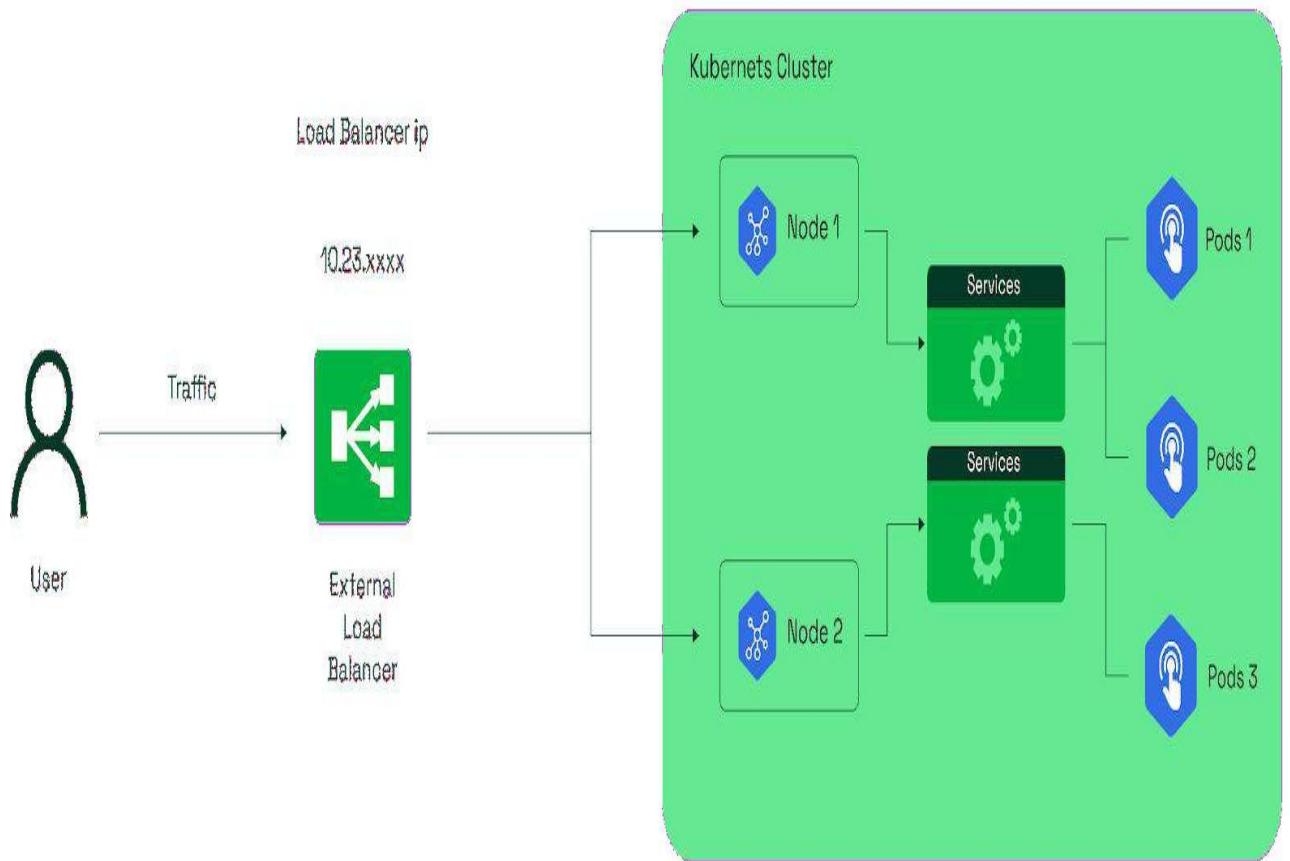


# KUBERNETES PROJECT WITH LOAD BALANCER SERVICE



# **Files and Configurations Used in This Project**

## **1. NGINX Configuration**

- NGINX Deployment with 3 Replicas
- NGINX Load Balancer Service
- NGINX Ingress
- External Name Service

## **2. Backend Configuration**

- backend-app.js file
- Dockerfile
- backend-app-pod

## **3. Secrets and Configurations**

- RDS Secrets
- RDS ConfigMaps
- Service Account for Secrets and ConfigMaps
- Read-Only Roles for Secrets and ConfigMaps
- Read-Only Role-Binding for Secrets and ConfigMaps

## **4. Database Clients**

- MySQL Client
- Oracle Client
- PostgreSQL Client

## **5. Monitoring and Logging**

- DaemonSet (Prometheus)
- Prometheus RBAC
- Prometheus NodePort Exposed on Port 30090
- Promtail DaemonSet
- Loki DaemonSet
- Loki NodePort Exposed on Port 30091
- Grafana Daemon with Exposed Port 30080

## **Project Summary**

### **1. NGINX Deployment and Services**

- Sabse pehle NGINX ke 3 replicas ka creation aur unko Load Balancer service se expose karna.
- Phir, Ingress service ka use karna routing ke liye.

### **2. RDS Implementation**

- Amazon RDS ko external service ke through access karna using ExternalName service. Isme aapko bata sakte hain ki aap kaise MySQL, Oracle, ya PostgreSQL ko RDS se access kar rahe hain.

### **3. HTTPS Implementation with AWS ACM**

- HTTPS ko enable karne ke liye AWS ACM service ka istemal, jo HTTP ko HTTPS me convert karta hai.

### **4. Dockerfile and Custom Application Deployment**

- Dockerfile ka use karke Node.js application (app.js) create karna, jise Amazon RDS se data fetch kar sakein or connect kar sekein.
- Required dependencies install karke Docker image banani aur usko Docker Hub par push karna.

### **5. Secrets, ConfigMaps, and RBAC**

- Security ke liye Secrets (Amazon RDS ka password aur username store karne ke liye), ConfigMaps (ports, hostnames, database names ke liye) ka use.
- Service Account, Role, aur Role Binding create karna jo Secrets aur ConfigMaps ke liye read-only permissions provide karein.

### **6. Client Creation (MySQL, Oracle, PostgreSQL)**

- MySQL-client, Oracle-client, aur PostgreSQL-client create karna jo RDS se data fetch kar sakein or connect kar sakein.

## **7. Monitoring and Logging**

- Prometheus DaemonSet deploy karna, RBAC configure karna aur port 30090 par expose karte hue read-only permissions dena.
- Promtail ka use karna logs collect karne ke liye.
- Loki ko logging backend banakar port 30091 par expose karna.
- Grafana deploy karna jo metrics visualize karega aur port 30080 par expose hogा.

### **Understanding Each File**

**Is project me hum har YAML aur Configuration file ko samjhenge ki kaise ye Deployment, Networking ko manage karne, Security implement karne, aur Monitoring Setup karne me madad kar rahi hai.**

## Achievements

### 1. Successfully Deployed NGINX with Load Balancer Service

NGINX web server ko 3 replicas ke saath deploy kiya aur Load Balancer service ka use karke expose kiya, jo high availability aur efficient traffic distribution ensure karta hai.

### 2. Domain Name Configuration with Ingress

Ingress ka use karke custom domain name set kiya, jo better routing aur easy access provide karta hai.

### 3. Converted HTTP to HTTPS using AWS ACM

AWS Certificate Manager (ACM) ka use karke SSL/TLS certificate implement kiya aur HTTPS enable kiya.

### 4. External Name Service for Database Connectivity

Amazon RDS ko ExternalName service ke through securely aur seamlessly cluster ke andar access kiya.

### 5. Secrets & ConfigMaps Implementation

Sensitive information jaise database credentials, endpoints, aur configuration settings securely manage kiye.

### 6. RBAC-based Access Control

RBAC (Role-Based Access Control) implement kiya jo Secrets aur ConfigMaps ke liye read-only access provide karta hai.

### 7. Monitoring & Logging Setup

Prometheus, Loki, aur Grafana ka use karke complete observability stack setup kiya jo system ki health aur performance track karne me madad karta hai.

