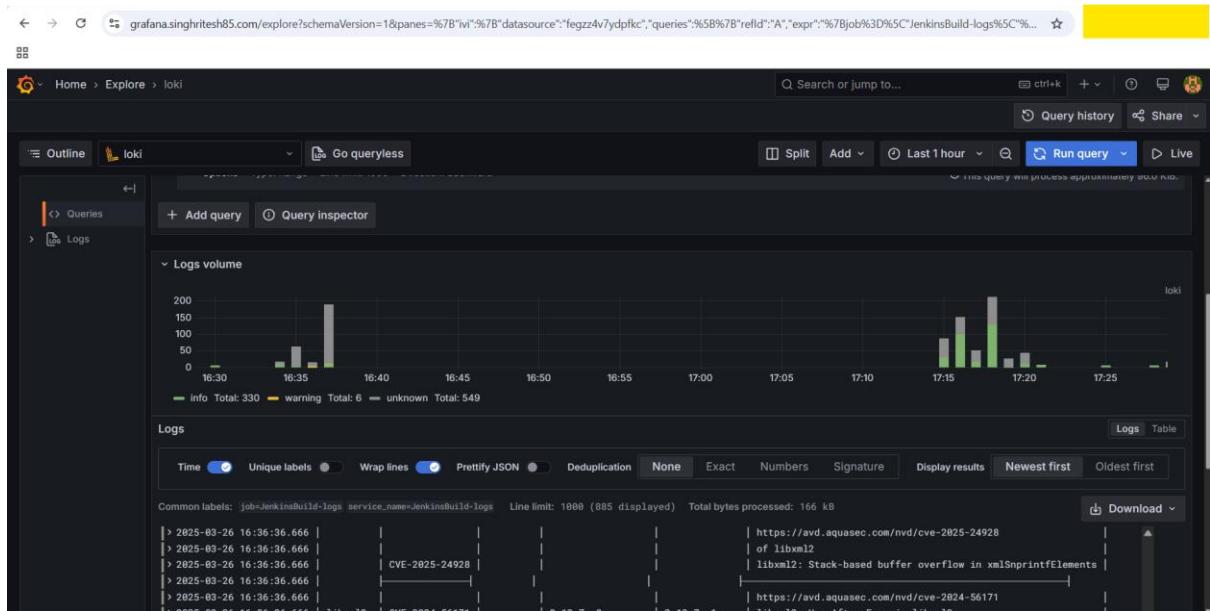
Grafana Metrics I started exploring as shown in the screenshot attached below.





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





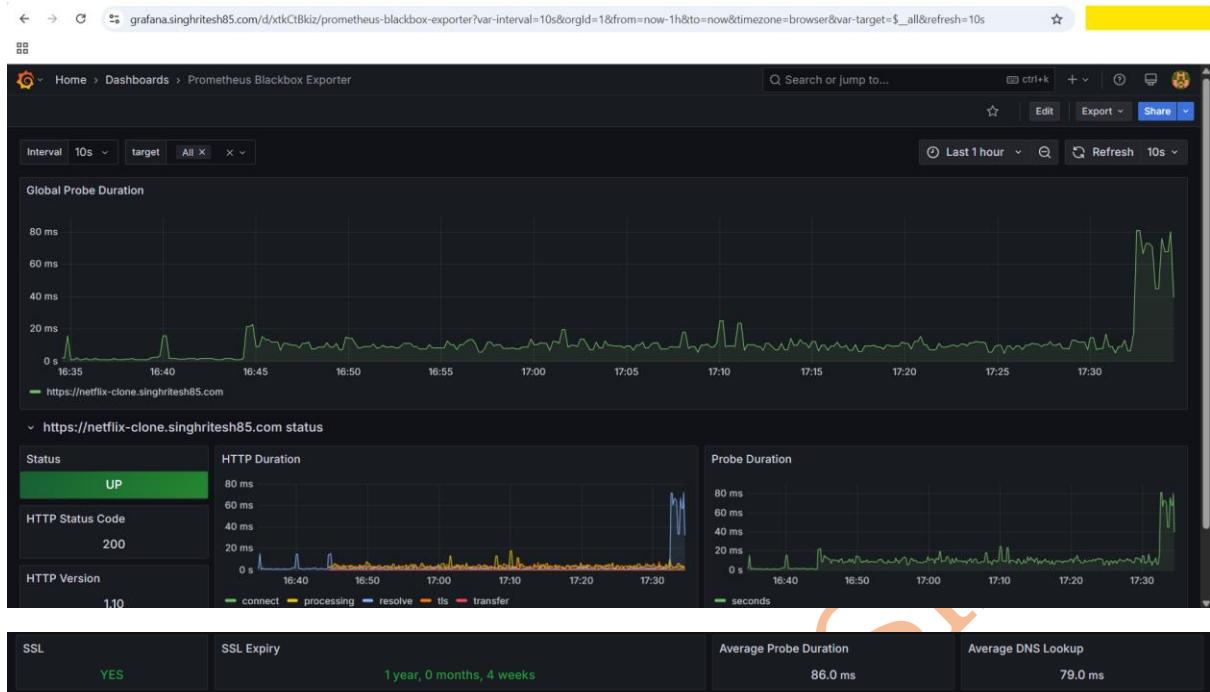
To achieve synthetic monitoring using Prometheus Blackbox Exporter I updated the `/etc/resolv.conf` file for Blackbox Exporter Server as shown in the screenshot attached below. I had used Google's Public DNS Server which is shown in the attached screenshot below.

