CI_CD_SONARQUBE_JFROG_EKS_JENKINS

Prerequisite

1) Need AWS Account
2) Need Terraform latest Version
3) AWS-CLI-v2
4) kubectl
5) Jenkins Server - create ec2 vm in aws and install jenkins

Jenkins server setup & installation

ssh login created new jenkins server

1) apt update

2) sudo apt install default-jre

3) wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

4) sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

5) sudo apt update

6) sudo apt install jenkins

7) sudo apt install docker.io

8) sudo usermod -aG docker jenkins

9) aws configure

10) kubectl install

11) `curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"`

```
12 ) sudo install -o root -g root -m  0755 kubectl /usr/local/bin/kubectl
```

```
ubuntu@ip-172-31-45-194:~$
ubuntu@ip-172-31-45-194:~$
ubuntu@ip-172-31-45-194:~$ sudo su jenkins
jenkins@ip-172-31-45-194:/home/ubuntu$ aws --version
aws-cli/2.7.35 Python/3.9.11 Linux/5.15.0-1019-aws exe/x86_64.ubuntu.20 prompt/off
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$ aws configure
AWS Access Key ID [None]: AKIAW3OSIJKALT5ICGE6
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
jenkins@ip-172-31-45-194:/home/ubuntu$
```

```
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
00   138  100   138    0     0    484      0 --:--:-- --:--:-- --:--:--   484
00 42.9M  100 42.9M    0     0  48.8M      0 --:--:-- --:--:-- --:--:--  184M
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$
buntu@ip-172-31-45-194:~$ kubectl version --client
ARNING: This version information is deprecated and will be replaced with the output from kubectl version --short.  Use --output=yaml|json to
et the full version.
lient Version: version.Info{Major:"1", Minor:"25", GitVersion:"v1.25.2", GitCommit:"5835544ca568b757a8ecae5c153f317e5736700e", GitTreeState:"
```

```
enkins@ip-172-31-45-194:/home/ubuntu$ aws configure
WS Access Key ID [****************CGE6]: AKIAZ53A5C4HFUDV7W46
WS Secret Access Key [****************YlHu]: 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
efault region name [ap-south-1]:
efault output format [None]:
enkins@ip-172-31-45-194:/home/ubuntu$ aws eks update-kubeconfig --name greens-cluster-cluster --region ap-south-1
dded new context arn:aws:eks:ap-south-1:682566162190:cluster/greens-cluster-cluster to /var/lib/jenkins/.kube/config
enkins@ip-172-31-45-194:/home/ubuntu$
enkins@ip-172-31-45-194:/home/ubuntu$
enkins@ip-172-31-45-194:/home/ubuntu$
enkins@ip-172-31-45-194:/home/ubuntu$
enkins@ip-172-31-45-194:/home/ubuntu$
enkins@ip-172-31-45-194:/home/ubuntu$
enkins@ip-172-31-45-194:/home/ubuntu$
enkins@ip-172-31-45-194:/home/ubuntu$ kubectl get nodes
AME                                           STATUS   ROLES    AGE   VERSION
p-172-31-34-185.ap-south-1.compute.internal   Ready    <none>   24h   v1.22.12-eks-ba74326
enkins@ip-172-31-45-194:/home/ubuntu$ client_loop: send disconnect: Broken pipe
```

INSTALLING SONAR-QUBE in AWS EKS CLUSTER

First aws configure in local

Then

aws eks update-kubeconfig –name cluster_name –region

INSTALLATION

```
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$ helm repo add bitnami https://charts.bitnami.com/bitnami
"bitnami" already exists with the same configuration, skipping
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$ helm install sonarqube bitnami/sonarqube
NAME: sonarqube
LAST DEPLOYED: Mon Sep 26 11:08:16 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
** Please be patient while the chart is being deployed **
```

```
2. Open a browser and access SonarQube using the obtained URL.

3. Login with the following credentials below:

  echo Username: user
  echo Password: $(kubectl get secret --namespace default sonarqube -o jsonpath="{.data.sonarqube-password}" | base64 -d)
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
```

```
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$ $(kubectl get secret --namespace default sonarqube -o jsonpath="{.data.sonarqube-password}" | base64 -d)
yMnfCHxDLA: command not found
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
```

You get the password yMnfCHxDLA and username user

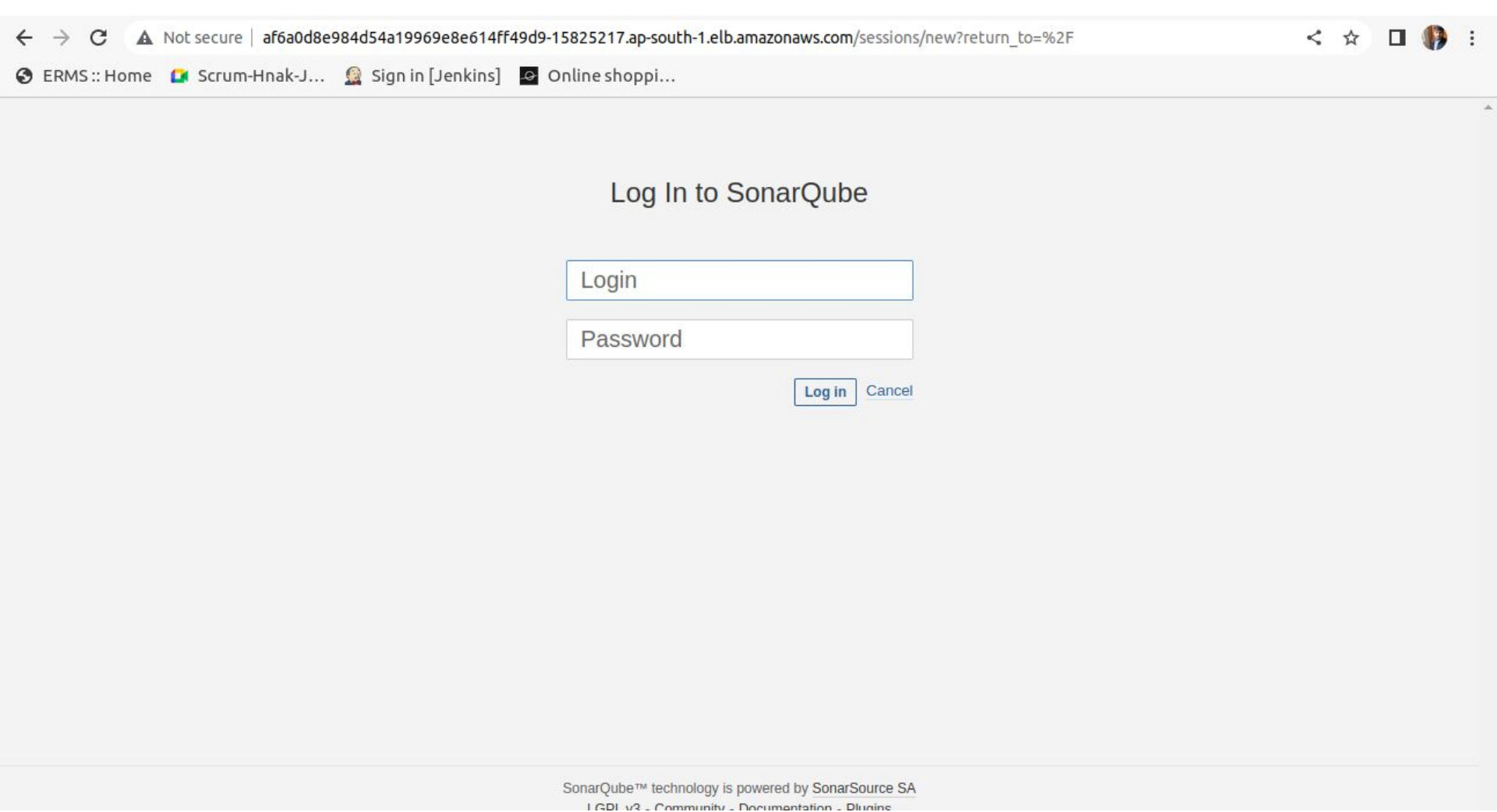
```
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
jenkins-0                     2/2     Running   0          14h
sonarqube-7466c89955-z8b29    1/1     Running   0          5m30s
sonarqube-postgresql-0        1/1     Running   0          5m30s
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
```

```
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$ kubectl get svc
NAME                     TYPE           CLUSTER-IP       EXTERNAL-IP                                                              PORT(S)
                  AGE
jenkins                  LoadBalancer   10.100.87.67     a31b7ccdbfdaa48a2abd8939e16635fd-222433920.ap-south-1.elb.amazonaws.com   8080:32268
/TCP              14h
jenkins-agent            ClusterIP      10.100.122.38    <none>                                                                   50000/TCP
                  14h
kubernetes               ClusterIP      10.100.0.1       <none>                                                                   443/TCP
                  17h
sonarqube                LoadBalancer   10.100.118.240   af6a0d8e984d54a19969e8e614ff49d9-15825217.ap-south-1.elb.amazonaws.com   80:31930/T
CP,9001:30936/TCP 6m23s
sonarqube-postgresql     ClusterIP      10.100.142.240   <none>                                                                   5432/TCP
                  6m23s
sonarqube-postgresql-hl  ClusterIP      None             <none>                                                                   5432/TCP
                  6m23s
prakash@prakash-Latitude-3420:~/prakash_hank/test-projects$
```

