

Infrastructure

Start by creating the infrastructure using Terraform on AWS environment. First we will run the terraform script to create the Kubernetes cluster (Master & 2 Worker Nodes) in addition to the Monitoring instance.

Terraform

Providers

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 5.0" # Use the latest version that suits your needs
    }
  }
}

provider "aws" {
  region = "us-east-1"
}
```

Use the default VPC:

```
data "aws_vpc" "Default-VPC" {
  default = true
}
```

Generate a Key Pair and exporting the DEPI-KeyPair.pem file:

```
resource "tls_private_key" "DEPI-Key" {
  algorithm = "RSA"
  rsa_bits = 4096
}
```

Create the key pair using the public key generated above

```
resource "aws_key_pair" "DEPI-KeyPair" {
  key_name = "DEPI-KeyPair"
  public_key = tls_private_key.DEPI-Key.public_key_openssh
}
```

Create a local file to save the private key

```
resource "local_file" "KeyPair" {
  content = tls_private_key.DEPI-Key.private_key_pem
  filename = "DEPI-KeyPair.pem"
}
```

Output the private key path

```
output "private_key_path" {
  value = local_file.KeyPair.filename
}
```

```
}
```

Configure the Security Group for the instances:

```
resource "aws_security_group" "DEPI-SecurityGroup" {  
  vpc_id = data.aws_vpc.Default-VPC.id  
  name = "DEPI-SecurityGroup"  
  ingress { #To access the VMs via SSH  
    from_port = 22  
    to_port = 22  
    protocol = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

```
  ingress { #For email notifications (will not be used in the project)  
    from_port = 25  
    to_port = 25  
    protocol = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

```
  ingress { #Range used for most of the applications  
    from_port = 3000  
    to_port = 10000  
    protocol = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

```
  ingress { #HTTP  
    from_port = 80  
    to_port = 80  
    protocol = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

```
  ingress { #HTTPS  
    from_port = 443  
    to_port = 443  
    protocol = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

```
  ingress { #Required when setting up Kubernetes cluster  
    from_port = 6443  
    to_port = 6443  
    protocol = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}
```

```

ingress { #Range used to send mail notification from our Jenkins pipeline to our gmail address
    from_port = 465
    to_port = 465
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
}

ingress { #Range used for deployment of applications
    from_port = 30000
    to_port = 32767
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
}

egress {
    from_port = 0
    to_port = 0
    protocol = "-1"
    cidr_blocks = ["0.0.0.0/0"]
}

}

```

Kubernetes Cluster

```

# Create the EC2 instances
resource "aws_instance" "Master" {
    ami          = "ami-0866a3c8686eaeeba"
    instance_type = "t2.micro"
    key_name     = aws_key_pair.DEPI-KeyPair.key_name
    tags = {
        Name = "Master"
    }
}

resource "aws_instance" "Slave-01" {
    ami          = "ami-0866a3c8686eaeeba"
    instance_type = "t2.micro"
    key_name     = aws_key_pair.DEPI-KeyPair.key_name
    tags = {
        Name = "Slave-01"
    }
}

resource "aws_instance" "Slave-02" {
    ami          = "ami-0866a3c8686eaeeba"
    instance_type = "t2.micro"
    key_name     = aws_key_pair.DEPI-KeyPair.key_name

```

```
tags = {
  Name = "Slave-02"
}
```

Monitoring Server

```
resource "aws_instance" "Monitoring" {
  ami          = "ami-0866a3c8686eaeeba"
  instance_type = "t2.micro"
  key_name     = aws_key_pair.DEPI-KeyPair.key_name
  tags = {
    Name = "Monitoring"
  }
}
```

Terraform apply

1. Use the default VPC

The screenshot shows the AWS VPC console with a table of VPCs. The table has columns: Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP option set, Main route table, and Main DNS attributes. One VPC is listed: DEPI-VPC with ID vpc-0287596bada3bdad4d, state Available, IPv4 CIDR 172.31.0.0/16, and DHCP option set dopc-094aa4a1c9524cbf7.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Main DNS attributes
DEPI-VPC	vpc-0287596bada3bdad4d	Available	172.31.0.0/16	-	dopc-094aa4a1c9524cbf7	rtb-0e95a9ada13cb9ce7	-

2. Key Pair

The screenshot shows the AWS Key Pairs console with a table of key pairs. The table has columns: Name, Type, Created, Fingerprint, and ID. One key pair is listed: DEPI-KeyPair with type rsa, created on 2024/10/12 20:19 GMT+3, fingerprint d5+f9c4-9b1a1c3e09c7850-10-48-bae9-26-27, and ID key-022c3488526d01940.

Name	Type	Created	Fingerprint	ID
DEPI-KeyPair	rsa	2024/10/12 20:19 GMT+3	d5+f9c4-9b1a1c3e09c7850-10-48-bae9-26-27	key-022c3488526d01940

3. Security Group

The screenshot shows the AWS Security Groups console with a table of security groups. The table has columns: Name, Security group ID, Security group name, VPC ID, Description, and Owner. Two security groups are listed: DEPI-SecurityGroup (sg-0237e3e3956084435d) and default (sg-00a2d77e5fa4e90c3). Below the table, the inbound rules for the DEPI-SecurityGroup are shown, including rules for Custom TCP, SSH, HTTPS, SMTP, HTTP, and Custom TCP.

Name	Security group ID	Security group name	VPC ID	Description	Owner
DEPI-SecurityGroup	sg-0237e3e3956084435d	DEPI-SecurityGroup	vpc-0287596bada3bdad4d	Managed by Terraform	761018886548
default	sg-00a2d77e5fa4e90c3	default	vpc-0287596bada3bdad4d	default VPC security group	761018886548

4. EC2 instances created using Terraform

The screenshot shows the AWS EC2 console with a table of instances. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 address, and Elastic IP. Four instances are listed: Monitoring, Slave-02, Slave-01, and Master, all in a running state.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address	Elastic IP
Monitoring	i-0cf37f47429a1d619	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-18-209-31-67.com...	18.209.31.67	-
Slave-02	i-02e19c12ae417437d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-18-212-145-159.co...	18.212.145.159	-
Slave-01	i-048dac6dbd78cca2b	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-54-144-40-75.com...	54.144.40.75	-
Master	i-034de18356a504311	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-54-209-235-133.co...	54.209.235.133	-