**Note: Sabse pehle EKS Cluster ki installation karni hai, jaise hum pehle Installation part mein kar chuke hain. Agar tumne EKS setup nahi kiya hai, toh neeche diye gaye URL ko Click aur EKS Cluster ka setup complete karo.**

**URL:**

<https://github.com/Faikhan147/Kubernetes/blob/main/03-Installations/03-EKS-Cluster-Creation.pdf>

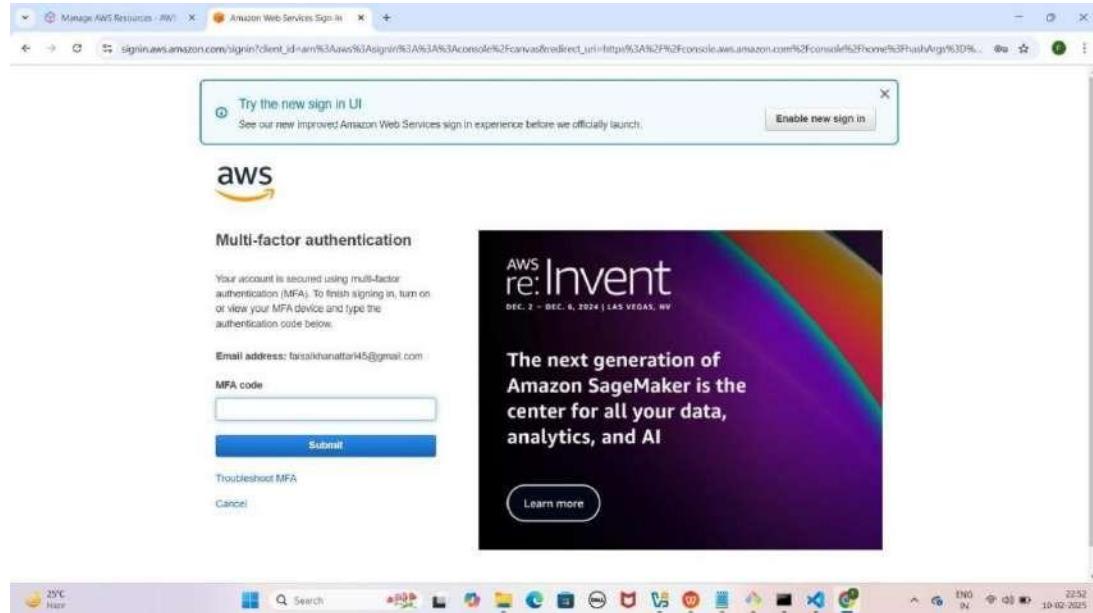
PROJECT -->(Stockholm Region Select Karo)

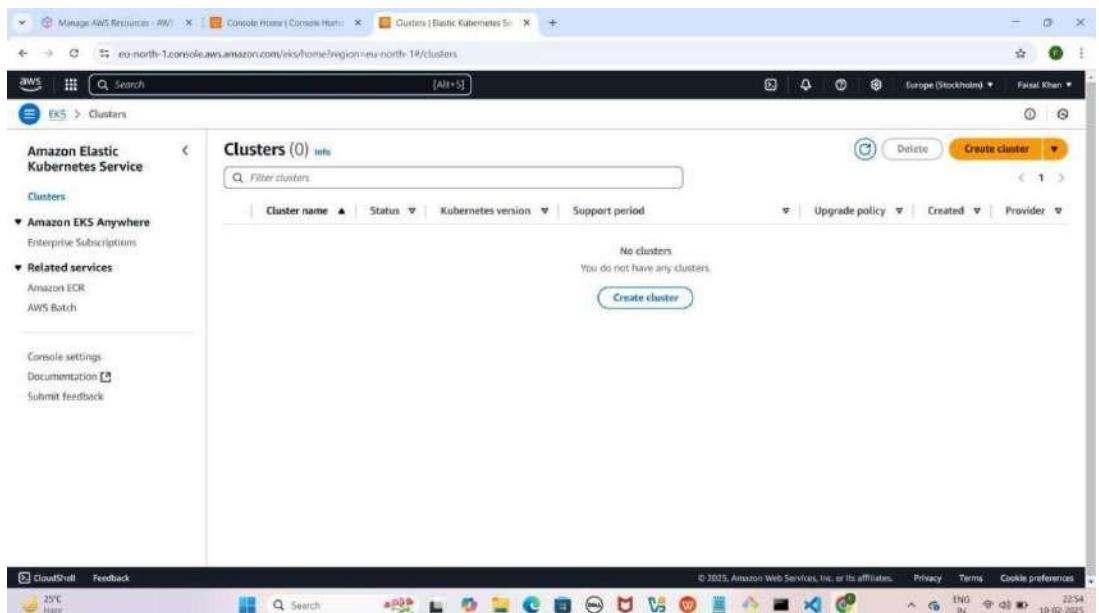
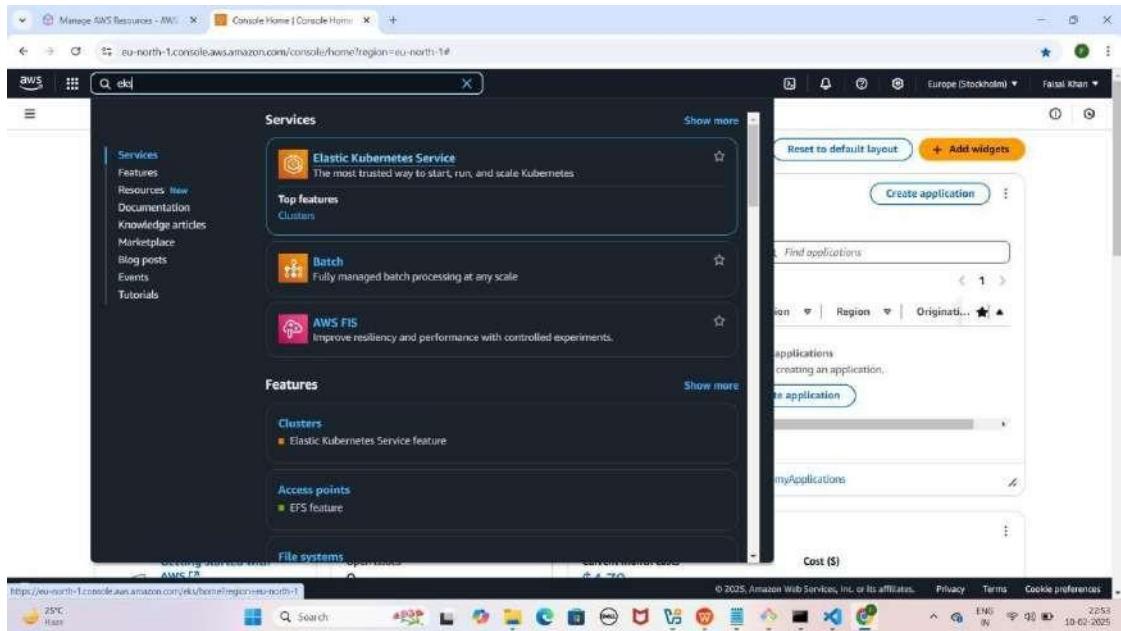
## Part 1: EKS Cluster Creation and Setup

## **Step 1: AWS Login aur EKS Cluster Create Karna**

1. AWS login karo
  2. Search bar mein "EKS" likho aur open karo
  3. "Create Cluster" pe click karo
  4. Custom configuration select karo aur "EKS Auto Mode - New" disable karo
  5. Cluster Configuration:
    1. Cluster Name: Faisal name karo ---> (Aapka Name Likho)
    2. Cluster IAM Role: AmazonEKSAutoClusterRole select karo
  6. Scroll down karo, "Secrets encryption" enable karo aur KMS key select karo
  7. Next pe click karo

## YE KUCH ISTARHA LAGEGA





Screenshot of the AWS CloudWatch Metrics console showing the creation of an EKS cluster. The user is on Step 1: Configure cluster, specifically choosing the 'Custom configuration' option. They are naming the cluster 'eks-test-cluster'. The IAM role selected is 'AmazonEKSAutoClusterRole'. The Kubernetes version is set to '1.31'. The upgrade policy is set to 'Standard'.

**Configure cluster**

**Configuration options - new**

- Quick configuration (with EKS Auto Mode) - new
- Custom configuration

**EKS Auto Mode - new**

Use EKS Auto Mode

EKS automatically runs cluster tasks for compute, storage, and networking. When a new pod can't fit existing nodes, EKS creates a new node. EKS combines cluster infrastructure managed by AWS with integrated Kubernetes capabilities to meet application compute needs. [View pricing](#)

**Cluster configuration**

**Name**

Enter a unique name for this cluster. This property cannot be changed after the cluster is created.

**Cluster IAM role**

Select the Cluster IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This cannot be changed after the cluster is created. To create a new custom role, follow the instructions in the [Amazon EKS User Guide](#).