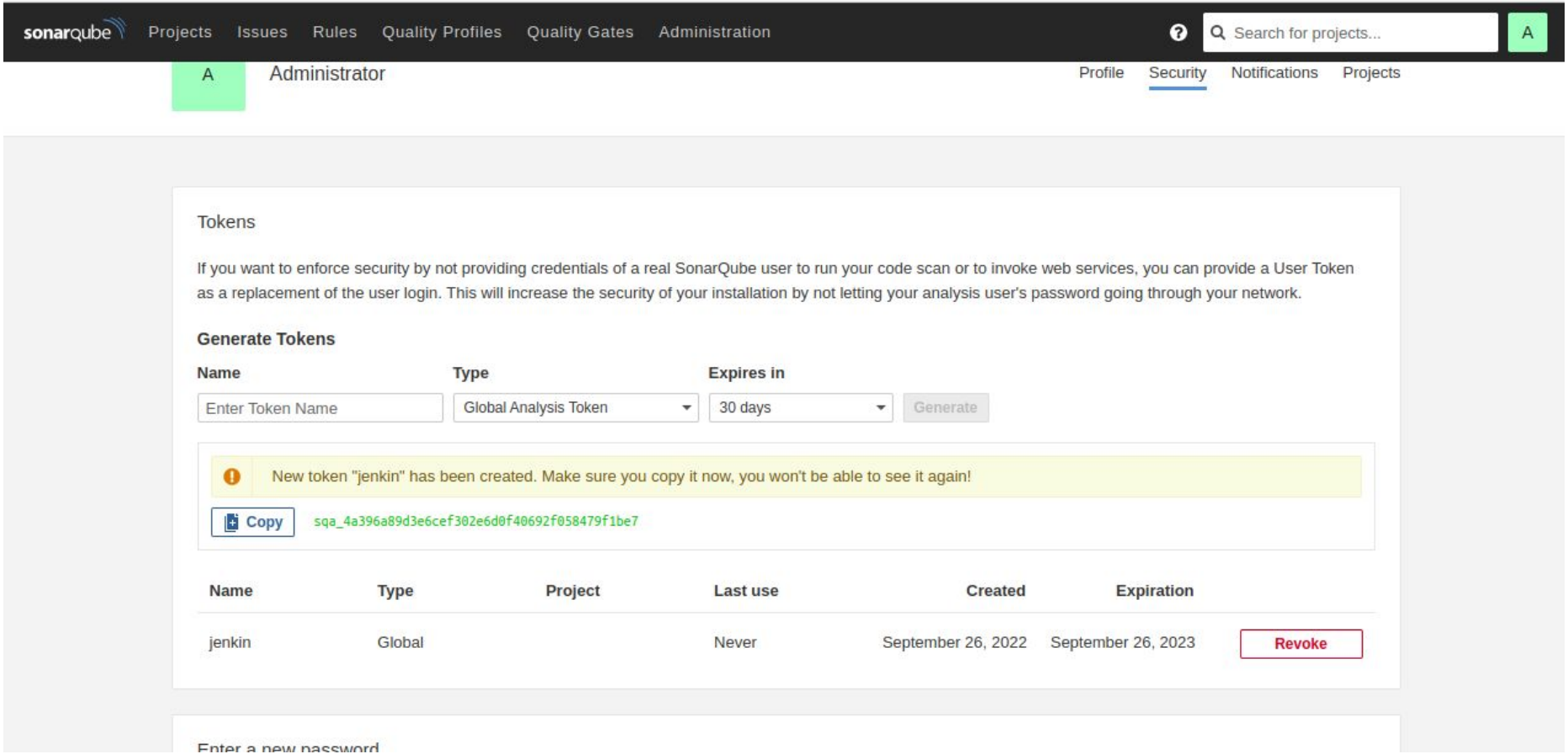
Here u get sonarqube loadbalancer elasticip