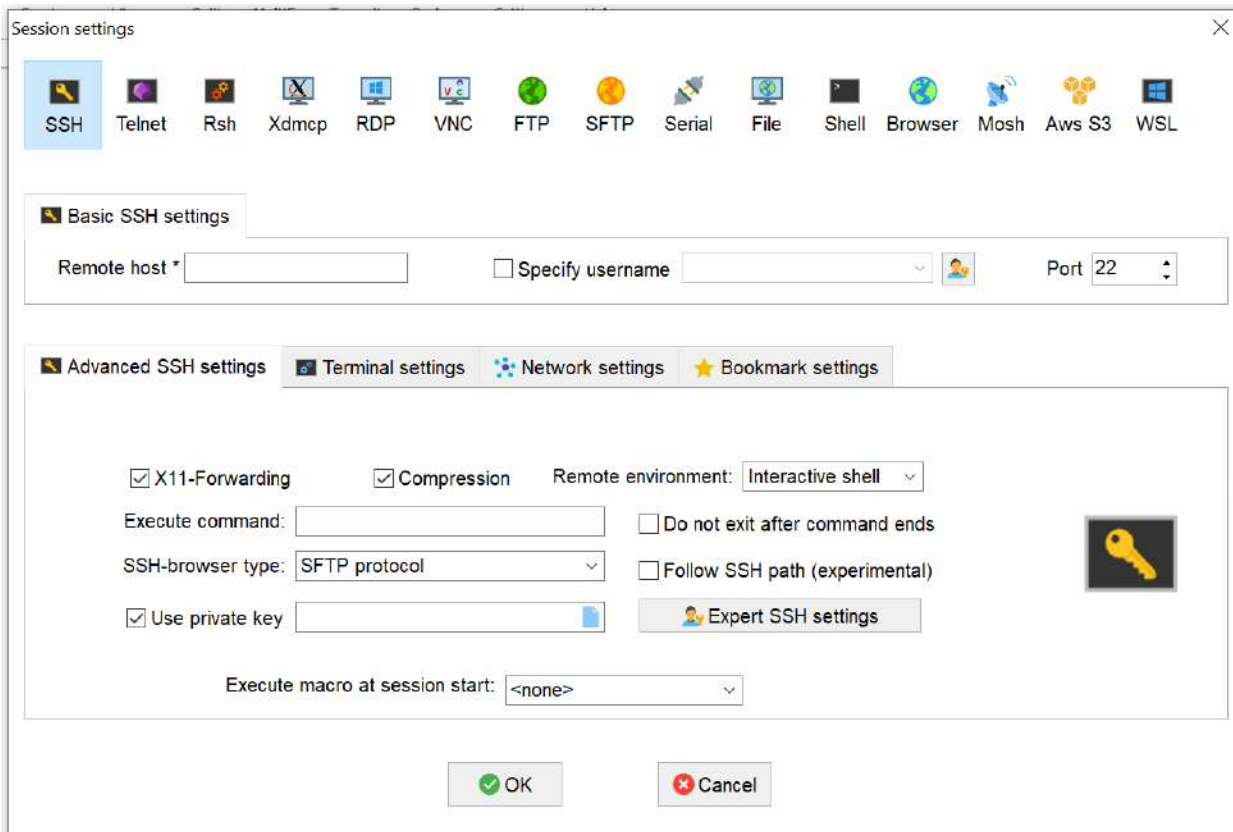
Setup the Kubernetes Cluster

1. Access the instances using MobaXterm application.

1. Create a new session.
2. Get the public IP address for each instance from AWS.



3. Copy the public IP address for each instance to the Remote host.
4. Check the Specify username box and enter “ubuntu” as the username.
5. In the Advanced SSH settings, check the Use private key box and place the .pem file.
6. Duplicate the session to create the 2 worker nodes and the Monitoring sessions as well by replacing the Remote host with each IP address.



2. Setup the Master and Worker Nodes

1. Run the below command to change to root [On Master & Worker Node]

- `sudo su`

2. Create an executable file and place the following commands then run the script [On Master & Worker Node]

Update System Packages

- `sudo apt-get update`

Install Docker

- `sudo apt install docker.io -y`

- `sudo chmod 666 /var/run/docker.sock`

Install Required Dependencies for Kubernetes

- `sudo apt-get install -y apt-transport-https ca-certificates curl gnupg`
- `sudo mkdir -p -m 755 /etc/apt/keyrings`

Add Kubernetes Repository and GPG Key

- `Curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg -- dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg`
- `echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list`

Update Package List

- `sudo apt update`

Install Kubernetes Components

- `sudo apt install -y kubeadm=1.28.1-1.1 kubelet=1.28.1-1.1 kubectl=1.28.1-1.1`

3. Run the following commands on the Master node only

Initialize Kubernetes Master Node

- `sudo kubeadm init--pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all`

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.41.124:6443 --token d3l7q0.4mjjuwm6ux2x5vjr \
--discovery-token-ca-cert-hash sha256:1fba7145bb3c4a7f7e0cbeea5122b4bbae9f210226b5b8de4652239c2cb73f5b
```

After running the above command then our vm will acts as master node and it will generate token to connect this with slave node-copy the token and run the command in slave machines 1 & 2

```
root@ip-172-31-41-124:/home/ubuntu# kubectl get nodes
NAME                 STATUS    ROLES    AGE     VERSION
ip-172-31-37-132     Ready    <none>    11s     v1.28.1
ip-172-31-38-48      Ready    <none>    15s     v1.28.1
ip-172-31-41-124     Ready    control-plane 6m18s   v1.28.1
```

Configure Kubernetes Cluster

- `mkdir -p $HOME/.kube`
- `sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config`
- `sudo chown $(id -u):$(id -g) $HOME/.kube/config`

Deploy Networking Solution (Calico)

- `kubectl apply -f https://docs.projectcalico.org/v3.20/manifests/calico.yaml`

Deploy Ingress Controller (NGINX)

- `kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml`

4. We'll Scan Kubernetes Cluster For Any Kind Of Issues Using Cube Audit

Go To The Website & Copy The Linux_amd_64 Link

- <https://github.com/shopify/kubeaudit/releases>

Paste It Using wget Command

Now Untar The File Using tar-xvf File Name

sudo mv kubeaudit /usr/local/bin/->kubeaudit all

```
root@ip-172-31-41-124:/home/ubuntu# wget https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz
--2024-10-12 17:26:37-- https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/103579225/b7d91b50-ef71-4085-9d2a-5db1ff0f7779?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241012%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241012T172638Z&X-Amz-Expires=300&X-Amz-Signature=03d7733cef32ee942b47b30c929fe0dafa6c057257403e98eb8863a5d378516d&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dkubeaudit_0.22.2_linux_amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2024-10-12 17:26:38-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/103579225/b7d91b50-ef71-4085-9d2a-5db1ff0f7779?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241012%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241012T172638Z&X-Amz-Expires=300&X-Amz-Signature=03d7733cef32ee942b47b30c929fe0dafa6c057257403e98eb8863a5d378516d&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dkubeaudit_0.22.2_linux_amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10181094 (9.7M) [application/octet-stream]
Saving to: 'kubeaudit_0.22.2_linux_amd64.tar.gz'

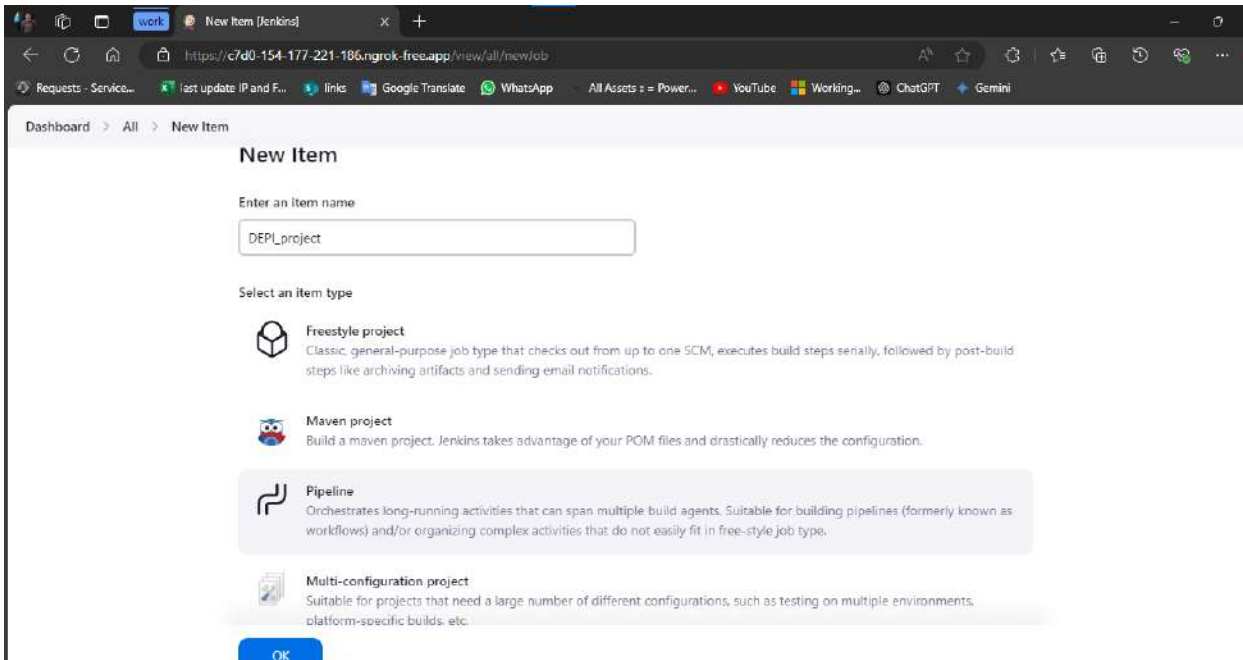
kubeaudit_0.22.2_linux_amd64.ta 100%[=====] 9.71M 18.0MB/s in 0.5s