```
[root@yellow ~]# 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 Netflix Clone Application URL as shown in the screenshot attached below. Application URL <https://netflix-clone.singhrites85.com> had been monitored using blackbox exporter.

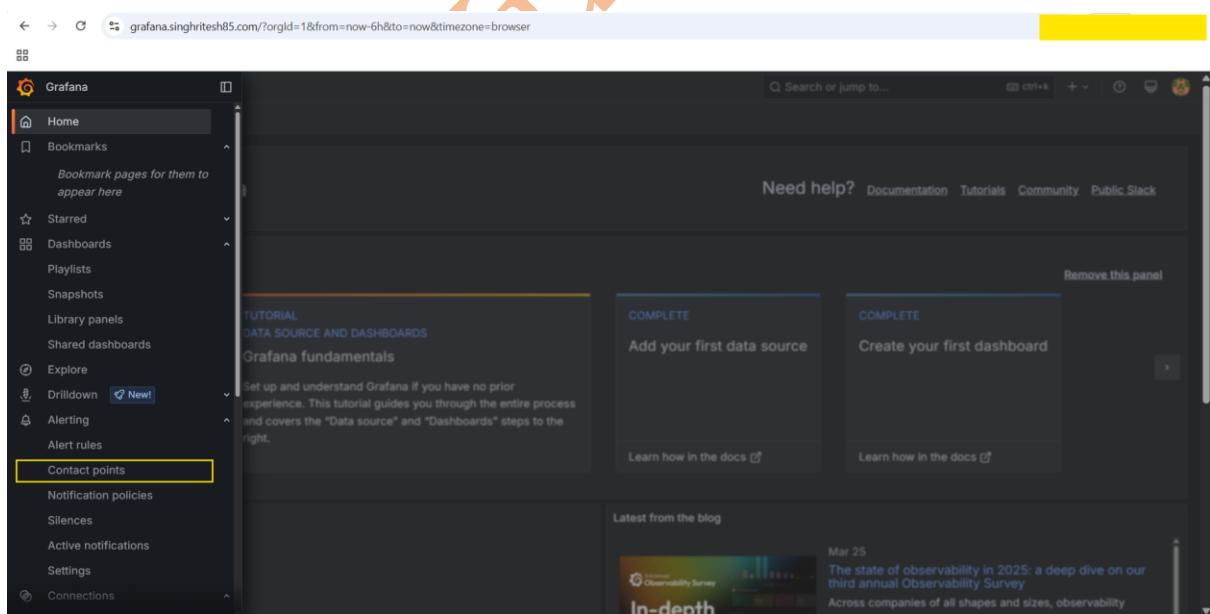
I had installed Blackbox Exporter on a different server and not on the Prometheus Server. The **module name** is `monitor_website.yml` present of the blackbox exporter server at the path `(/opt/blackbox_exporter_linux_amd64/monitor_website.yml)`. Prometheus blackbox operator is used for endpoint monitoring (Synthetic Monitoring) across the protocol http, https, TCP and ICMP. In this project I am monitoring the Application URL <https://netflix-clone.singhrites85.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.



The screenshot shows the 'Contact points' creation page in Grafana. A contact point named 'dexter' is being created with an 'Email' integration. The 'Addresses' field contains a single email address: [REDACTED]@gmail.com. There are tabs for 'Optional Email settings' and 'Notification settings'. A yellow box highlights the 'Test' button at the top right of the main form area.

The screenshot shows the 'Contact points' creation page again. A modal window titled 'Test contact point' is open, with the 'Predefined' tab selected. It displays a message: '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.' A yellow box highlights the 'Send test notification' button. A success message 'Test alert sent.' is visible in the background.

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

The screenshot shows the 'Notification policies' page in Grafana. A modal window titled 'Edit notification policy' is open, showing the 'Default contact point' set to 'dexter'. A yellow box highlights the 'Update default policy' button at the bottom of the modal.