And go to browser

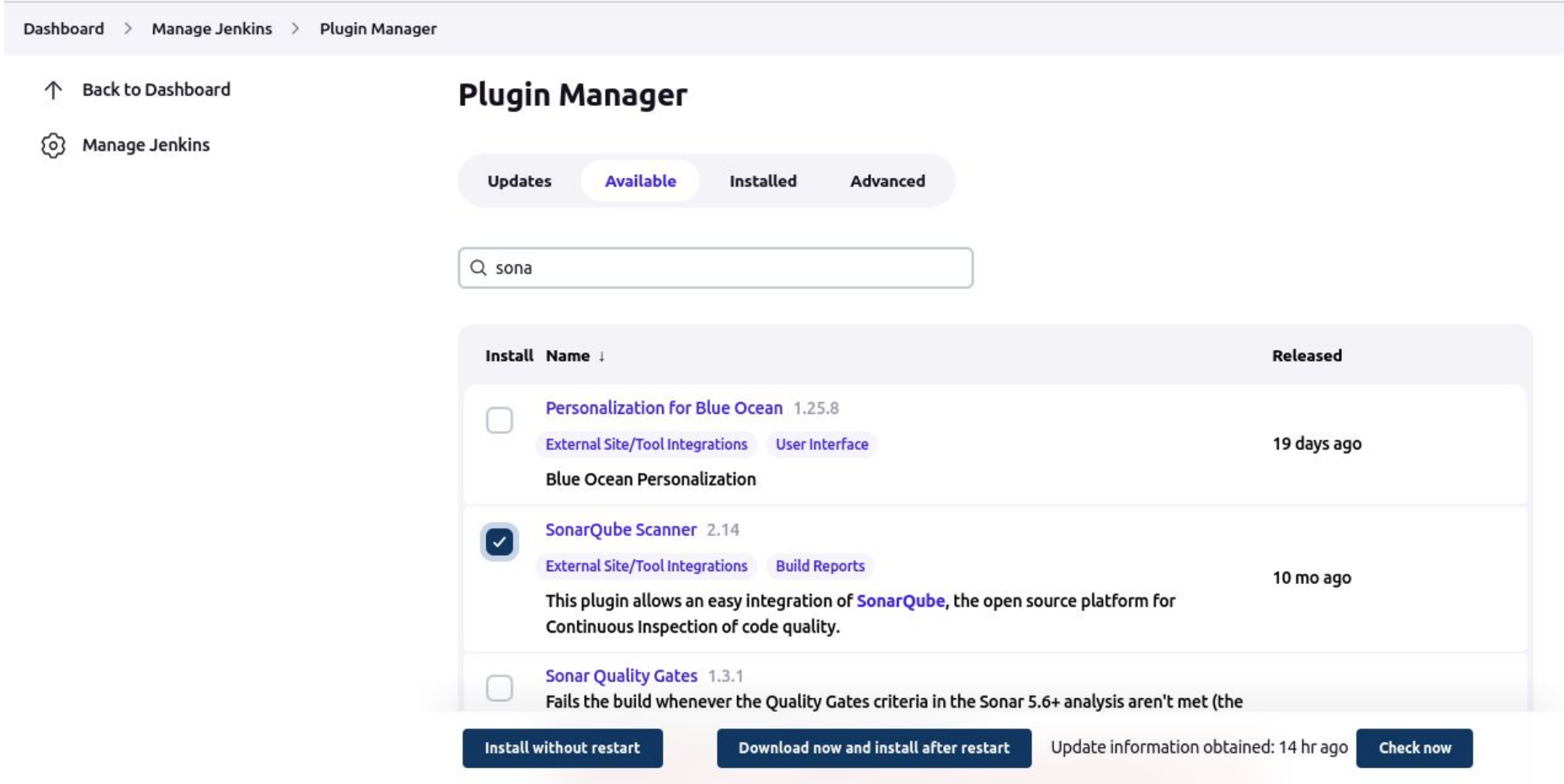
Paste the ip get the SonarQube UI


Not secure
af6a0d8e984d54a19969e8e614ff49d9-15825217.ap-south-1.elb.amazonaws.com/sessions/new?return_to=%2F
ERMS :: Home
Scrum-Hnak-J…
Sign in [Jenkins]
Online shoppi…
Log In to SonarQube
Login
Password
Log in
Cancel
SonarQube™ technology is powered by SonarSource SA
LGPL v3 - Community - Documentation - Plugins