2024-10-12 17:26:38 (18.0 MB/s) - 'kubeaudit_0.22.2_linux_amd64.tar.gz' saved [10181094/10181094]

root@ip-172-31-41-124:/home/ubuntu# tar -xvf kubeaudit_0.22.2_linux_amd64.tar.gz
README.md
kubeaudit
root@ip-172-31-41-124:/home/ubuntu# sudo mv kubeaudit /usr/local/bin/
root@ip-172-31-41-124:/home/ubuntu# kubeaudit all
```

CI / CD Pipeline

Open Jenkins then Start new project



Jenkins Plugin Installation Steps

1. Eclipse Temurin Installer

- Navigate to **Jenkins Dashboard** -> **Manage Jenkins** -> **Manage Plugins**.
- Go to the **Available** tab, search for **Eclipse Temurin Installer**.
- Select it and click **Install without restart**.

2. Pipeline Maven Integration

- Repeat the above steps, searching for **Pipeline Maven Integration**.
- Select and click **Install without restart**.

3. Config File Provider

- Follow the same procedure, searching for **Config File Provider**.
- Select and click **Install without restart**.

4. SonarQube Scanner

- Search for **SonarQube Scanner** in the **Available** tab.
- Select and click **Install without restart**.

5. Kubernetes CLI

- Search for **Kubernetes CLI** and select it.

- Click **Install without restart**.

6. Kubernetes

- Find **Kubernetes** in the **Available** tab.
- Select and click **Install without restart**.

7. Docker

- Search for **Docker** and select it.
- Click **Install without restart**.

8. Docker Pipeline Step

- Search for **Docker Pipeline Step**.
- Select and click **Install without restart**.

9. Prometheus metrics

- Search for **Prometheus metrics** and select it.
- Select and click **Install with restart**.

10. Slack

- Finally, search for Slack Notification
- Select and click **Install with restart**.

Configure Above Plugins in Jenkins Pipeline

Configure the tools choose manage jenkins → Tools

1. Choose jdk and fill as given below

JDK installations

Add JDK

≡ JDK

Name

jdk17

☒ Install automatically ?

≡ Install from adoptium.net ?

Version ?

jdk-17.0.9+9

Add Installer ▾

2. choose sonarqube scanner and configure

SonarQube Scanner installations

Add SonarQube Scanner

☰ SonarQube Scanner

Name

sonar-scanner

☒ Install automatically ?

☰ Install from Maven Central

Version

SonarQube Scanner 6.10.4477

Add Installer ▾

3. choose maven and Configure

Maven installations

Add Maven

☰ Maven

Name

maven3

☒ Install automatically ?

☰ Install from Apache

Version

3.6.1

Add Installer ▾

4. choose Docker and Configure

Docker installations

Add Docker

≡ Docker

Name

docker

☒ Install automatically ?

≡ Download from docker.com

Docker version ?

latest

Add Installer ▾

1. choose Slack and Configure
 - a. follow steps from <https://plugins.jenkins.io/slack/>

Create Credentials

Jenkins » Credentials [Jenkins] +

localhost:8080/manage/credentials/

80%

Jenkins

Search (CTRL+K)

admin

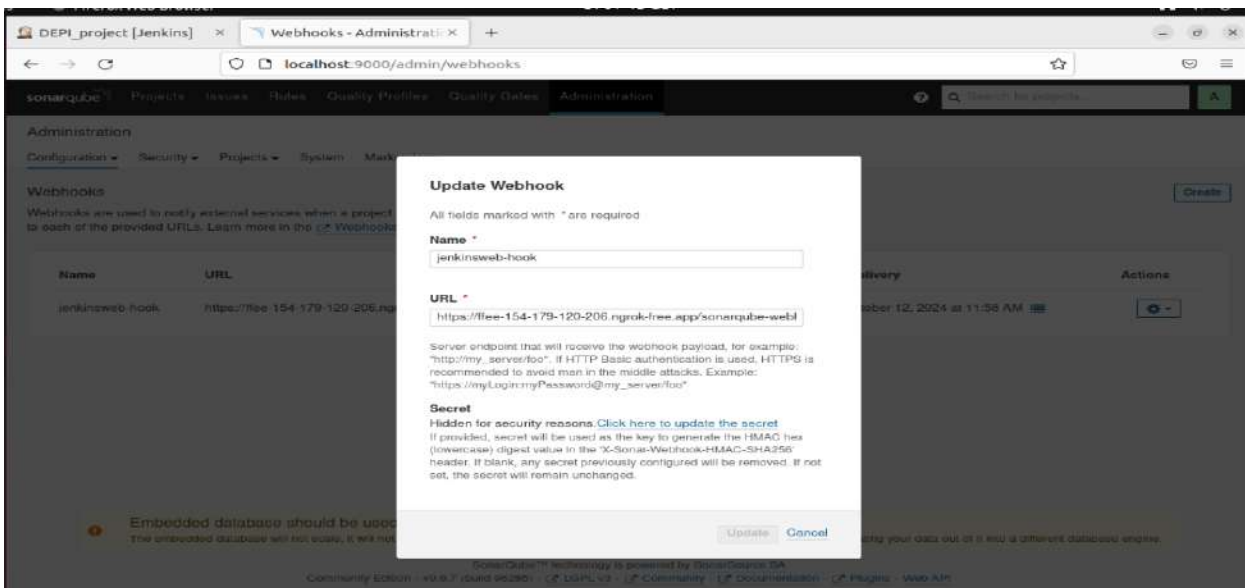
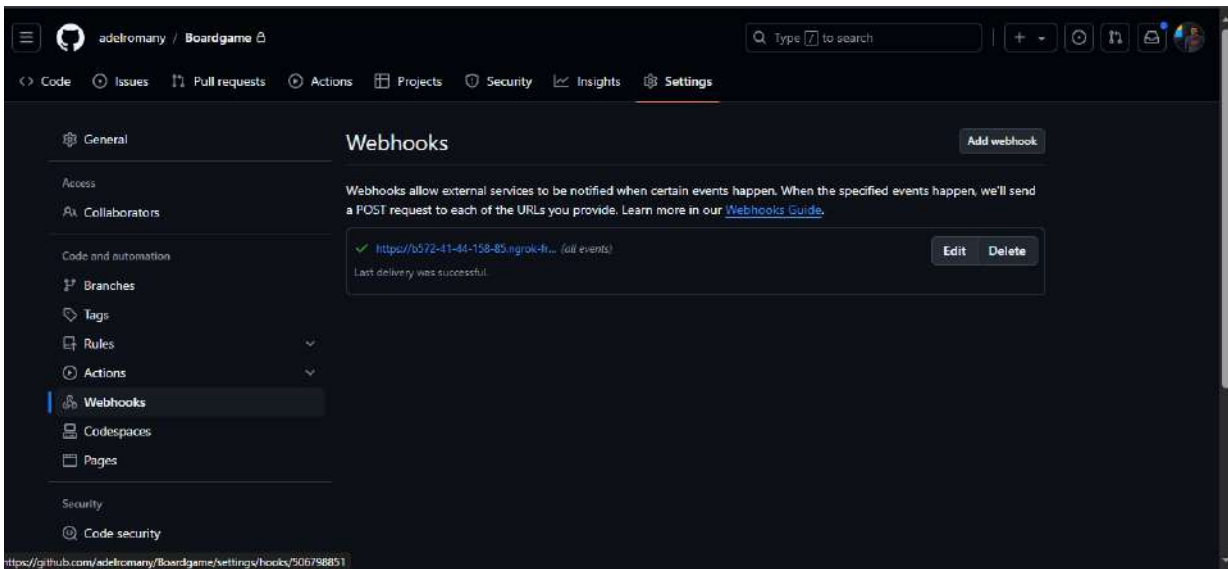
Dashboard » Manage Jenkins » Credentials

Credentials

T	P	Store	Domain	ID	Name
		System	(global)	sonar-token	sonar-token
		System	(global)	git-cred	adetriomany/***** (git-cred)
		System	(global)	docker-cred	adetriomany/***** (docker-cred)
		System	(global)	k8s-cred	k8s-cred
		System	(global)	slack-card	slack-card
		System	(global)	project_git_token	MohamedHalim2812/***** (project_git_token)
		System	(global)	project_k8s_token	project_k8s_token
		System	(global)	project_docker_token	mohamedhalim2812/***** (project_docker_token)

Stores scoped to Jenkins

configure webhooks in GitHub and SonarQube



Configure nexus on the pom.xml

```
<distributionManagement>
```

```
<repository>
```

```
<id>maven-releases</id>
```

```
<url>http://localhost:8081/repository/maven-releases/</url>
```

```
</repository>
```

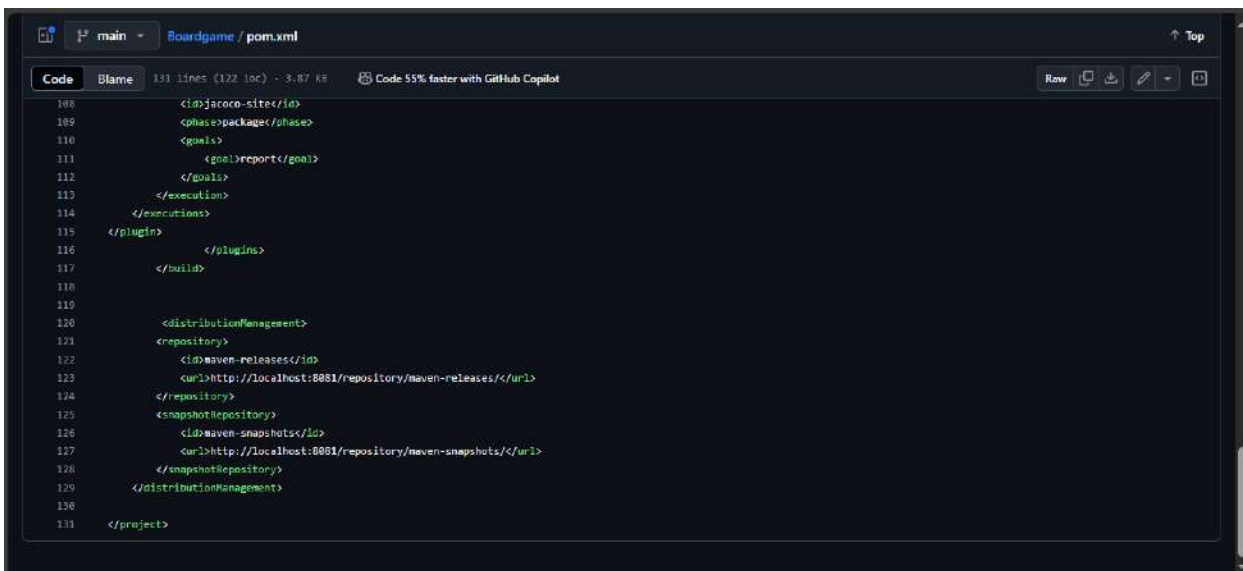
```
<snapshotRepository>
```