**Kubernetes version settings**

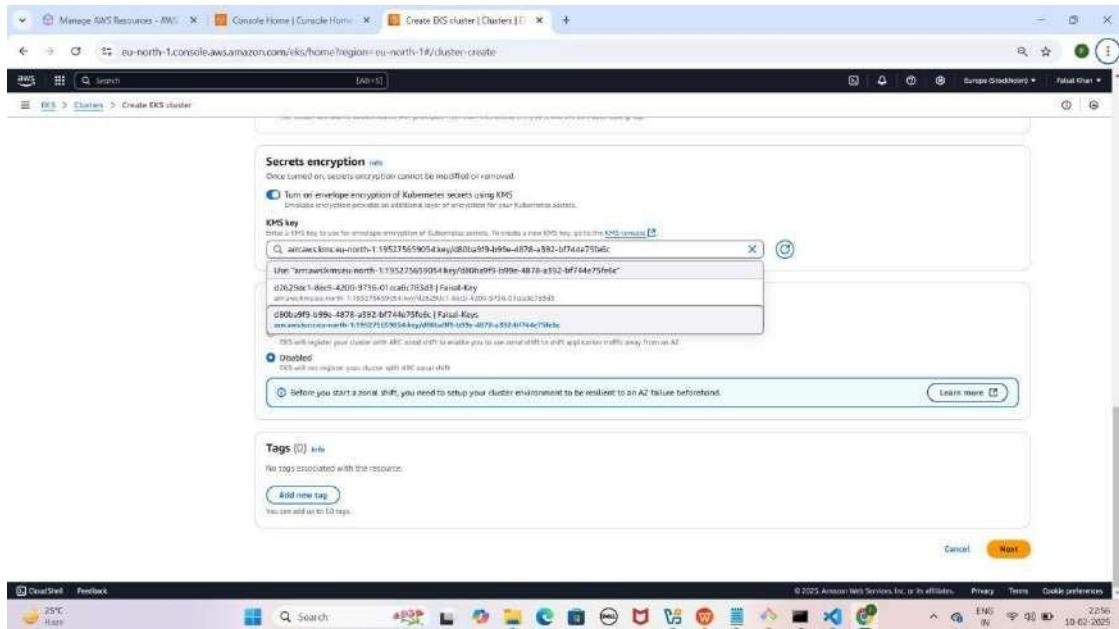
**Kubernetes version**

Selected Kubernetes version for this cluster:

**Upgrade policy**

Choose one of the following options. You can switch the setting later while the standard support period is in effect.

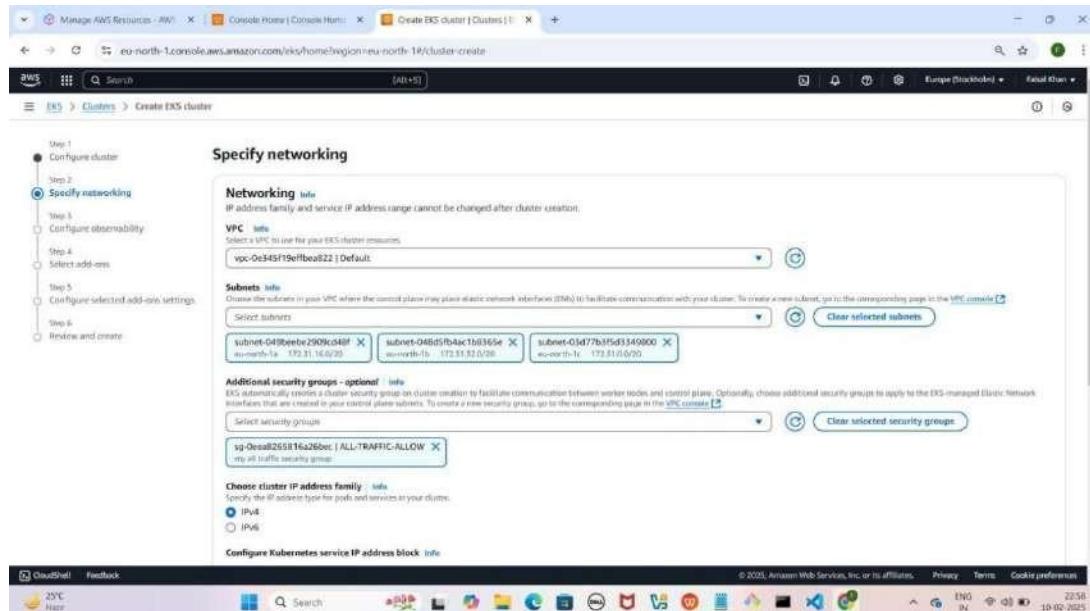
- Standard
- This option supports the Kubernetes version for 18 months after the release date. There is no additional cost. When standard support ends, your cluster will be auto-upgraded to the next version.
- Extended
- This option supports the Kubernetes regular for 28 months after the release date. The extended support period has an additional hourly cost that begins after the standard support period ends. When extended support ends, your cluster will be auto-upgraded to the next version.

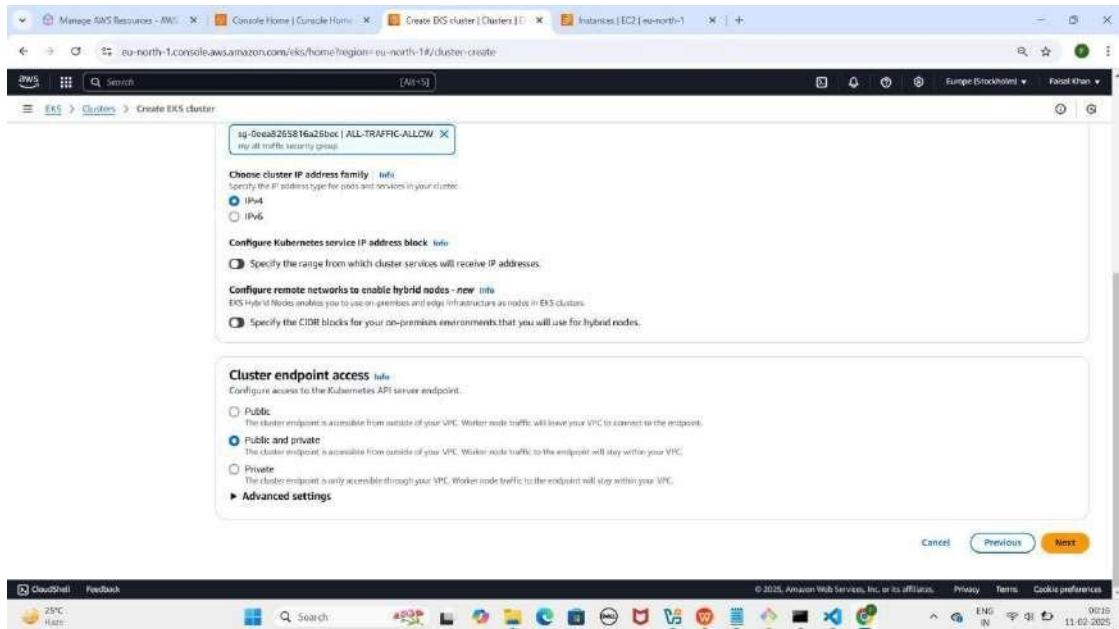


## Step 2: Networking Configuration

1. **Specify networking section mein jao**
2. **"Additional Security Groups - Optional" mein "ALL-TRAFFIC-ALLOW" select karo**
3. **Next pe click karo**