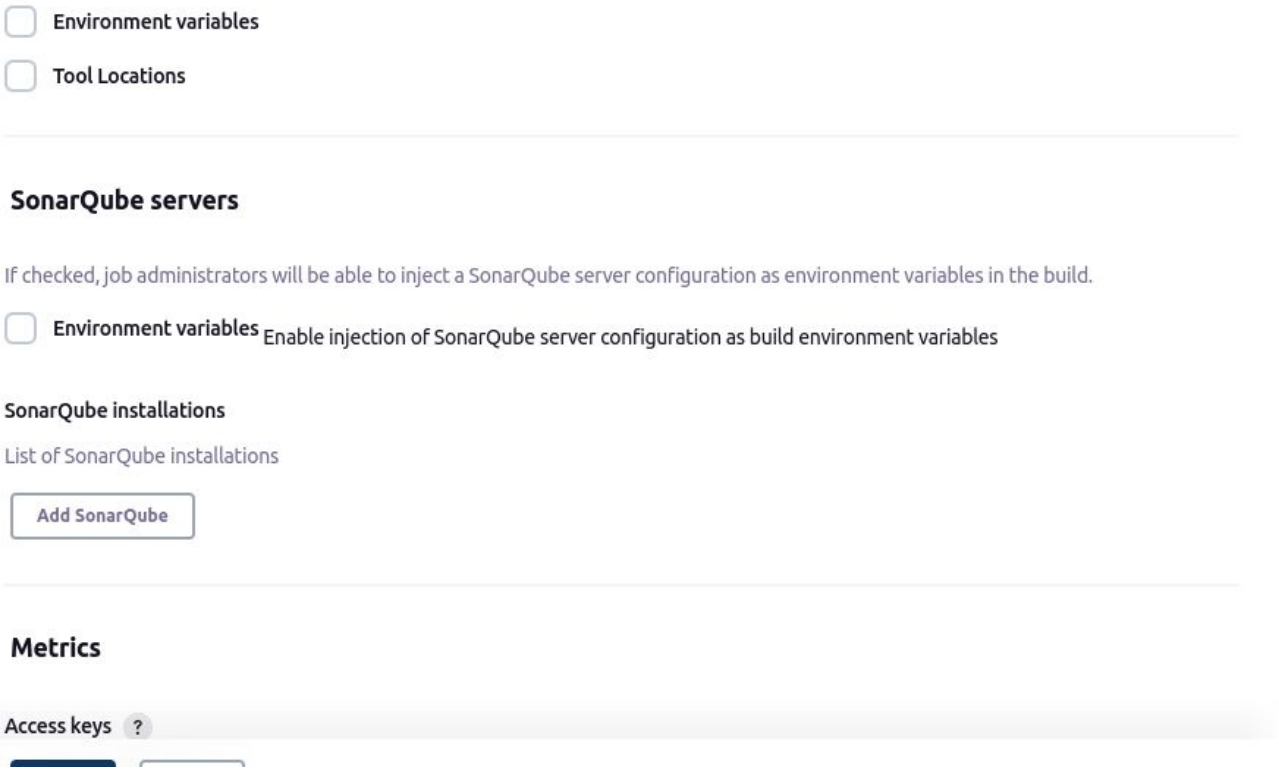
Go to profile settings generate tokens



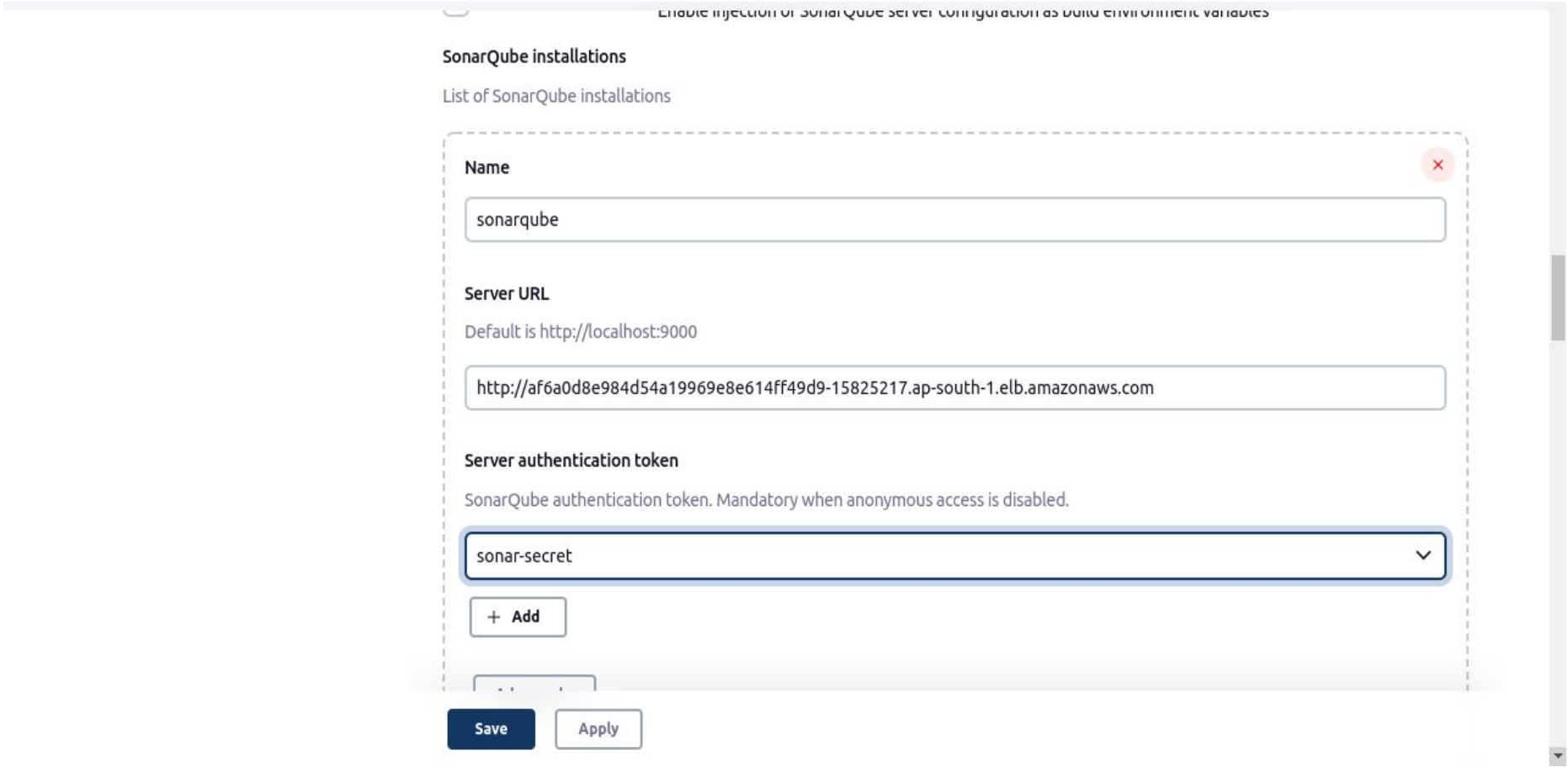
Go to jenkins plugin manager install SonarQube Scanner



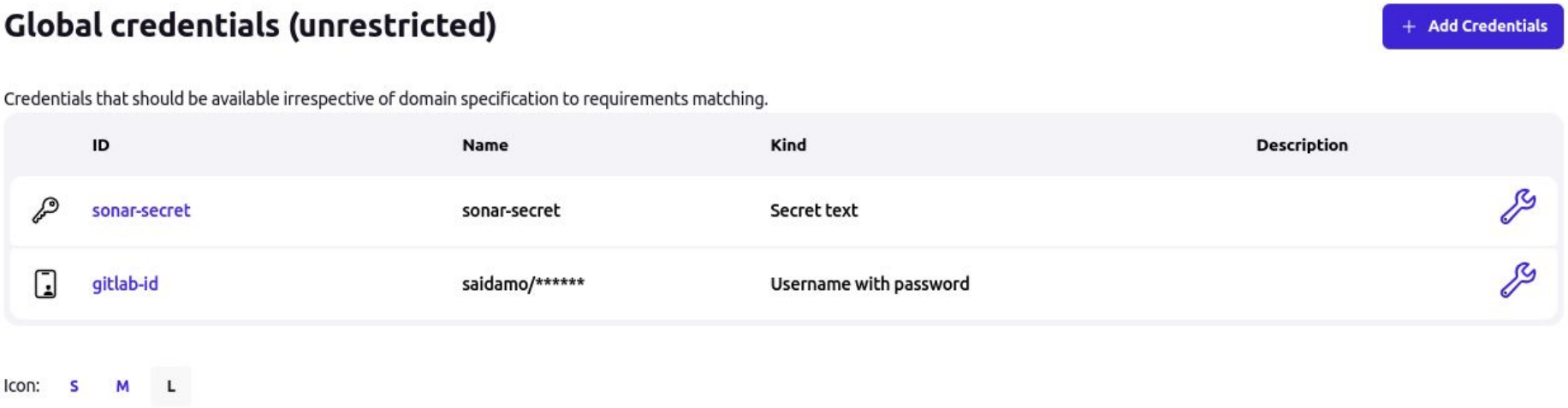
And go to jenkins global tool configuration and setup sonarqube server host and generated token



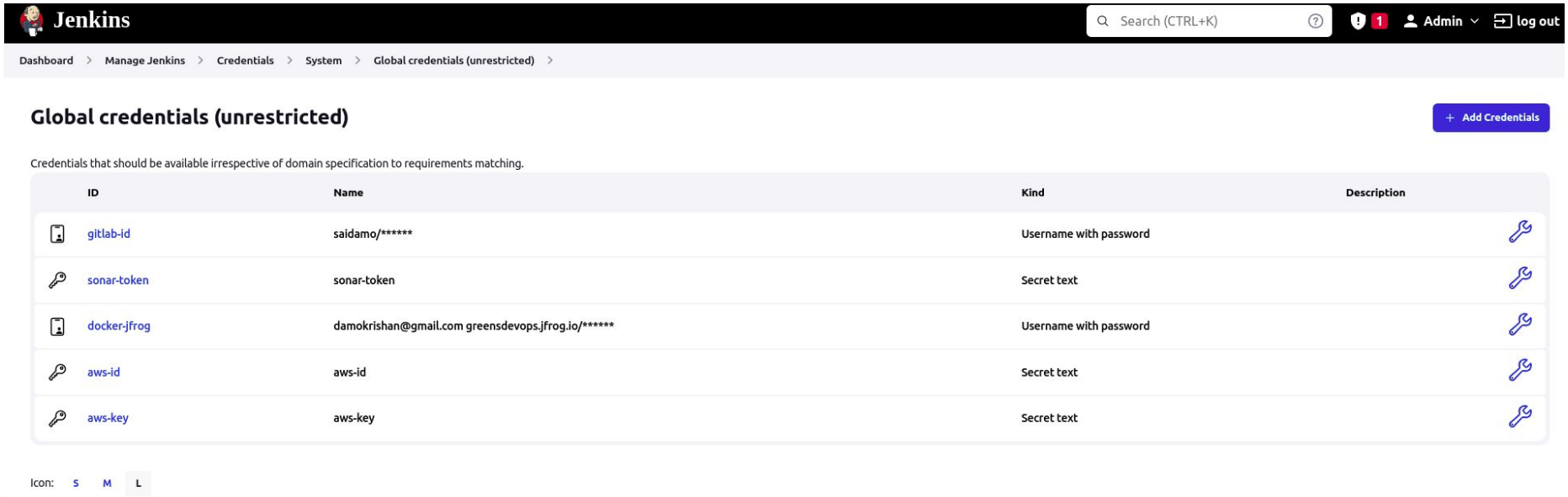
Give sonar qube generated token here


Enable injection of SonarQube server configuration as build environment variables
SonarQube installations
List of SonarQube installations
Name
sonarqube
Server URL
Default is http://localhost:9000
http://af6a0d8e984d54a19969e8e614ff49d9-15825217.ap-south-1.elb.amazonaws.com
Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.
sonar-secret
+ Add
Save
Apply