```
<id>maven-snapshots</id>
```

```
<url>http://localhost:8081/repository/maven-snapshots/</url>
```

```
</snapshotRepository>
```

```
</distributionManagement>
```



```
108     <id>jacoco-site</id>
109     <phase>package</phase>
110     <goals>
111       <goal>report</goal>
112     </goals>
113   </execution>
114 </executions>
115 </plugin>
116 </plugins>
117 </build>
118
119
120   <distributionManagement>
121     <repository>
122       <id>maven-releases</id>
123       <url>http://localhost:8081/repository/maven-releases</url>
124     </repository>
125     <snapshotRepository>
126       <id>maven-snapshots</id>
127       <url>http://localhost:8081/repository/maven-snapshots</url>
128     </snapshotRepository>
129   </distributionManagement>
130
131 </project>
```

The pipeline:

```
pipeline {
```

```
  agent any
```

```
  tools {
```

```
    jdk 'jdk17'
```

```
    maven 'maven3'
```

```
  }
```

```
  environment {
```

```
    SCANNER_HOME = tool 'sonar-scanner'
```

```
    DOCKER_BUILDKIT = "1"
```

```
  }
```

```
  stages {
```

```
    stage('Git Checkout') {
```

```
      steps {
```

```
        git branch: 'main', credentialsId: 'project_git_token', url:
'https://github.com/MohamedHalim2812/UlimateCICD.git' //change with your git repository
```

```
      }
```

```
    }
```

```
stage('Compile') {  
    steps {  
        sh "mvn compile"  
    }  
}
```

```
stage('Test') {  
    steps {  
        sh "mvn test"  
    }  
}
```

```
stage('File System Scan') {  
    steps {  
        sh "trivy fs --format table -o trivy-fs-report.html ."  
    }  
}
```

```
stage('SonarQube Analysis') {  
    steps {  
        withSonarQubeEnv('sonar') {  
            sh "' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=BoardGame -  
Dsonar.projectKey=BoardGame \  
-Dsonar.java.binaries=. "'  
        }  
    }  
}  
  
stage('Quality Gate') {  
    steps {  
        script {  
            waitForQualityGate abortPipeline: false, credentialsId: 'sonar-token'  
        }  
    }  
}
```

```

    }
}

stage('Build') {
    steps {
        sh "mvn package"
    }
}

stage('Publish To Nexus') {
    steps {
        withMaven(globalMavenSettingsConfig: 'global-settings', jdk: 'jdk17', maven: 'maven3',
mavenSettingsConfig: "", traceability: true) {
            sh "mvn deploy"
        }
    }
}

stage('Build & Tag Docker Image') {
    steps {
        script {
            withDockerRegistry(credentialsId: 'project_docker_token', toolName: 'docker') {
                sh "docker build -t mohamedhalim2812/boardgame:latest ." //change to your docker account
            }
        }
    }
}

stage('Docker Image Scan') {
    steps {
        sh "trivy image --format table -o trivy-image-report.html mohamedhalim2812/boardgame:latest"
//change to your docker account
    }
}

```

```
}  
}
```

```
stage('Push Docker Image') {  
  steps {  
    script {  
      withDockerRegistry(credentialsId: 'project_docker_token', toolName: 'docker') {  
        sh "docker push mohamedhalim2812/boardgame:latest" //change to your docker account  
      }  
    }  
  }  
}
```

```
stage('Deploy To Kubernetes') {  
  steps {  
    withKubeConfig(caCertificate: "", clusterName: 'kubernetes', contextName: "", credentialsId:  
'project_k8s_token', namespace: 'ultimate-cicd', restrictKubeConfigAccess: false, serverUrl:  
'https://54.209.235.133') {  
      sh "kubectl apply -f deployment-service.yaml" //change with your credentialsId  
    }  
  }  
}
```