YE KUCH ISTARHA LAGEGA

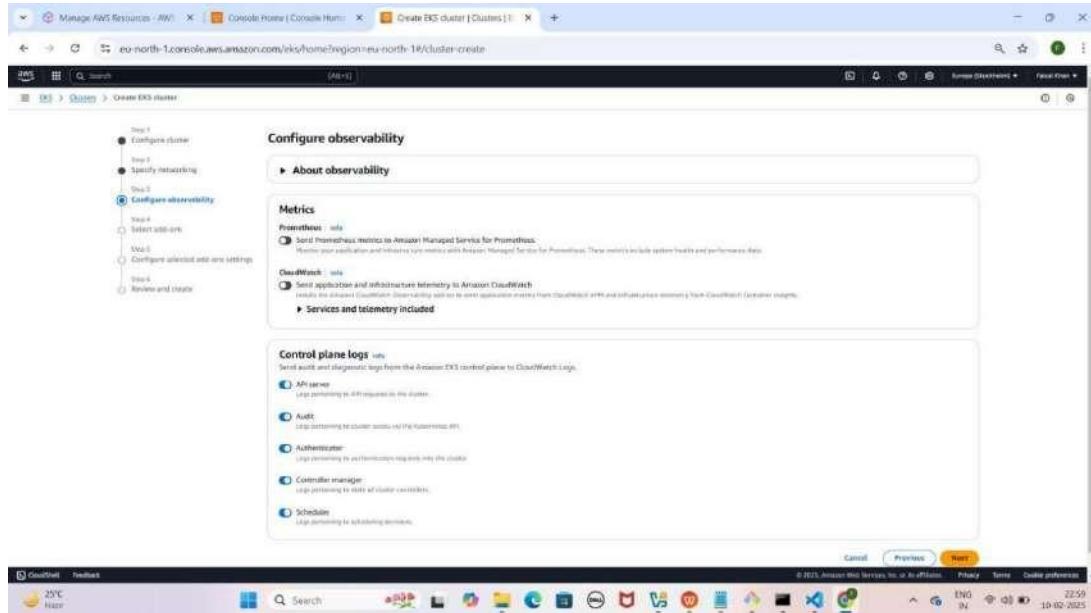




## Step 3: Observability Configure Karna

1. "Configure observability" section mein
2. "Control Plane Logs" enable karo
3. Next pe click karo

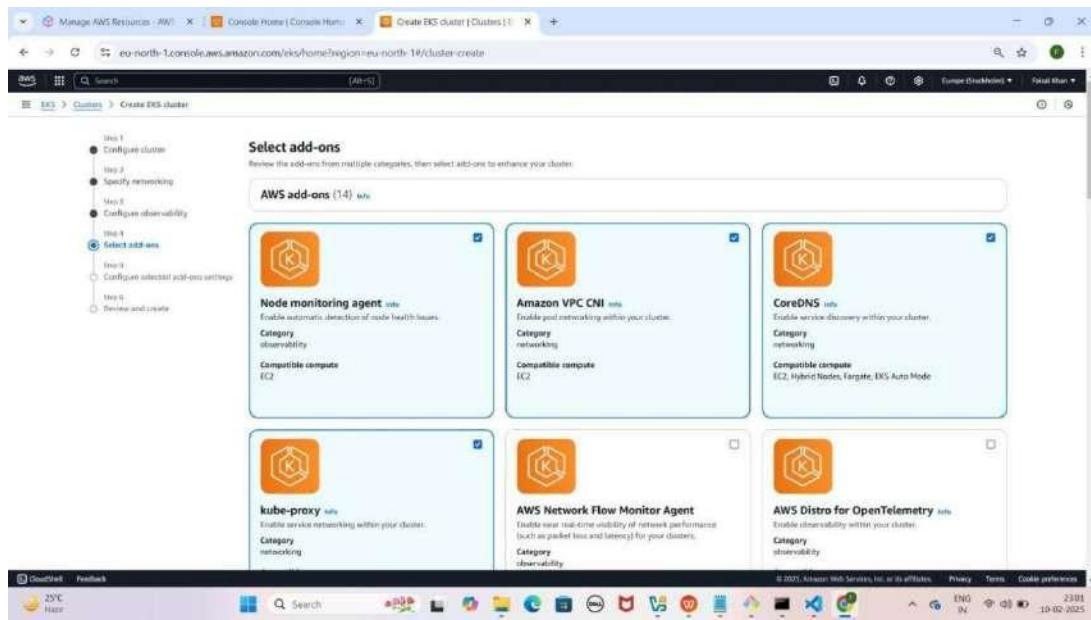
YE KUCH ISTARHA LAGEGA



## Step 4: Add-ons Configure Karna

1. "Select add-ons" section mein kuch default selected add-ons honge, unko waise hi rehne do
2. Next pe click karo
3. "Amazon VPC CNI" section mein:
4. "Pod Identity IAM Role for Service Account: PODS-ROLE" select karo
5. Next pe click karo

YE KUCH ISTARHA LEGEGA



Screenshot of the AWS EKS Cluster Creation Wizard - Step 3: Configure Add-ons.

The screenshot shows the configuration of add-ons for an EKS cluster. The wizard has reached step 3, "Configure selected add-ons settings".

**Configure selected add-ons settings**

The left sidebar lists the steps: Step 1 (Configure cluster), Step 2 (Specify networking), Step 3 (Configure observability), Step 4 (Select add-ons), Step 5 (Review and create), and Step 6 (Configure selected add-ons settings). Step 6 is currently selected.

**Node monitoring agent**

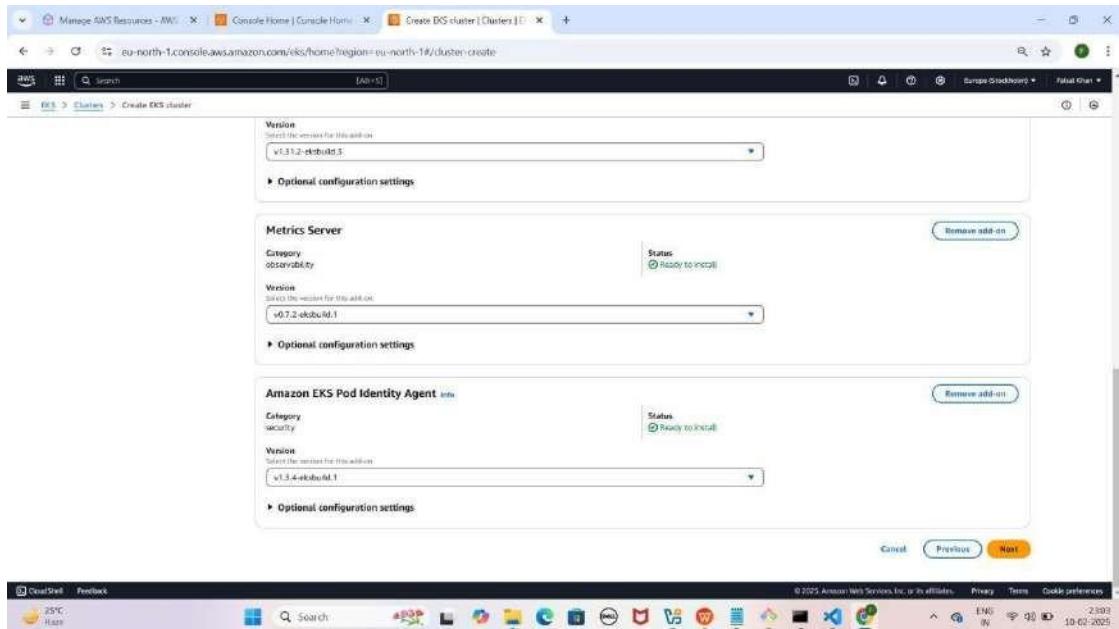
Category: observability  
Status: Ready to install  
Version: v1.0.2-4504b12  
Optional configuration settings

**Amazon VPC CNI**

Category: networking  
Status: Ready to install  
Version: v1.0.2-7a09e54  
Optional configuration settings

**Next Step**

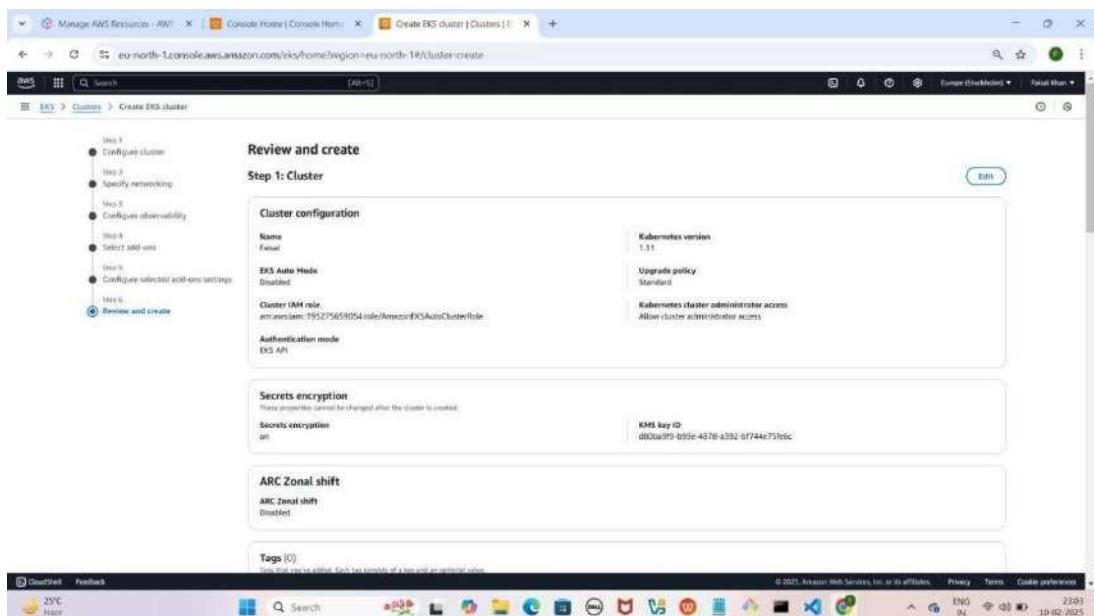
At the bottom of the screen, the taskbar shows the date as 10.02.2025 and the time as 23:02.