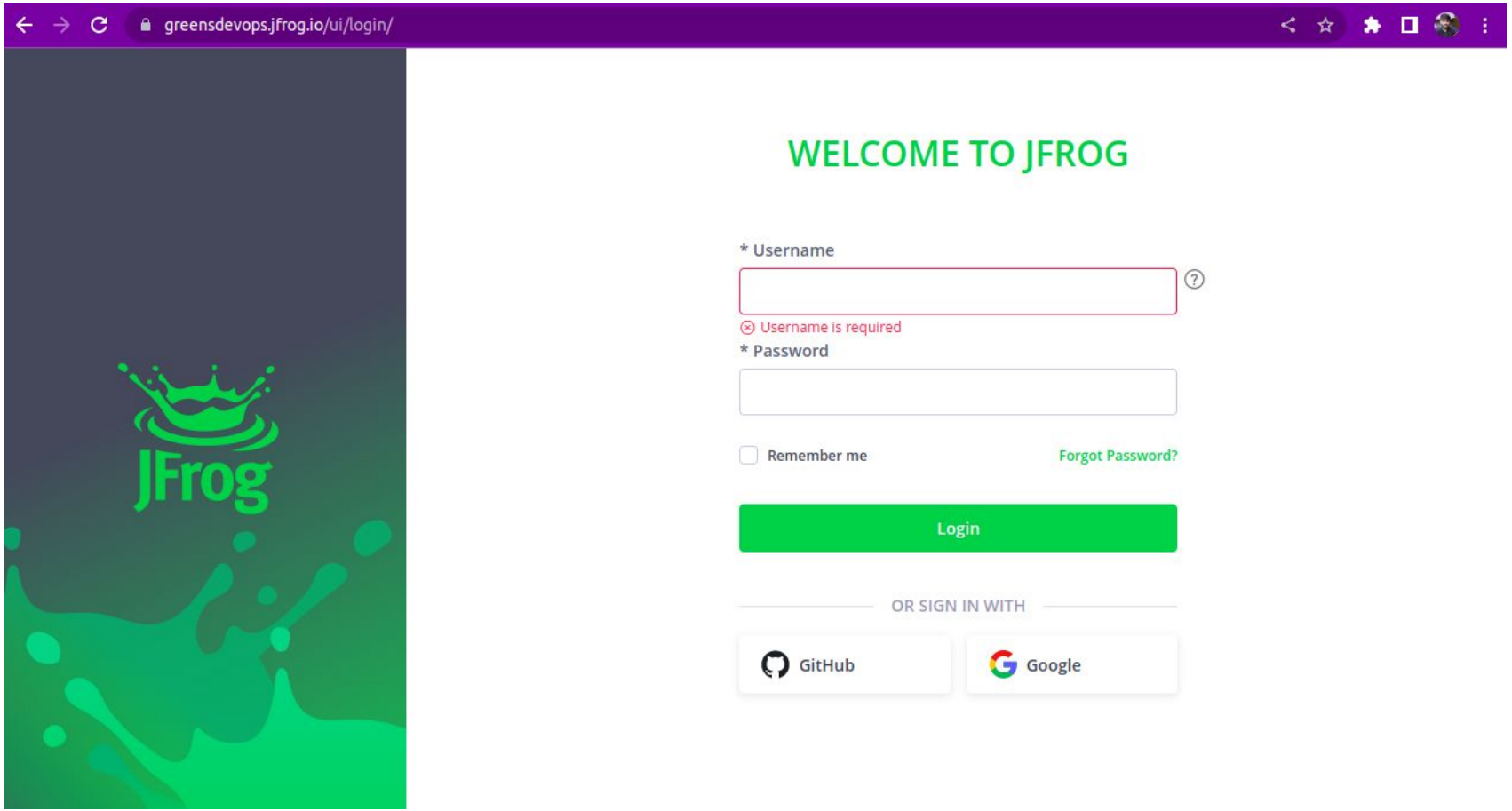
Store secrets like here

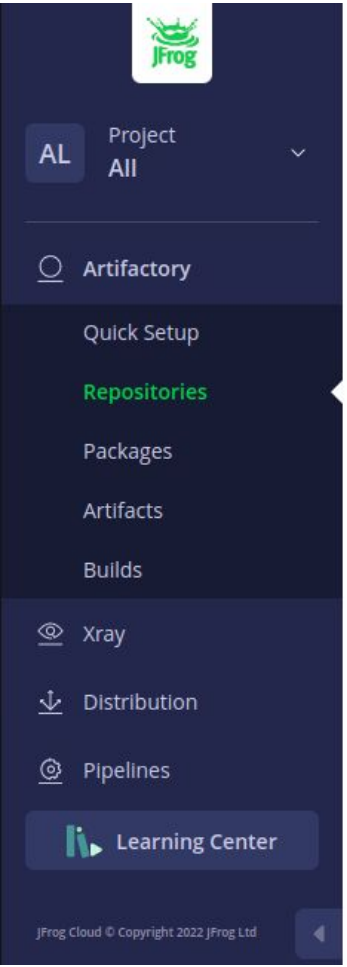


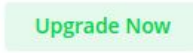
Store jFROG username like below with username docker url