```
stage('Verify the Deployment') {  
  steps {  
    withKubeConfig(caCertificate: "", clusterName: 'kubernetes', contextName: "", credentialsId:  
'project_k8s_token', namespace: 'ultimate-cicd', restrictKubeConfigAccess: false, serverUrl:  
'https://54.209.235.133') {  
      sh "kubectl get pods -n ultimate-cicd"  
      sh "kubectl get svc -n ultimate-cicd"  
    }  
  }  
}
```


}

}

Build log [#43] [Jenkins] x BoardGame x Welcome - Sonatype Nexu x +

localhost:8080/job/UltimateCICDs/43/pipeline-console/ 80%

Jenkins

Dashboard > UltimateCICDs > #43 > Pipeline Console

[Rebuild](#) [Overview](#) [Configure](#) [...](#)

< Build #43

In progress 3 min 43 sec ago in 3 min 43 sec and counting

- Tool Install
- Git Checkout
- Compile
- Test
- File System Scan
- SonarQube Analysis
- Quality Gate
- Build
- Publish To Nexus
- Build & Tag Docker Image
- Docker Image Scan
- Push Docker Image
- Deploy To Kubernetes
- Verify the Deployment

Started 4.9 sec ago
Queued 0 ms
Took 4.1 sec
Success
[View as plain text](#)

jdk17
Use a tool from a predefined Tool Installation 0.18 sec [🔗](#) [🔄](#) [⌵](#)

Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. 0.23 sec [🔗](#) [🔄](#) [⌵](#)

maven3
Use a tool from a predefined Tool Installation 0.14 sec [🔗](#) [🔄](#) [⌵](#)

Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. 96 ms [🔗](#) [🔄](#) [⌵](#)

kubectl get pods -n ultimate-cicd
Shell Script 1.2 sec [🔗](#) [🔄](#) [⌵](#)

kubectl get svc -n ultimate-cicd
Shell Script 1.6 sec [🔗](#) [🔄](#) [⌵](#)

	NAME	TYPE	CLUSTER IP	EXTERNAL IP	PORT(S)	AGE
1	boardgame-svc	LoadBalancer	10.108.251.237	<pending>	80:30289/TCP	94s

Jenkins 2.462.3

Firefox Web Browser 16:11 13 Oct

Build log [#43] [Jenkins] x BoardGame x Browse - Sonatype Nexu x +

localhost:8081/#browse/browse:maven-releases 80%

Sonatype Nexus Repository OSS 3.72.0-04

[Browse](#) [Welcome](#) [Search](#) [Browse](#)

Browse / maven-releases

HTML View [Advanced](#)

- com
- javaproject
 - database_service_project
 - 0.0.4
 - database_service_project-0.0.4.jar
 - database_service_project-0.0.4.jar.md5
 - database_service_project-0.0.4.jar.sha1
 - database_service_project-0.0.4.pom
 - database_service_project-0.0.4.pom.md5
 - database_service_project-0.0.4.pom.sha1
 - maven-metadata.xml
 - maven-metadata.xml.md5
 - maven-metadata.xml.sha1

ultimateCICDs #43 [Jenki...]

BoardGame

New Tab