## Step 5: Cluster Create Karna

1. "Review and Create" section mein scroll down karo aur "Create" pe click karo
2. "Clusters" section mein jao, Status "Creating" hogा
3. 15-20 min tak wait karo aur har 5 min baad refresh karo
4. Jab status "Active" hojaye, tab next step karo

YE KUCH ISTARHA LAGEGA



Screenshot of the AWS EKS Cluster Creation Step 5: Versions screen.

The screen shows the selected add-ons version table:

Add-on name	Version
cni-cni	v1.11.3-eksbuild.1
eks-node-monitoring-agent	v1.0.2-eksbuild.2
eks-pod-identity-agent	v1.3.4-eksbuild.1
kube-proxy	v1.31.2-eksbuild.3
metrics-server	v0.7.2-eksbuild.1
tpc-cni	v1.10.0-eksbuild.1

Below the table is the EKS Pod Identity section:

Add-on name	IAM role	Service account
tpc-cni	arn:aws:iam::195375659014:role/PODS_ROLE	aws-node

At the bottom right are the "Cancel", "Previous", and "Create" buttons.

Screenshot of the AWS EKS Cluster Overview screen for cluster "Faisal".

The cluster status is "Creating". Other details include:

- Kubernetes version: v1.31
- Support period: Standard support until November 26, 2025
- Provider: EKS
- OpenID Connect provider URL: https://FFEDB8825A6EA118E571B06F19D4A88D.ig7.eu-north-1.eks.amazonaws.com
- Cluster IAM role ARN: arn:aws:iam::195375659014:role/AmazonEKSAutoClusterRole
- Cluster ARN: arn:aws:eks:eu-north-1:195375659014:cluster/Faisal
- Platform version: v1.31

At the bottom left is the "EKS Auto Mode" link.

The screenshot shows the AWS EKS Cluster Management interface. A blue success message at the top states: "Add-on(s) eks-node-monitoring-agent, vpc-cni, coredns, kube-proxy, metrics-server, eks-pod-identity-agent successfully added to cluster Faisal." Below this, a table lists the cluster "Faisal". The table columns include Cluster name, Status, Kubernetes version, Support period, Upgrade policy, Created, and Provider. The cluster details are: Cluster name: Faisal, Status: Active, Kubernetes version: 1.31, Support period: Standard support until November 26, 2025, Upgrade policy: Standard, Created: a few seconds ago, and Provider: EKS.

## Step 6: Node Group Create Karna

1. Cluster open karo, "Compute" section mein jao
2. "Add Node Group" pe click karo
3. Node Group Configuration:
  - **Node Group Name:** Faisal-Node name karo
  - **Node IAM Role:** AmazonEKSAutoNodeRole select karo
4. Next pe click karo

# YE KUCH ISTARHA LAGEGA

The screenshot shows two views of the AWS EKS Cluster Faisal dashboard. The top view displays the main cluster details, including status (Active), Kubernetes version (1.31), support period (Standard support until November 26, 2025), and provider (EKS). The bottom view shows the Compute tab, which lists nodes (0) and node groups (0). Both views include navigation tabs for Overview, Resources, Compute, Networking, Add-ons, Access, Observability, Update history, and Tags.

**Faisal Cluster Details:**

- Status: Active
- Kubernetes version: 1.31
- Support period: Standard support until November 26, 2025
- Provider: EKS

**Compute Tab (Nodes):**

- Nodes (0)
- No Nodes (Message: This cluster does not have any Nodes, or you don't have permission to view them.)

Screenshot of the AWS EKS Node Groups management interface:

**Node groups (0)**

No node groups

This cluster does not have any node groups.

**Add node group**

**Fargate profiles (0)**

No Fargate profiles

This cluster does not have any Fargate profiles.

**Add Fargate profile**

**Add node group** (Success message: Add-on(s) eks-node-monitoring-agent, vpc-cni, coredns, kube-proxy, metrics-server, eks-pod-identity-agent successfully added to cluster Faisal.)

**Configure node group**

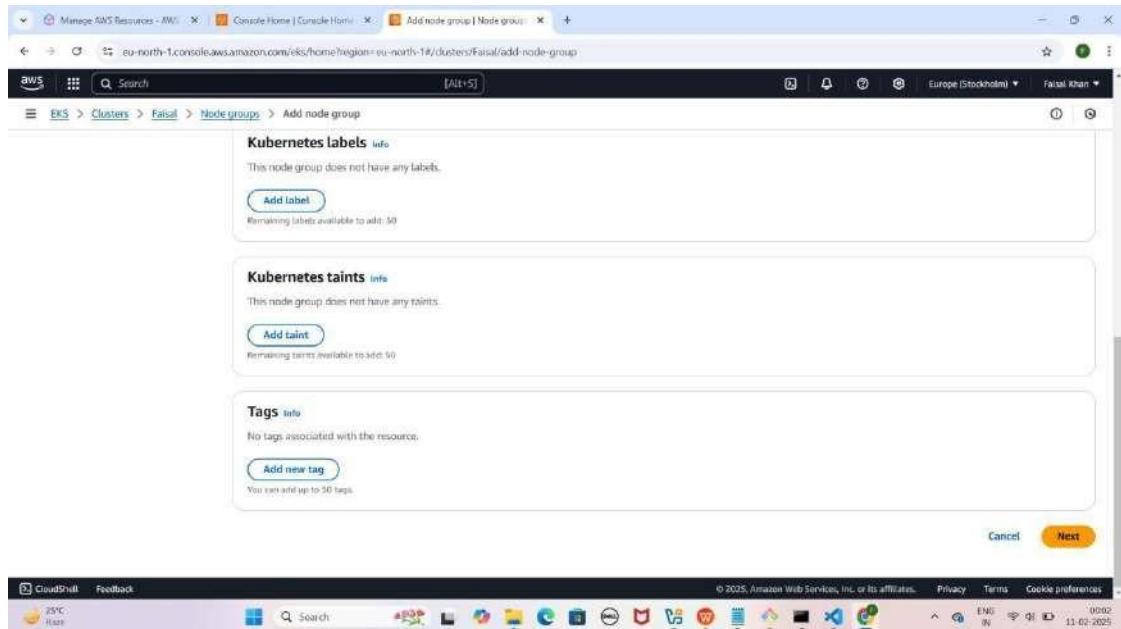
**Node group configuration**

Name: Faisal-Node

**Node IAM role**

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

**Launch template**



## Step 7: Compute aur Scaling Configuration

1. "Set Compute and Scaling Configuration" section
  - "**t3.medium**" remove karo aur "**t3.large**" add karo
  - **Disk Size:** 100 GiB set karo
  - **Desired Size, Minimum Size, Maximum Size:** 1 rakho
  - "**Node Auto Repair**" enable karo
2. **Next** pe click karo

# YE KUCH ISTARHA LAGEGA

The image consists of three vertically stacked screenshots from the AWS EKS console, showing the process of adding a node group to an existing cluster.