Lets go to jfrog setup







Repositories

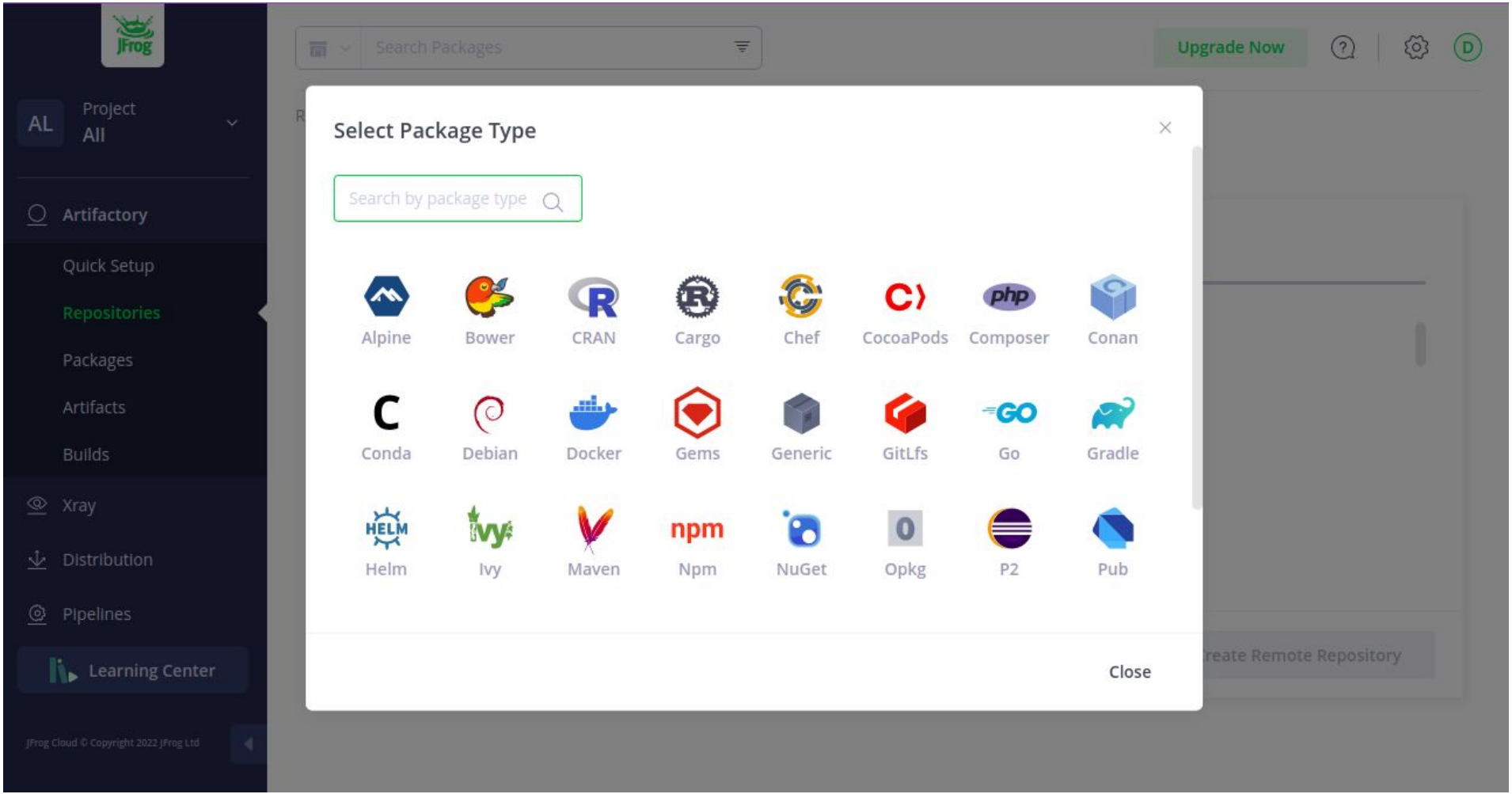
Local Remote Virtual Federated

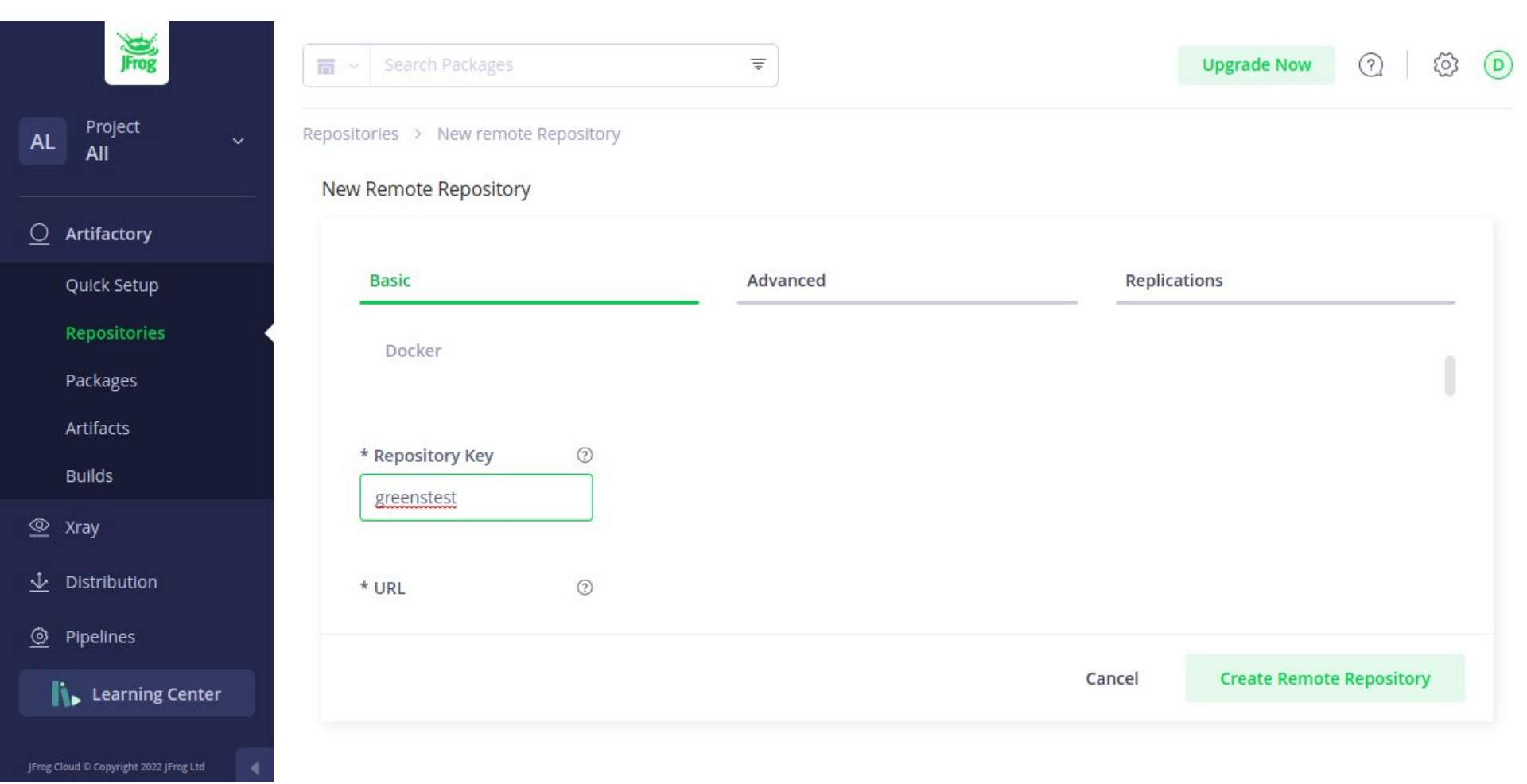
A remote repository serves as a caching proxy for a repository managed at a remote URL (which may itself be another Artifactory remote repository).