localhost:9000/dashboard?id=BoardGame

80%

sonarqube

ProjectsIssuesRulesQuality ProfileQuality GateAdministration

BoardGamemain

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

QUALITY GATE STATUS

Failed

1 condition failed

31.12% Duplicated Lines (%) on New Code is greater than 2.0%

15 Bugs

0 Vulnerabilities

1 Security Hotspots

0.0% Reviewed

3h 28min Debt

35 Code Smells

30.8% Coverage on 244 Lines to cover

6 Unit Tests

10.0% Duplications on 1.8k Lines

12 Duplicated Blocks

Reliability

Security

Security Review

Maintainability

ACTIVITY

to this VM, click inside or press Ctrl+G

Filter...

#43

(pending—Finished waiting)

Firefox Web Browser

16:06 13 أكت

Jenkins

Webhooks - AdministrationNew Tab

localhost:8080/job/UltimateCICDs/GitHubPollLog/

Search (CTRL+K)

adminlog out

Dashboard > UltimateCICDs > GitHub Hook Log

Status

Changes

Build Now

Configure

Delete Pipeline

GitHub

SonarQube

Stages

Rename

Pipeline Syntax

GitHub Hook Log

Build History

trend

Last GitHub Push

Started on Oct 13, 2024, 4:05:36 PM

Started by event from 149.82.115.41 - <https://b572-41-44-158-85.ngrok-free.app:8080/github-webhook/> on Sun Oct 13 10:05:30 EEST 2024

Using strategy: Default

[poll] Last Built Revision: Revision 161a7041a0edf51a21f1dc9cc10d4668d17d965 (refs/remotes/origin/main)

The recommended git tool is: NONE

using credential git-cred

> git --version # timeout=10

> git --version # 'git version 2.34.1'

using GIT_ASKPASS to set credentials git-cred

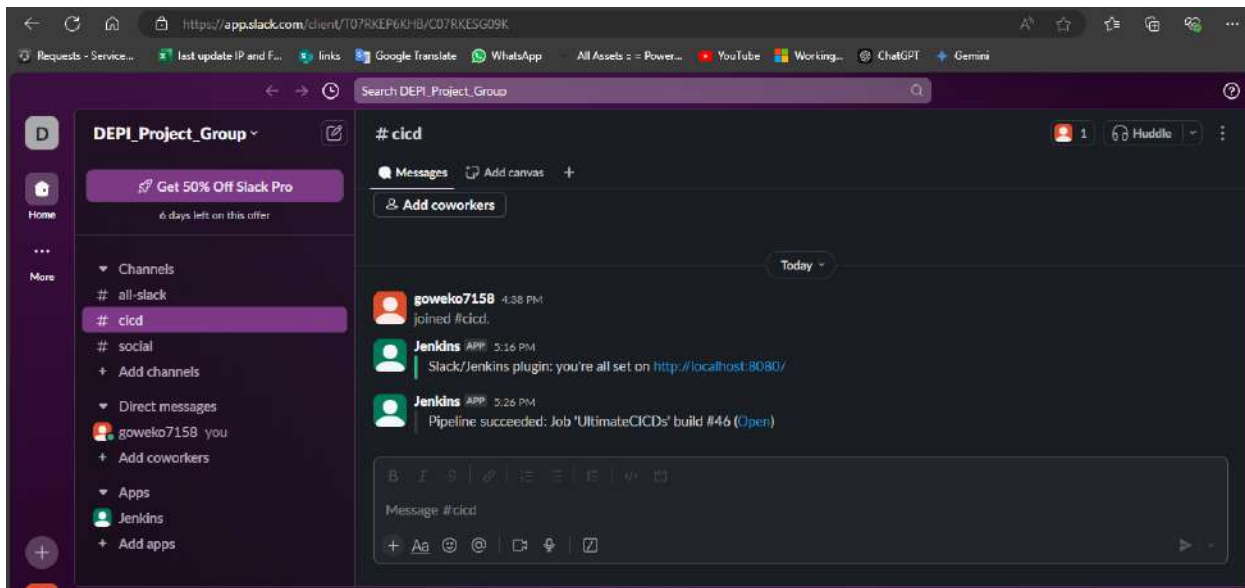
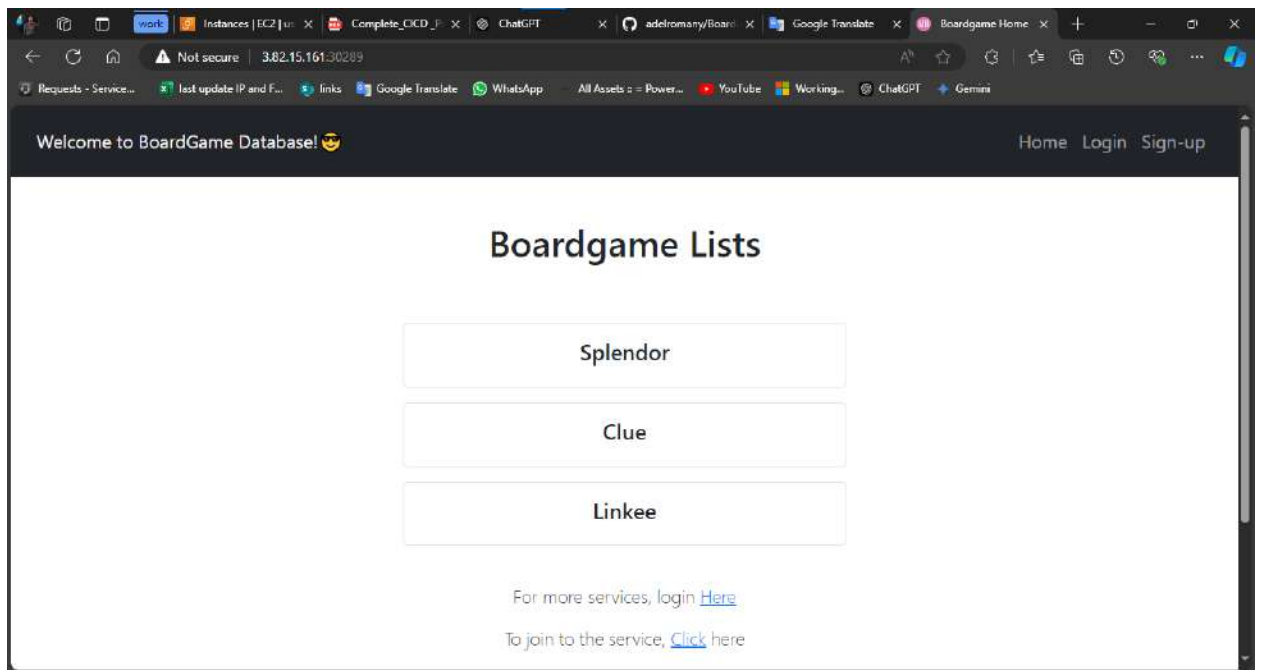
> git ls-remote -h -- <https://github.com/adelromany/Boardgame> # timeout=10

Found 1 remote heads on <https://github.com/adelromany/Boardgame>

[poll] Latest remote head revision on refs/heads/main is: 09c53eacc7ffbef35821ab05df6e080adc42bcf9

Done. Took 1.1 sec

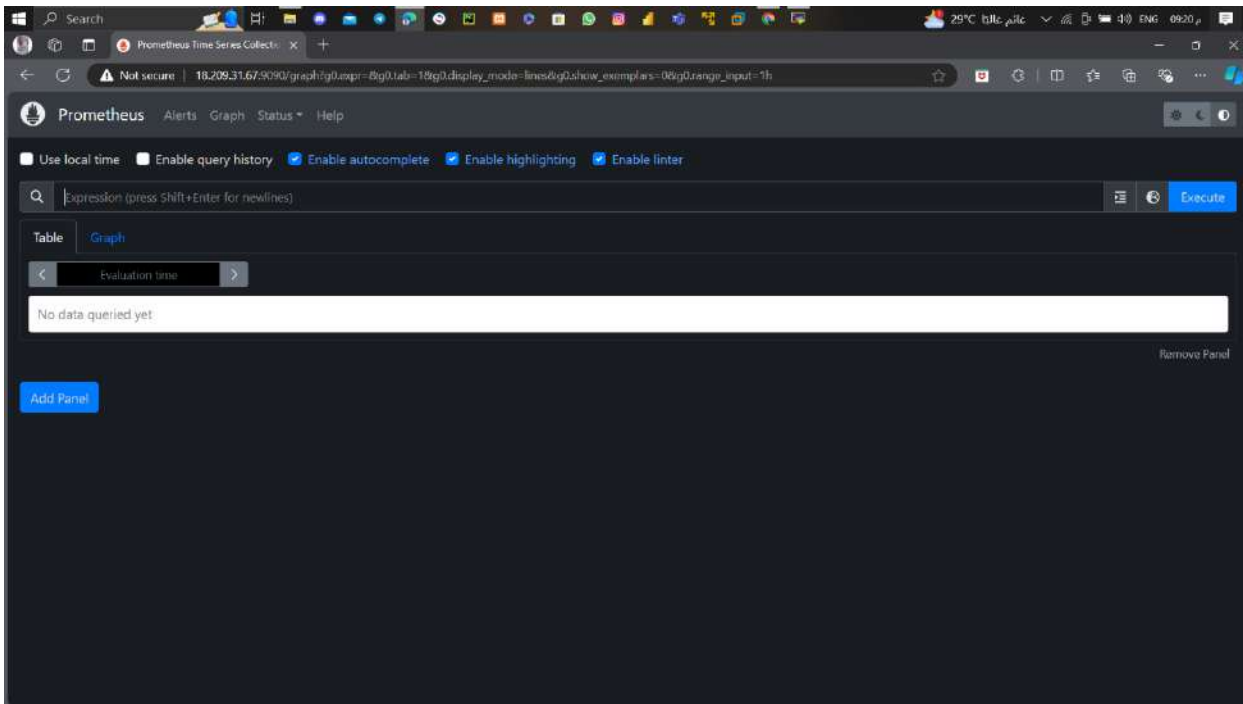
Changes found



Monitoring

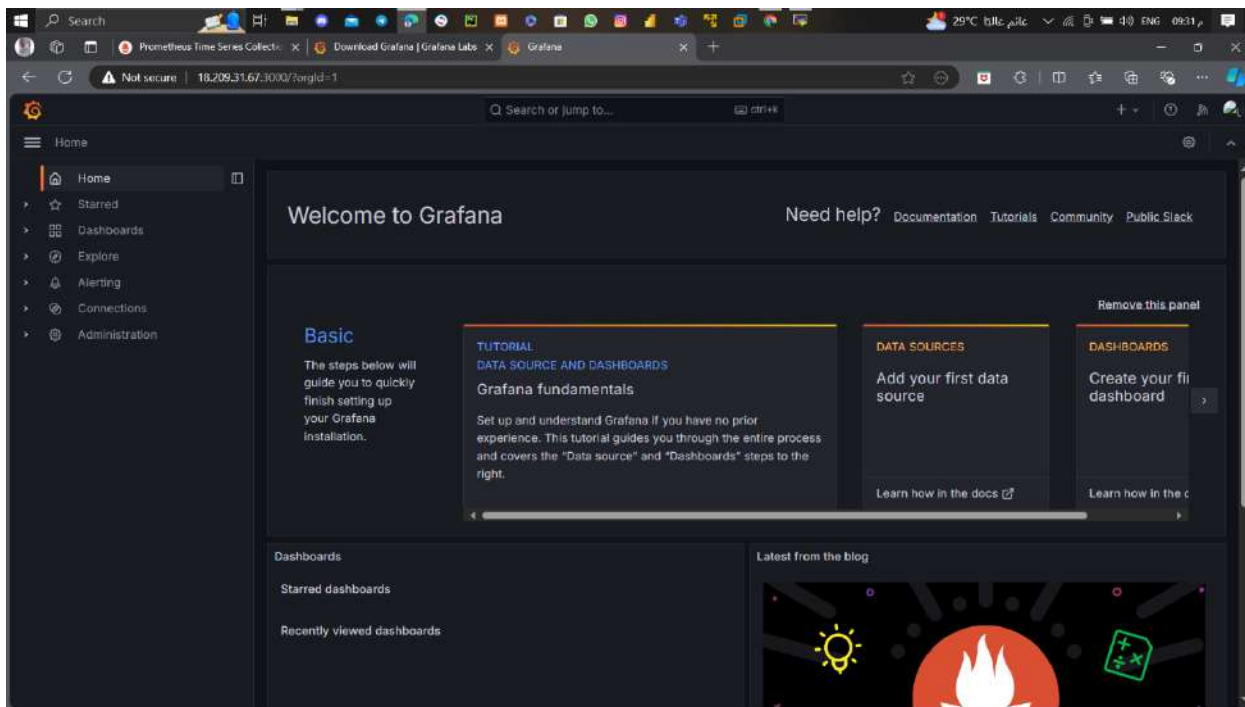
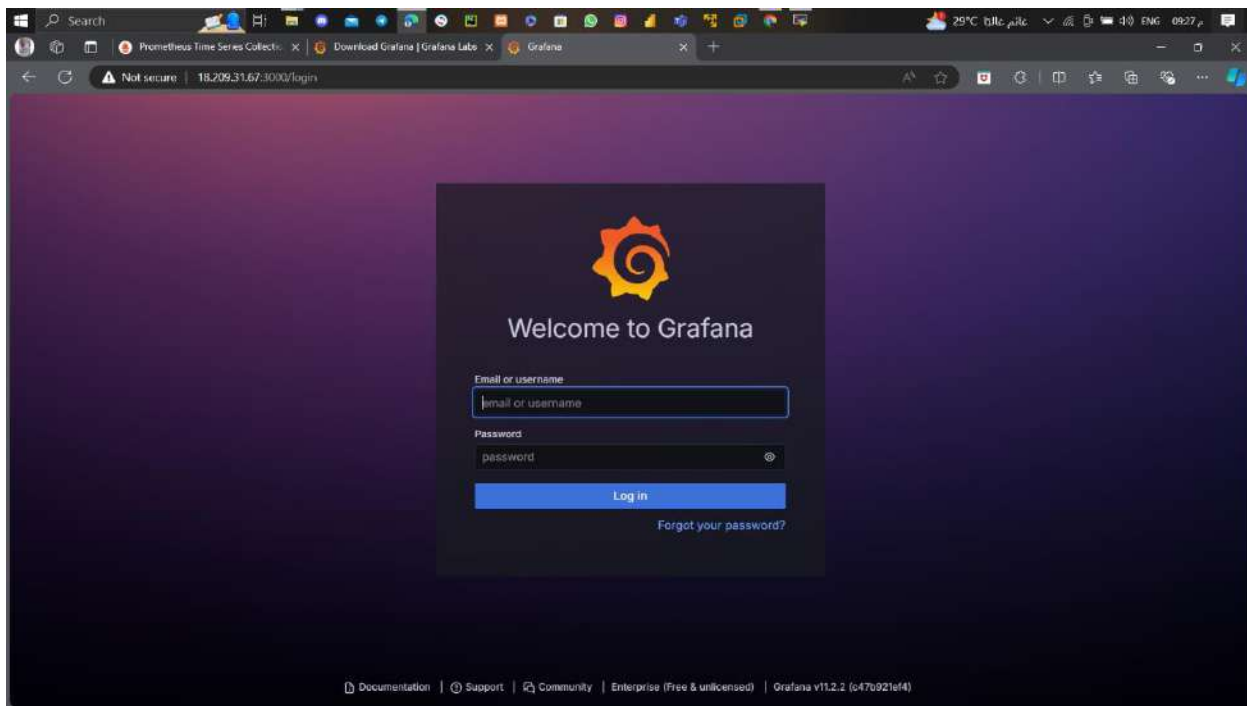
##Installing Prometheus

- sudo apt update
- wget <https://github.com/prometheus/prometheus/releases/download/v2.53.2/prometheus-2.53.2.linux-amd64.tar.gz>
- tar -xvf prometheus-2.53.2.linux-amd64.tar.gz
- cd prometheus-2.53.2.linux-amd64/
- ./prometheus & ## to start Prometheus
- 18.209.31.67:9090 ## to access prometheus



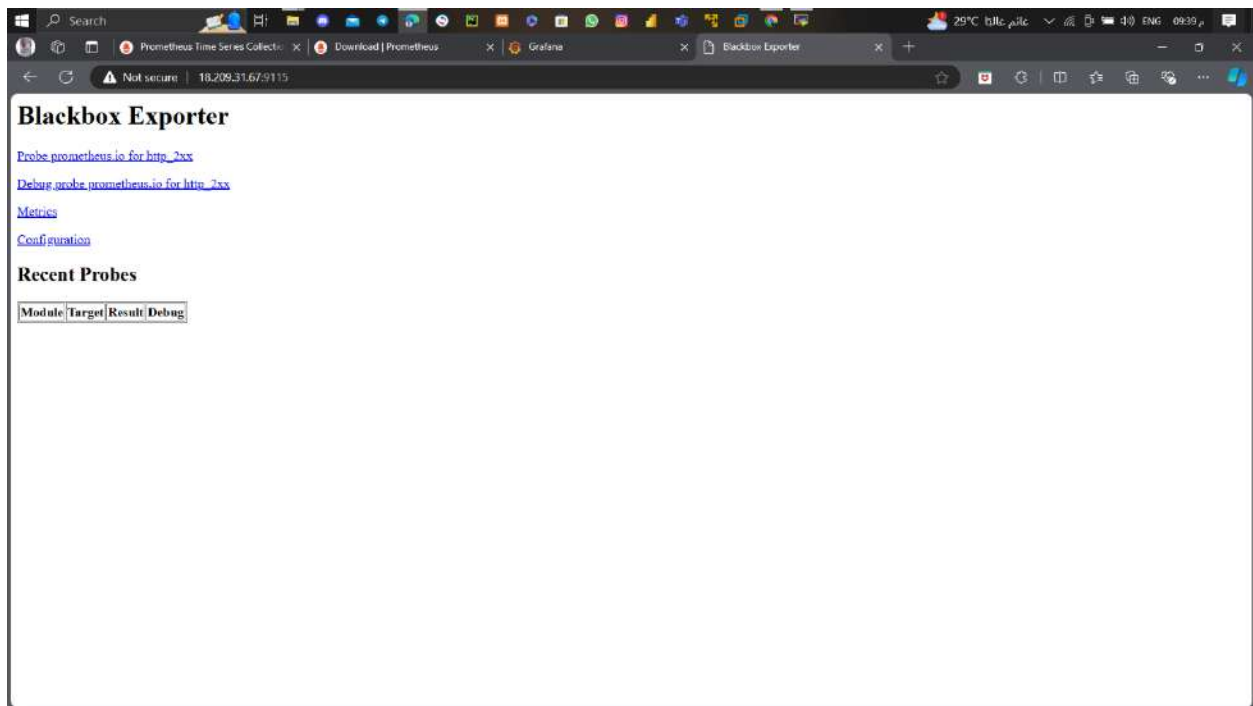
##Installing Grafana