**Screenshot 1: Set compute and scaling configuration**

This step allows you to define the compute configuration for the node group. It includes:

- Node group compute configuration:** A note stating "These properties cannot be changed after the node group is created."
- AMI type:** Set to "Amazon Linux 2023 (x86\_64) Standard (AL2023\_x86\_64\_STANDARD)"
- Capacity type:** Set to "On-Demand"
- Instance types:** Set to "t3.large" (vCPU: 2 vCPUs, Memory: 8 GiB, Network: Up to 5 Gbps, Max. IOPS: 36)
- Disk size:** Set to "100" GB

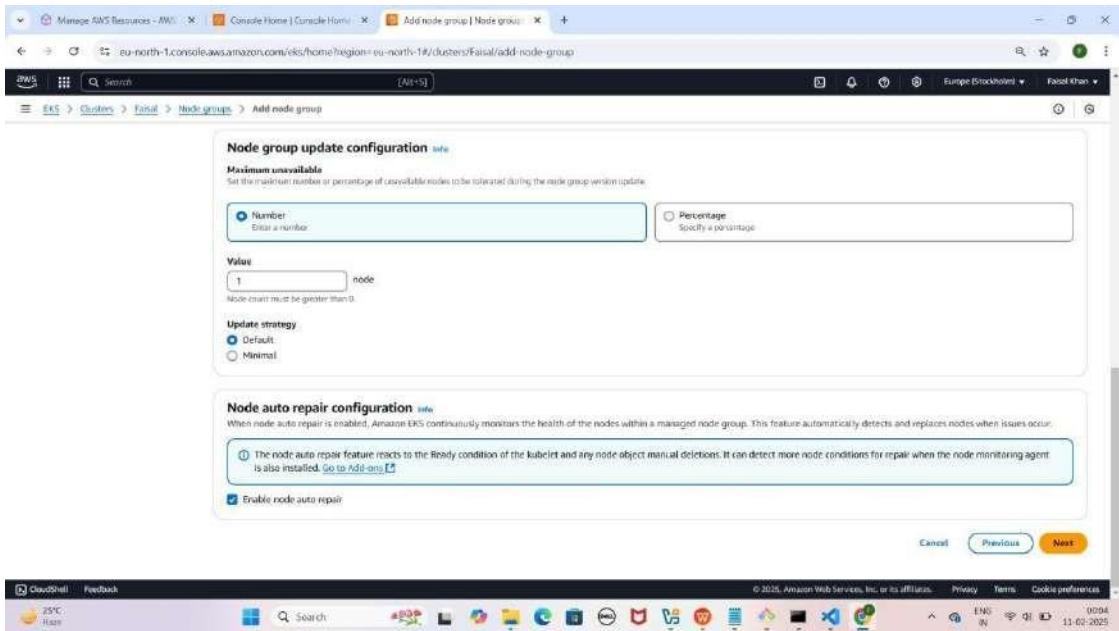
**Screenshot 2: Node group scaling configuration**

This step allows you to set the scaling parameters for the node group. It includes:

- Unset size:** Set to "100" GB
- Node group scaling configuration:**
  - Desired size:** Set to "1" nodes
  - Minimum size:** Set to "1" nodes
  - Maximum size:** Set to "1" nodes
- Node group update configuration:**
  - Maximum unavailable:** Set to "0" (Number or Percentage)

**Screenshot 3: Summary and next steps**

This step summarizes the configuration and provides a "Review and create" button to proceed to the final step.



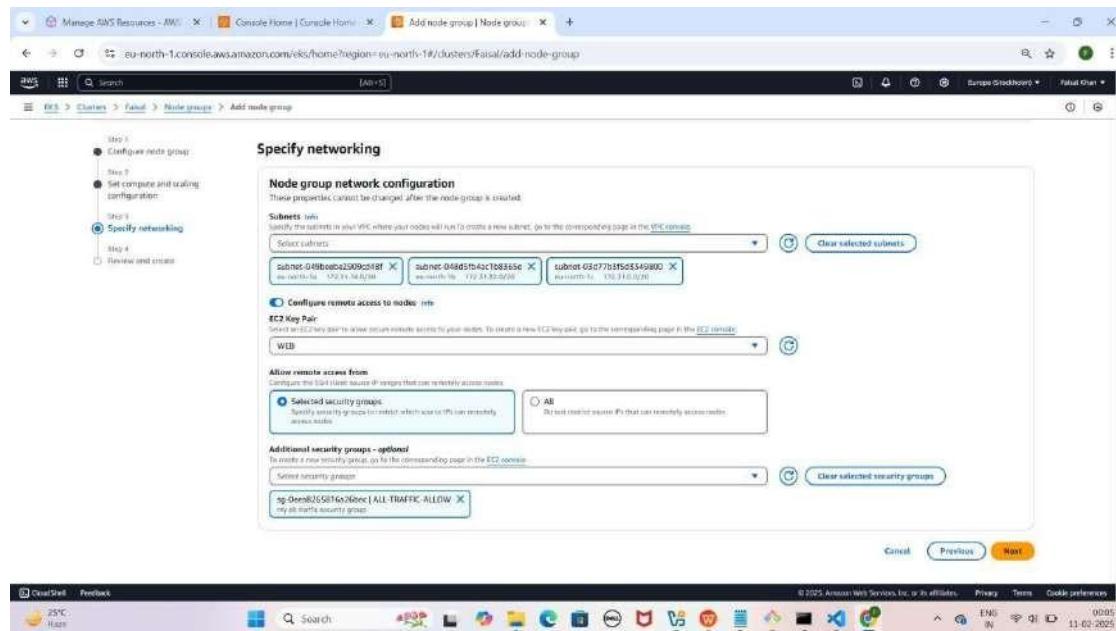
## Step 8: Node Group ki Networking Configure Karna

### 1. "Specify Networking" section

- "Configure Remote Access to Nodes" enable karo
- Key Pair select karo
- "Additional Security Groups - Optional" mein "ALL-TRAFFIC-ALLOW" select karo