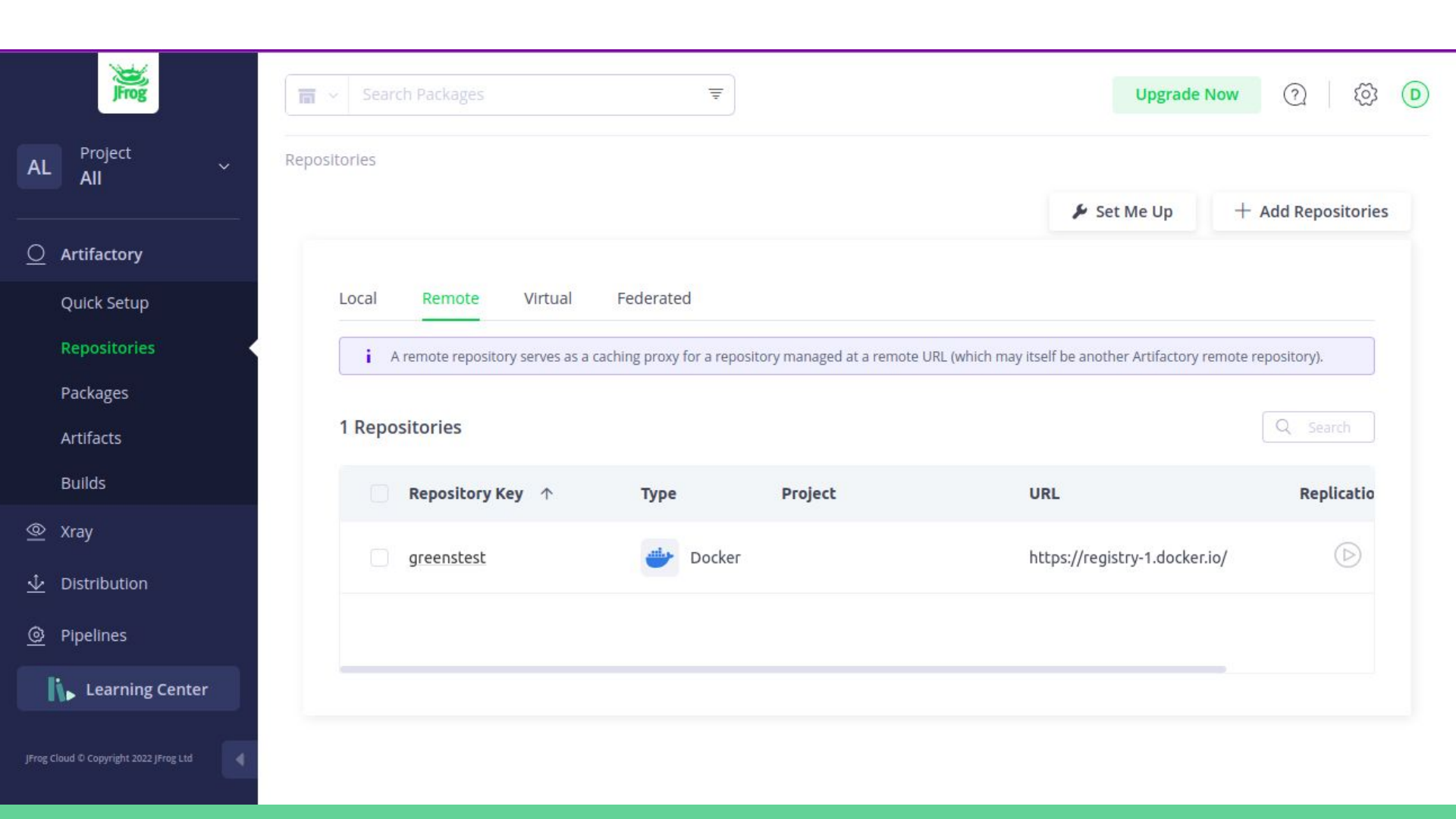
0 Repositories

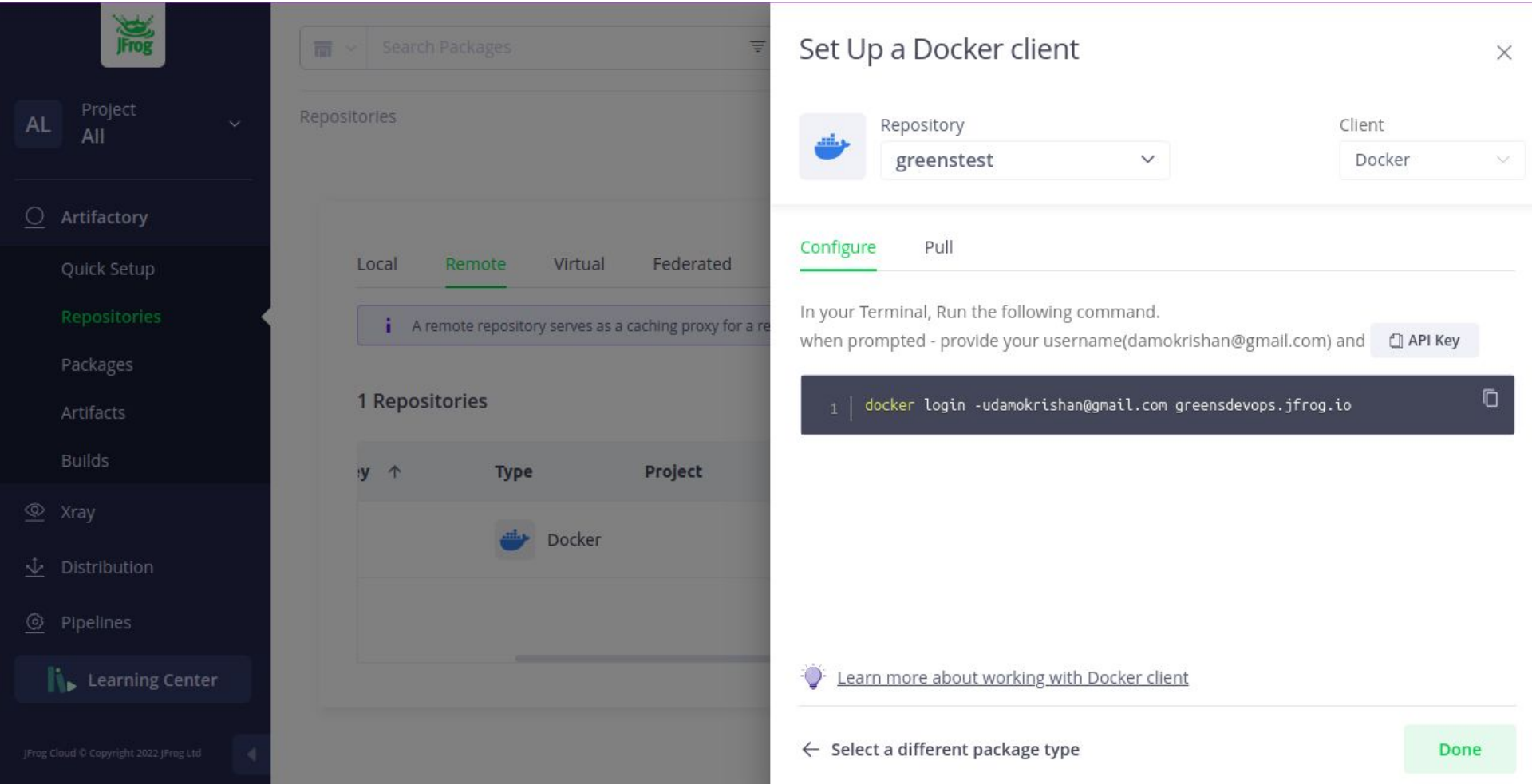
Repository Key	Type	Project	URL	Replica

No results were found
Try to change your search


Search Packages
Upgrade Now
AL
Project
All
Artifactory
Quick Setup
Repositories
Packages
Artifacts
Builds
Xray
Distribution
Pipelines
Learning Center
JFrog Cloud © Copyright 2022 JFrog Ltd
Select Package Type
Search by package type
Alpine
Bower
CRAN
Cargo
Chef
CocoaPods
Composer
Conan
Conda
Debian
Docker
Gems
Generic
GitLfs
Go
Gradle
Helm
Ivy
Maven
Npm
NuGet
Opkg
P2
Pub
Close
reate Remote Repository


Search Packages
Upgrade Now
D
AL
Project
All
Artifactory
Quick Setup
Repositories
Packages
Artifacts
Builds
Xray
Distribution
Pipelines
Learning Center
JFrog Cloud © Copyright 2022 JFrog Ltd
Repositories > New remote Repository
New Remote Repository
Basic
Advanced
Replications
Docker
* Repository Key
greenstest
* URL
Cancel
Create Remote Repository