- ```
- sudo apt-get install -y adduser libfontconfig1 musl
- wget https://dl.grafana.com/enterprise/release/grafana-enterprise_11.2.2_amd64.deb
- sudo dpkg -i grafana-enterprise_11.2.2_amd64.deb
- sudo /bin/systemctl start grafana-server ## to start Grafana
- 18.209.31.67:3000 ## default username and password : admin
```

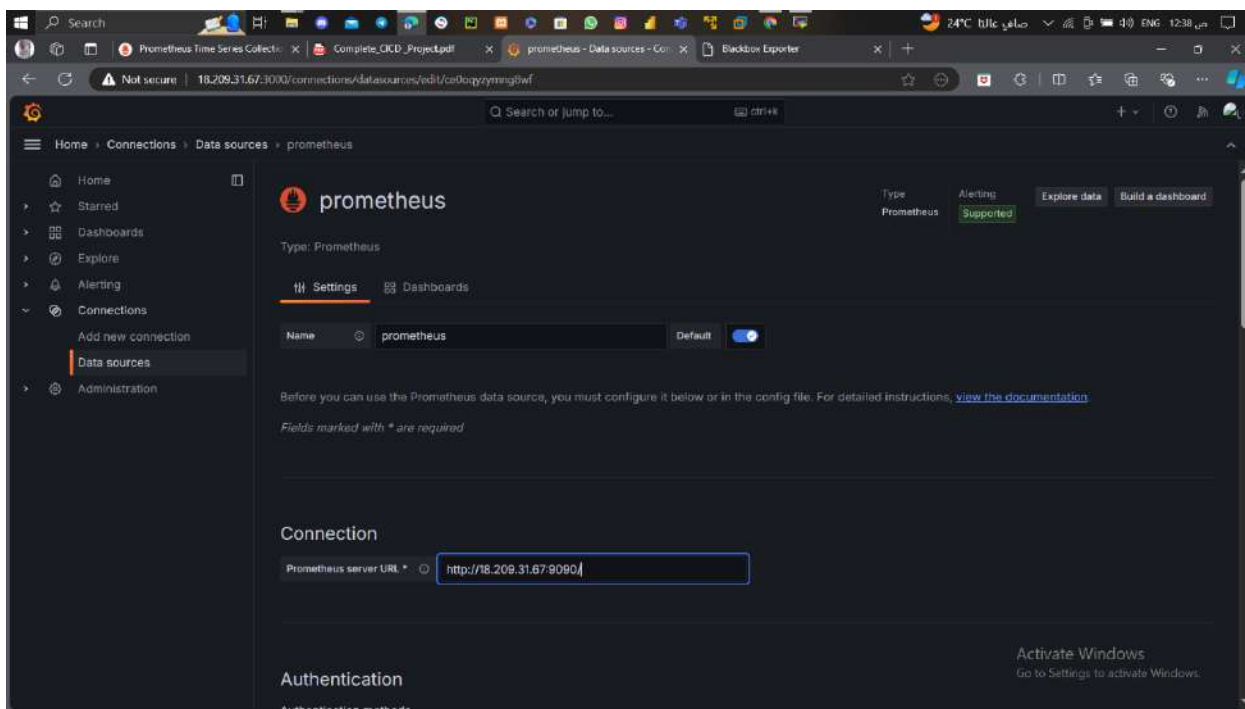


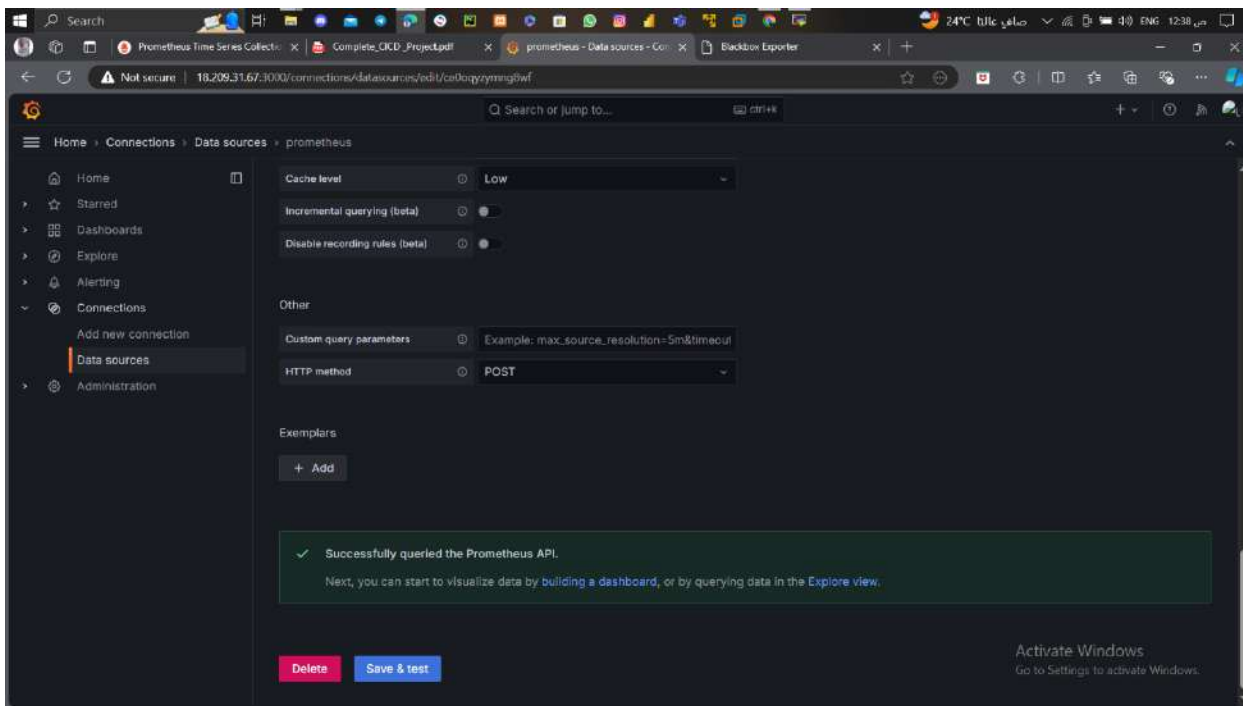
## ## Installing blackbox\_exporter

- Wget [https://github.com/prometheus/blackbox\\_exporter/releases/download/v0.25.0/blackbox\\_exporter-0.25.0.linux-amd64.tar.gz](https://github.com/prometheus/blackbox_exporter/releases/download/v0.25.0/blackbox_exporter-0.25.0.linux-amd64.tar.gz)
- tar -xvf blackbox\_exporter-0.25.0.linux-amd64.tar.gz
- cd blackbox\_exporter-0.25.0.linux-amd64/
- ./blackbox\_exporter &    ##to start blackbox
- 18.209.31.67:9115



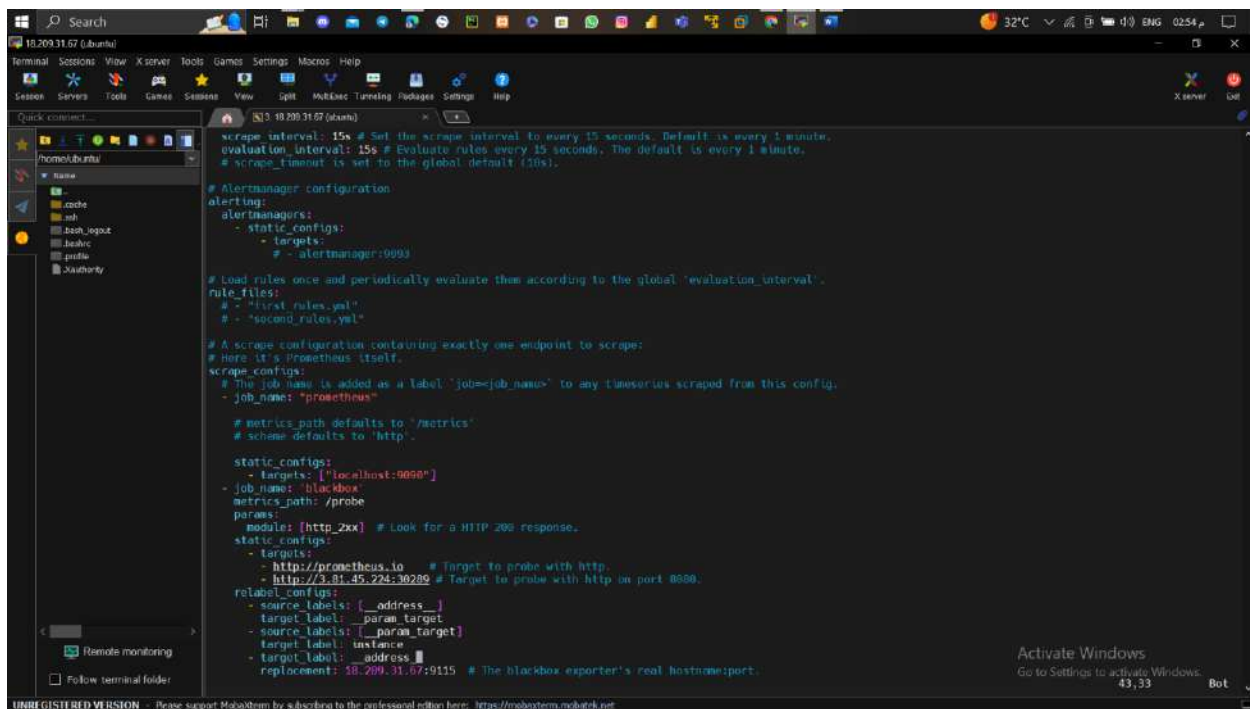
## ##Adding Prometheus as a Data Source to Grafana





## ## Monitoring the website

- Prometheus yaml configuration



## ## Restart Prometheus

- pgrep prometheus
- kill ##the id
- ./prometheus &