### 2. Next pe click karo

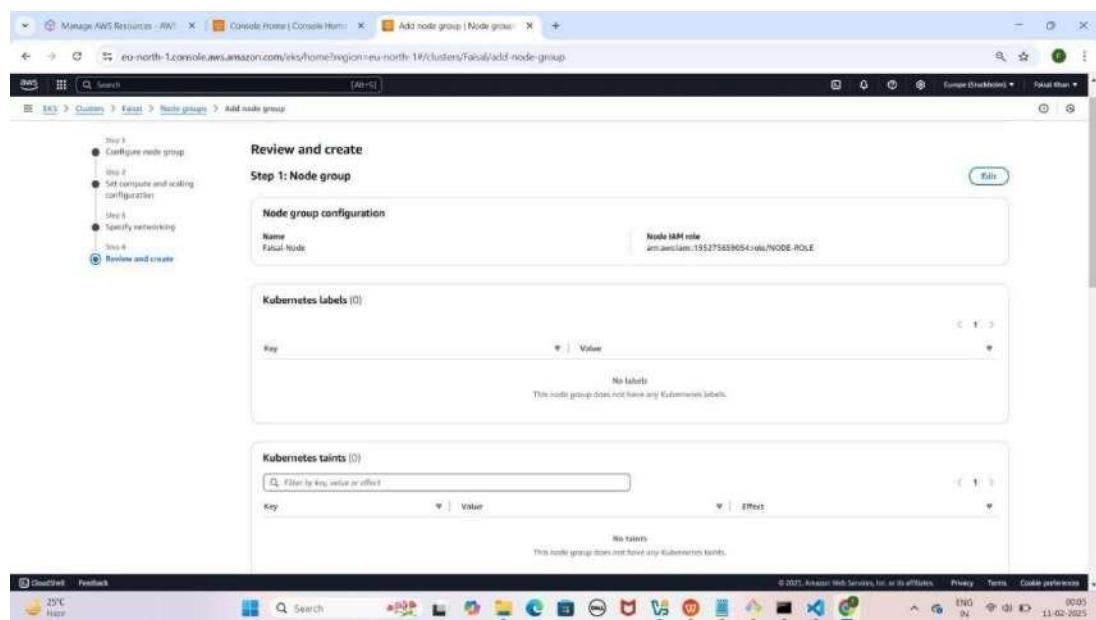
## YE KUCH ISTARHA LAGEGA

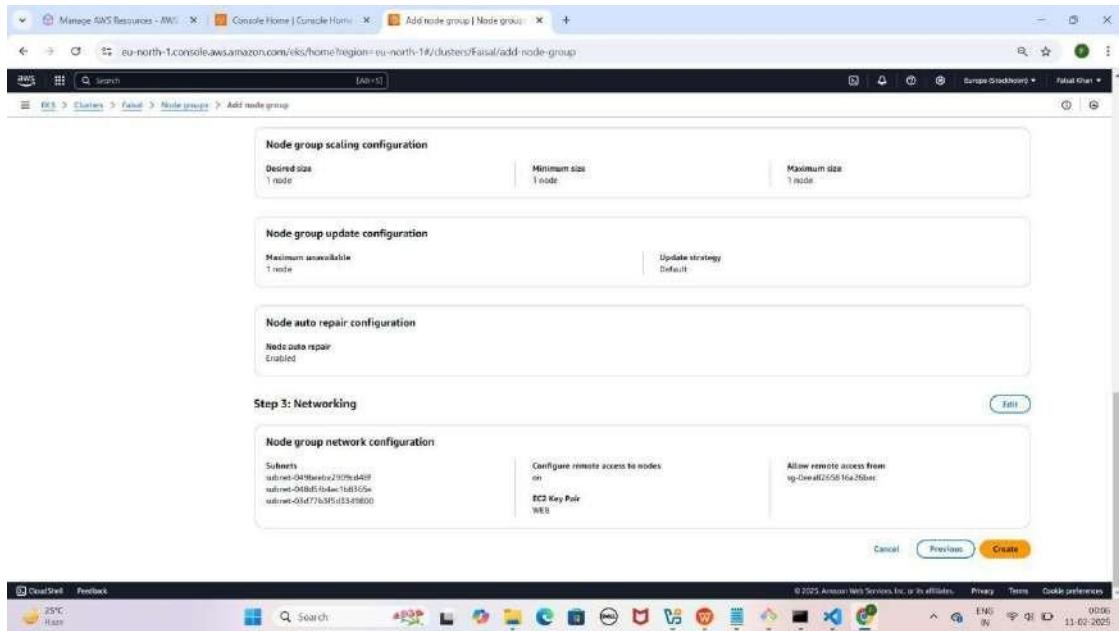


## Step 9: Node Group Review and Create Karna

1. "Review and Create" section mein jao
2. Scroll Down karo aur "Create" pe click karo

## YE KUCH ISTARHA LAGEGA

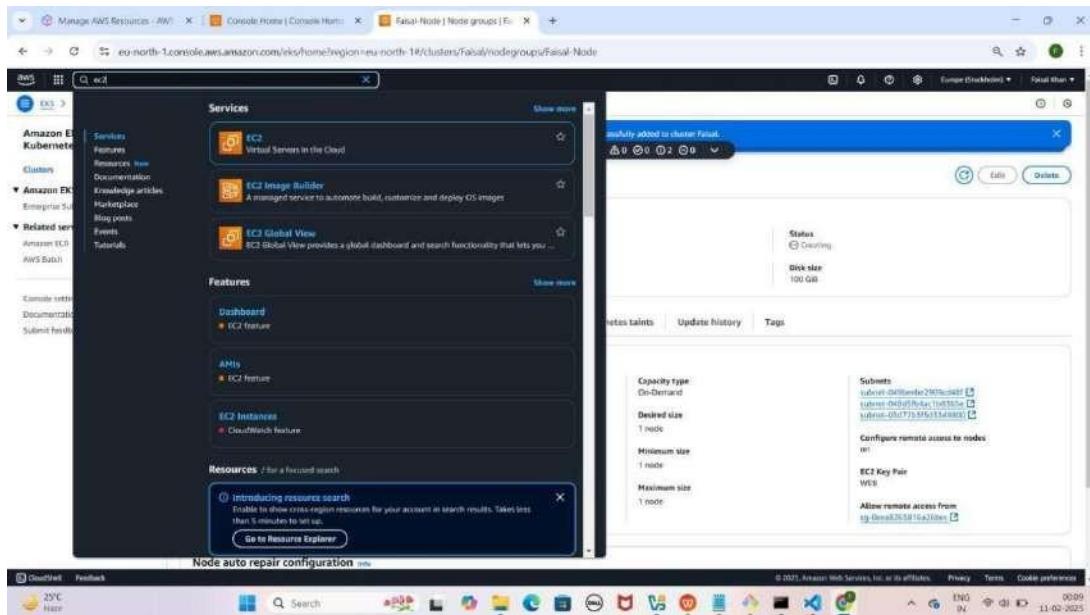




**AWS Console open karo, search bar mein "EC2" likho, Instances section mein ek running instance nazar aayega, uska name "Faisal-Node" karo.**

---> **(Aapka Name Likho)**

**YE KUCH ISTARHA LAGEGA**



Manage AWS Resources - AWS | eu-north-1 | Console Home | Compute | EC2 | eu-north-1 | Overview

eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#Overview

Search [AB+S]

Compute

## Amazon Elastic Compute Cloud (EC2)

Create, manage, and monitor virtual servers in the cloud.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.

**Launch a virtual server**

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

**Launch Instance** **View dashboard**

**Benefits and features**

**EC2 offers ultimate scalability and control**

Fully resizable compute capacity to support virtually any workload. This service is best if you want:

- Highest level of control of the entire technology stack, allowing full integration with all AWS services
- Wide variety of server size options
- Wide availability of operating systems to choose from including Linux, Windows, and macOS
- Global scalability

[Find out more about EC2](#)

**Get started**

Take our walkthroughs to help you launch an instance, learn about EC2 best practices, and set up your account.

**Get started walkthroughs** **Get started tutorial**

**Additional actions**

[View running instances](#)

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENGLISH IN 11-02-2025

CloudWatch Feedback 25C Hazel

Manage AWS Resources - AWS | eu-north-1 | Console Home | Compute | EC2 | eu-north-1 | Instances | eu-north-1 | +

eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#Instances|eu-north-1

Search [AB+S]

EC2

### Instances (1) Info

Last updated: Less than a minute ago

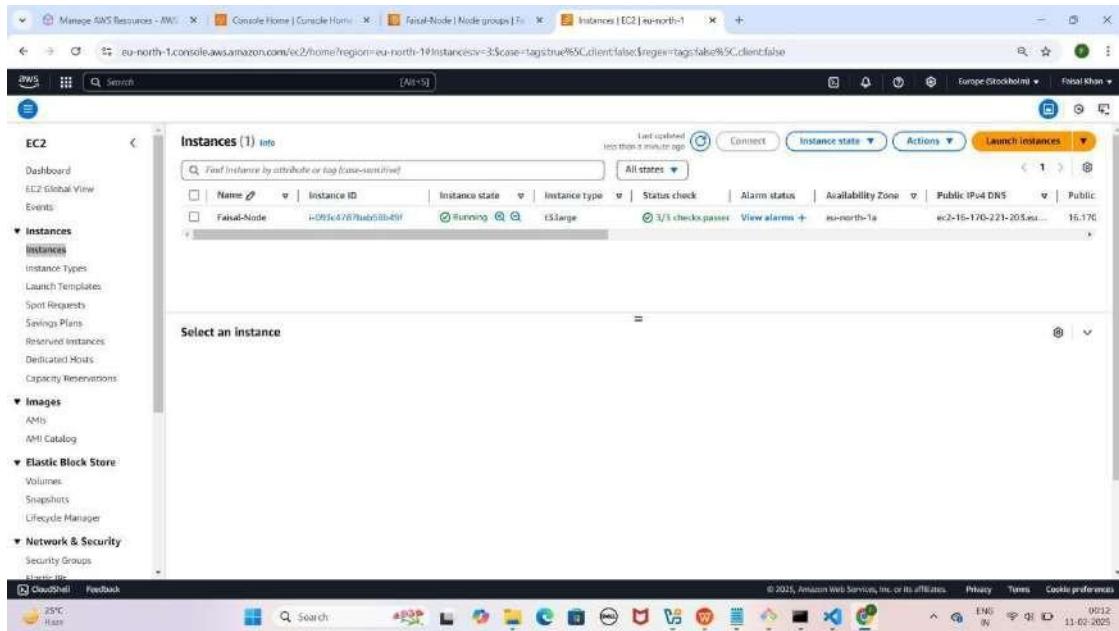
Actions ▾ **Launch instances**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
i-091e4787ab5b49f	i-091e4787ab5b49f	Running	t3.large	3/3 checks passed	View alarms	eu-north-1a	ec2-16-170-221-203.eu...	16.170

Select an instance

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENGLISH IN 11-02-2025

CloudWatch Feedback 25C Hazel



## Step 10: Security Groups Configure Karna EKS Cluster Ke liye

### 1. EKS Cluster Security Group

- **EKS Cluster** pe jao.
- **Networking** section mein jao aur **Cluster Security Group** select
- **Edit Inbound Rules** pe click karo.
- **Default rule** delete karo aur naya rule add karo:
  - **Type:** All Traffic
  - **Source:** Anywhere IPv4

### 2. Save Rules pe click karo.

# YE KUCH ISTARHA LAGEGA

The screenshot shows the AWS EKS console interface. In the top navigation bar, the URL is `eu-north-1.console.aws.amazon.com/eks/home?region=eu-north-1#/clusters`. A blue banner at the top states: "Node group creation in progress. Faisal-Node is now being created. This process may take several minutes." Below this, the "Clusters (1) info" section shows one cluster named "Faisal". The cluster details are as follows:

Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created	Provider	
Faisal	Active	1.31	Upgrade now	Standard support until November 26, 2025	Standard	13 minutes ago	EKS

On the left sidebar, under "Amazon Elastic Kubernetes Service", the "Clusters" section is expanded, showing "Amazon EKS Anywhere" and "Related services" like Amazon ECR and AWS Batch.