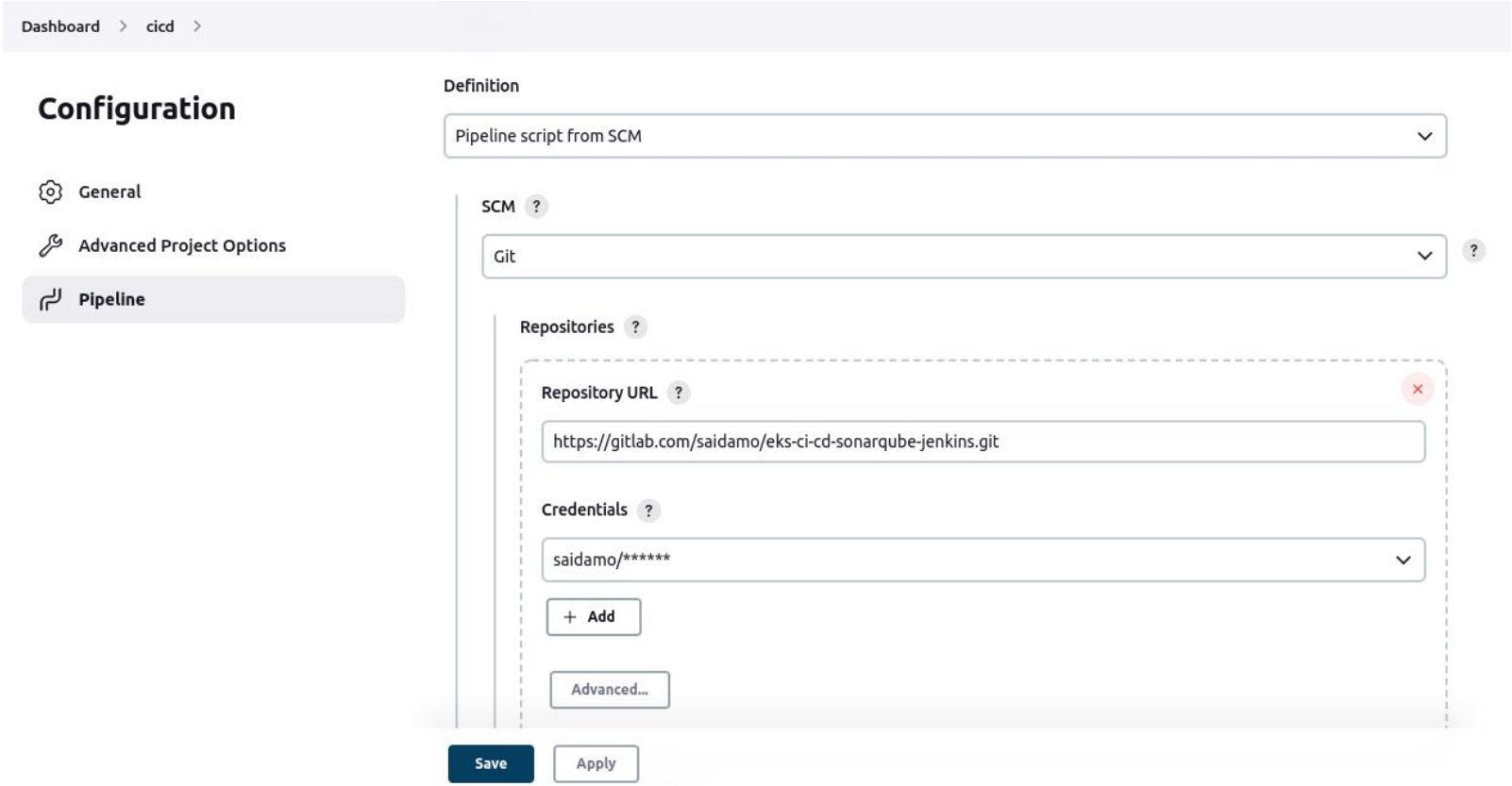

Search Packages
Upgrade Now
D
AL
Project
All
Repositories
Artifactory
Quick Setup
Repositories
Packages
Artifacts
Builds
Xray
Distribution
Pipelines
Learning Center
Set Me Up
Add Repositories
Local
Remote
Virtual
Federated
A remote repository serves as a caching proxy for a repository managed at a remote URL (which may itself be another Artifactory remote repository).
1 Repositories
Search
Repository Key
Type
Project
URL
Replicatio
greenstest
Docker
https://registry-1.docker.io/
JFrog Cloud © Copyright 2022 JFrog Ltd


Search Packages
Repositories
AL
Project
All
Artifactory
Quick Setup
Repositories
Packages
Artifacts
Builds
Xray
Distribution
Pipelines
Learning Center
JFrog Cloud © Copyright 2022 JFrog Ltd
Local
Remote
Virtual
Federated
A remote repository serves as a caching proxy for a re
1 Repositories
Type
Project
Docker
Set Up a Docker client
Repository
greenstest
Client
Docker
Configure
Pull
In your Terminal, Run the following command.
when prompted - provide your username(damokrishan@gmail.com) and
API Key
1 docker login -udamokrishan@gmail.com greensdevops.jfrog.io
Learn more about working with Docker client
Select a different package type
Done

Create private secret registry for docker

```
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$ kubectl create secret docker-registry privatecred --docker-server=greensdevops.jfrog.io --docker-us
ername=damokrishan@gmail.com --docker-password=AKCp8nGaHMJqWwZDSfty5bWj1VcxZ9X7Srhgmdk7jgDnfknUCSjeLCJTKabrGk2TqvHuynTPG --docker-email=damokr
ishan@gmail.com
secret/privatecred created
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
```

Go to jenkins
Create new pipeline



Finally i put the code and jenkins file inside the gitlab please refer
After deployment completed u get svc load balancer like below

```
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$ kubectl get pods
NAME                          READY   STATUS        RESTARTS   AGE
green-eks-5fd68f7846-8j7w8    1/1     Running       0          24s
green-eks-67cbf58579-sv7wx    1/1     Terminating   0          8m5s
sonarqube-6cc47bbf68-ts48m    1/1     Running       0          27h
sonarqube-postgresql-0        1/1     Running       0          27h
prakash@prakash-Latitude-3420:~/Downloads$ kubectl get svc
NAME                     TYPE           CLUSTER-IP       EXTERNAL-IP                                                                PORT(S)
                  AGE
green-eks                LoadBalancer   10.100.129.138   ac6ab9800b15349da85f061acfd056b6-1312106747.ap-south-1.elb.amazonaws.com   80:32520/
TCP,443:32673/TCP    21m
kubernetes               ClusterIP      10.100.0.1       <none>                                                                     443/TCP
                  27h
sonarqube                LoadBalancer   10.100.254.151   a0d7e5425c6f24b65beca8dc46032860-718461882.ap-south-1.elb.amazonaws.com    80:31358/
TCP,9001:32556/TCP   27h
sonarqube-postgresql     ClusterIP      10.100.188.139   <none>                                                                     5432/TCP
                  27h
sonarqube-postgresql-hl  ClusterIP      None             <none>                                                                     5432/TCP
                  27h
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
prakash@prakash-Latitude-3420:~/Downloads$
```

Thanks