```
ubuntu@ip-172-31-47-206:~/prometheus-2.53.2.linux-amd64$ vi prometheus.yml
ubuntu@ip-172-31-47-206:~/prometheus-2.53.2.linux-amd64$ pgrep prometheus
1056
ubuntu@ip-172-31-47-206:~/prometheus-2.53.2.linux-amd64$ kill 1056
ubuntu@ip-172-31-47-206:~/prometheus-2.53.2.linux-amd64$./prometheus &
[1] 6649
ubuntu@ip-172-31-47-206:~/prometheus-2.53.2.linux-amd64$ ts=2024-10-13T11:56:31.264Z caller=main.go:509 level=info msg="No time or size retention was set so u
sing the default time retention" duration=15d
ts=2024-10-13T11:56:31.264Z caller=main.go:633 level=info msg="Starting Prometheus Server" mode=server version="(version=2.53.2, branch=HEAD, revision=a6e97a7
d-995096d4bc4ffa50b6f282fcfac5fa9)"
ts=2024-10-13T11:56:31.264Z caller=main.go:638 level=info build_context="(go=go1.22.6, platform=linux/amd64, user=root@363b0aa99939, date=20240809-14:55:04, t
ags=netgo,builtinassets,stringlabels)"
ts=2024-10-13T11:56:31.264Z caller=main.go:639 level=info host_details="(Linux 6.0.0-1016-aws #17-Ubuntu SMP Mon Sep 2 13:48:07 UTC 2024 x86_64 ip-172-31-47-
206 (name))"
ts=2024-10-13T11:56:31.264Z caller=main.go:640 level=info fd_limits="(soft=1048576, hard=1048576)"
ts=2024-10-13T11:56:31.264Z caller=main.go:641 level=info vm_limits="(soft=unlimited, hard=unlimited)"
ts=2024-10-13T11:56:31.271Z caller=web.go:568 level=info component=web msg="Start listening for connections" address=0.0.0.0:9090
ts=2024-10-13T11:56:31.271Z caller=main.go:1148 level=info msg="Starting TSDB ..."
ts=2024-10-13T11:56:31.273Z caller=repair.go:56 level=info component=tsdb msg="Found healthy block" mint=1728795202205 maxt=1728805400000 ulid=01JA2H9DS12A0Q
P647PQ0PQ50H
ts=2024-10-13T11:56:31.274Z caller=repair.go:56 level=info component=tsdb msg="Found healthy block" mint=1728805402205 maxt=1728813600000 ulid=01JA2H9DS12A0Q
P647PQ0PQ50H
ts=2024-10-13T11:56:31.274Z caller=repair.go:56 level=info component=tsdb msg="Found healthy block" mint=1728757082205 maxt=1728795200000 ulid=01JA2H9DS12A0Q
P647PQ0PQ50H
ts=2024-10-13T11:56:31.282Z caller=head.go:626 level=info component=tsdb msg="Replaying on-disk memory mappable chunks if any"
ts=2024-10-13T11:56:31.282Z caller=head.go:713 level=info component=tsdb msg="On-disk memory mappable chunks replay completed" duration=560.39us
ts=2024-10-13T11:56:31.282Z caller=head.go:721 level=info component=tsdb msg="Replaying WAL, this may take a while"
ts=2024-10-13T11:56:31.286Z caller=tsdb/config.go:313 level=info component=web msg="Listening on" address=:9090
ts=2024-10-13T11:56:31.286Z caller=tsdb/config.go:316 level=info component=web msg="TLS is disabled." http2=false address=:9090
ts=2024-10-13T11:56:31.287Z caller=head.go:758 level=info component=tsdb msg="WAL checkpoint loaded"
ts=2024-10-13T11:56:31.297Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=6 maxSegment=9
ts=2024-10-13T11:56:31.328Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=7 maxSegment=9
ts=2024-10-13T11:56:31.352Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=8 maxSegment=9
ts=2024-10-13T11:56:31.352Z caller=head.go:793 level=info component=tsdb msg="WAL segment loaded" segment=9 maxSegment=9
ts=2024-10-13T11:56:31.352Z caller=head.go:639 level=info component=tsdb msg="WAL replay completed" checkpoint_replay_duration=4.636849ms wal_replay_duration=
67.853647ms wal_replay_duration=623ms chunk_snapshot_load_duration=0s mmap_chunk_replay_duration=560.39us total_replay_duration=73.30486ms
ts=2024-10-13T11:56:31.357Z caller=main.go:1169 level=info fs_type=EXT4_SUPER_MAGIC
ts=2024-10-13T11:56:31.357Z caller=main.go:1172 level=info msg="TSDB started"
ts=2024-10-13T11:56:31.357Z caller=main.go:1354 level=info msg="Loading configuration file" filename=prometheus.yml
ts=2024-10-13T11:56:31.363Z caller=main.go:1391 level=info msg="updated GC" old=100 new=75
```

Prometheus Time Series Collector

Targets

All scrape pools All Unhealthy Collapse All Filter by endpoint or labels

blackbox (2/2 up) show logs

| Endpoint                       | State | Labels                                             | Last Scrape | Scrape Duration | Error |
|--------------------------------|-------|----------------------------------------------------|-------------|-----------------|-------|
| http://18.209.31.67:9115/probe | UP    | instance="http://prometheus.io" job="blackbox"     | 3.454s ago  | 50.638ms        |       |
| http://18.209.31.67:9115/probe | UP    | instance="http://3.81.45.224:30209" job="blackbox" | 2.561s ago  | 16.203ms        |       |

prometheus (1/1 up) show logs

| Endpoint                      | State | Labels                                     | Last Scrape | Scrape Duration | Error |
|-------------------------------|-------|--------------------------------------------|-------------|-----------------|-------|
| http://localhost:9090/metrics | UP    | instance="localhost:9090" job="prometheus" | 8.543s ago  | 4.040ms         |       |

Activate Windows  
Go to Settings to activate Windows.

## Import the dashboard



Search

18.209.31.67:3000/dashboard/import

Home Dashboards Import dashboard

## Import dashboard

Import dashboard from file or Grafana.com

### Importing dashboard from Grafana.com

Published by **sparanoid**

Updated on **2018-08-19 18:22:07**

### Options

Name  
Prometheus Blackbox Exporter

Folder  
Dashboards

Unique identifier (UID)  
The unique identifier (UID) of a dashboard can be used to uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

xtbCIBkiz [Change uid](#)

signcd-prometheus  
prometheus

[Import](#) [Cancel](#)

Activate Windows  
Go to Settings to activate Windows.