Configure Alert Rule as shown in the screenshot attached below.

To create New Alerts
first click on + sign

1. Enter alert rule name
Enter a name to identify your alert rule.
Name
Therema

2. Define query and alert condition
Define query and alert condition Need help?
A prometheus Options 10 minutes Set as alert condition
Metric process_cpu_seconds_total
Label filters instance = 3.us-east-2.elb.amazonaws.com:9100
Operations hint: add rate
process_cpu_seconds_total(instance="e45af7da888ce46b89c296224b4a2fc7-1337846803.us-east-2.elb.amazonaws.com:9100")

Save rule and exit Cancel

process_cpu_seconds_total(instance="e45af7da888ce46b89c296224b4a2fc7-1337846803.us-east-2.elb.amazonaws.com:9100")

Add query

Rule type
 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
Input B
IS ABOVE 0.7
Custom recovery threshold

B Reduce Set "B" as alert condition
Input A
Function Last Mode Strict

Add expression Preview

grafana.singhritesh85.com/alerting/new

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.
 Creating 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?](#)

Select a folder before setting 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

[CPU time](#)

New folder
 Create a new folder to store your alert rule in.

Folder name
 [Create](#)

[Cancel](#) [Create](#)

5. Configure notifications
 Select who should receive a notification when an alert rule fires.

Recipient
 Notifications for firing alerts are routed to a selected contact point. [Need help?](#)

Alertmanager:  [grafana](#)

Contact point

5. Configure notifications
Select who should receive a notification when an alert rule fires.

Recipient
Notifications for firing alerts are routed to a selected contact point. [Need help?](#)

Alertmanager: **grafana**

Contact point: **dexter** [View or create contact points](#)

Email: singhritesh8515@gmail.com

Muting, grouping and timings (optional) ▾

6. Configure notification message
Add more context to your alert notifications. [Need help?](#)

Summary (optional)
Short summary of what happened and why.

Enter a summary...

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.

Alert rules
Rules that determine whether an alert will fire

Search by data sources [O](#) Dashboard State Rule type

All data sources Select dashboard Firing Normal Pending Alert Recording

Health Contact point

Ok No Data Error Choose

Search [Q](#) Search View as [Grouped](#) [List](#) [State](#)

1 rule **1 firing** [Export rules](#)

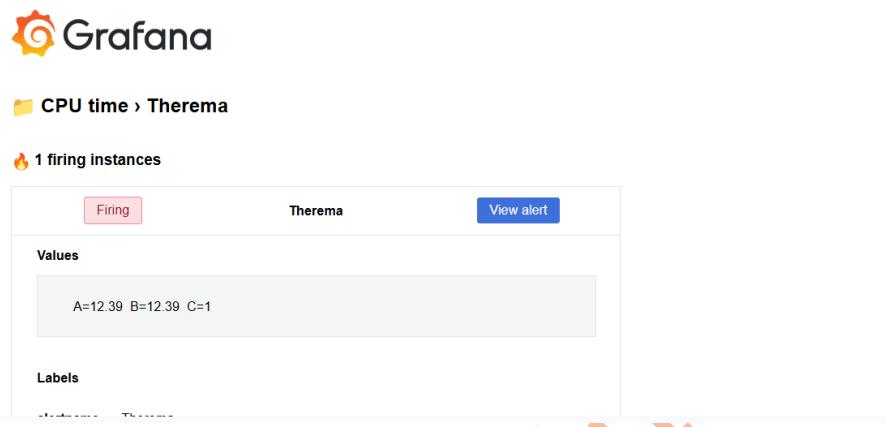
Grafana-managed

CPU time > CPU time

State	Name	Health	Summary	Next evaluation	Actions
Firing for 1m	Therema	ok	in a few seconds	View Edit More	

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

[FIRING:1] Therema CPU time (a45 [REDACTED] 03.us-east-2.elb.amazonaws.com:9100 EKS) [Inbox](#)



Source Code: - <https://github.com/singhritesh85/DevSecOps-Project.git>

GitHub Repo: - <https://github.com/singhritesh85/DevOps-Project-Netflix-Clone-Aws.git>

Helm Chart: - <https://github.com/singhritesh85/helm-repo-for-netflix-clone.git>

Terraform Script: - <https://github.com/singhritesh85/DevOps-Project-Netflix-Clone-Aws.git>

Reference: - <https://muditmathur121.medium.com/devsecops-netflix-clone-ci-cd-with-monitoring-email-990fbd115102>

Ritesh Kumar Singh