The screenshot shows the "Networking" tab for the "Faisal" cluster. At the top, a message says: "End of standard support for Kubernetes version 1.31 is November 26, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the [pricing page](#)." Below this, the "Cluster info" section provides basic cluster details:

Status	Kubernetes version	Support period	Provider
Active	1.31	Standard support until November 26, 2025	EKS

The "Networking" tab displays VPC settings, subnets, and security groups. The VPC section shows "vpc-0e45f19effba822" with an "info" link. The Subnets section lists three subnets: "subnet-048be0be290fc48f", "subnet-046d5fb4ac1b8365e", and "subnet-05d77hb5fd3349800", each with an "info" link. The Cluster security group is "sg-0fe4c3f789510519" with an "info" link. Additional security groups include "sg-0eef82655016a26bce" with an "info" link. API server endpoint access is set to "Public and private". The Public access source allowlist is "0.0.0.0/0 (open to all traffic)".

Screenshot of the AWS VPC Console showing the details of a security group named "sg-0fe4c3fa789510519 - eks-cluster-sg-Faisal-2096863792".

**Details:**

- Security group name:** eks-cluster-sg-Faisal-2096863792
- Security group ID:** sg-0fe4c3fa789510519
- Description:** EKS created security group applied to ENI that is attached to EKS Control Plane master nodes, as well as any managed workloads.
- VPC ID:** vpc-0e345f19cfbea822
- Owner:** 195275659054
- Inbound rules count:** 1 Permission entry
- Outbound rules count:** 1 Permission entry

**Inbound rules (1):**

Name	Type	Protocol	Port range
sgr-09a714cdcadcd801	All traffic	All	All

**Actions:** Manage tags, Edit inbound rules

Screenshot of the AWS VPC Console showing the "Edit inbound rules" page for the security group "sg-0fe4c3fa789510519 - eks-cluster-sg-Faisal-2096863792".

**Edit inbound rules:** Inbound rules control the incoming traffic that's allowed to reach the instance.

**Inbound rules:**

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-09a714cdcadcd801	All traffic	All	All	Custom	sg-0fe4c3fa789510519

**Buttons:** Add rule, Cancel, Preview changes, Save rules



Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

This security group has no inbound rules.

Add rule

Cancel Preview changes Save rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
-	All traffic	All	All	Anywhere	0.0.0.0/0

Add rule

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

CloudShell Feedback

26°C Partly cloudy

Search

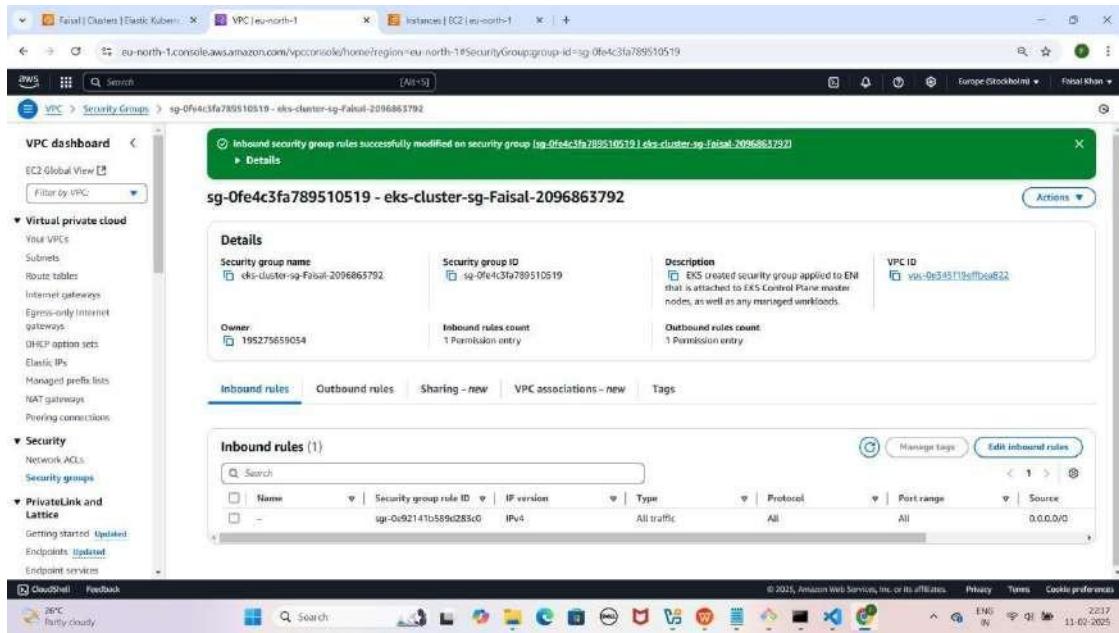
Instances | EC2 | eu-north-1

VPC | eu-north-1

eu-north-1.console.aws.amazon.com/vpc/console/home?region=eu-north-1#ModifyInboundSecurityGroupRules;securityGroupId=sg-0fe4c3fa789510519

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENGLISH 11-02-2025



## Step 11: Security Groups Configure Karna EKS Cluster Node Ke liye

### 1. EKS Cluster Node Security Group

- **EC2 Instances** section mein jao.
- **"Faisal-Node"** instance select karo.
- **Security section** mein scroll down karo aur **Outbound Rules** pe click karo.
- **Security Group ID** pe click karo.
- **Edit Inbound Rules** pe click karo.
- **Default rule** delete karo aur naya rule add karo:
  - **Type:** All Traffic
  - **Source:** Anywhere IPv4

### 2. Save Rules pe click karo.

## YE KUCH ISTARHA LAGEGA

The screenshot shows the AWS EC2 Instances page. A single instance, "Faisal-Node" (i-Oaac4ac8fcc5c7cf), is listed. The instance is running, has an E5ARGE instance type, and is in the eu-north-1b availability zone. It has a public IPv4 DNS name (ip-172-31-57-215.eu-north-1.compute.internal) and a private IP address (172.31.57.215). The instance was launched on Tuesday, February 11, 2025, at 22:15:31 GMT+0530 (India Standard Time).

The "Security" tab is selected on the instance details page, showing the IAM role (NODE-ROLE), security groups (sg-0d545cc1987897745 and sg-0f4c43fa789579519), and inbound rules.

Screenshot of the AWS EC2 Instances page showing a single instance named "Faisal-Node".

The instance details are as follows:

Name	Instance ID	Instance state	Instance type	Status check	Availability Zone	Public IPv4 DNS	Public IP
Faisal-Node	i-0aac44ac8fcc5c7cf	Running	t3large	Initializing	eu-north-1a	ec2-15-60-180-105.eu...	13.60.180.105

The Outbound rules section shows two rules:

Name	Security group rule ID	Port range	Protocol	Destination	Security groups
2 IDs	sgr-07e6955761c8b4132	22	TCP	sg-09e826501ba260cc	eks-remoteAccess-94ca7af2-27cf-1ed4-0dfe-2c565e31e34c
	sgr-0c92141b589d2850	All	All	0.0.0.0/0	eks-remoteAccess-94ca7af2-27cf-1ed4-0dfe-2c565e31e34c

Screenshot of the AWS EC2 Security Groups page showing a single security group named "eks-remoteAccess-94ca7af2-27cf-1ed4-0dfe-2c565e31e34c".

The security group details are as follows:

Name	Security group ID	Security group name	VPC ID	Description
sg-0d545acc0ff709177a3	eks-remoteAccess-94ca7af2-27cf-1ed4-0dfe-2c565e31e34c	sg-0d545acc0ff709177a3	vpc-0e53f019effbe22	Security group for all nodes in the nodegroup

Screenshot of the AWS Management Console showing the EC2 Security Groups page. The selected security group is "sg-0d545cc8978977743 - eks-remoteAccess-94ca7af2-27cf-1e4d-0a9e-2c565e31e34c".

**Details**

Security group name: eks-remoteAccess-94ca7af2-27cf-1e4d-0a9e-2c565e31e34c	Security group ID: sg-0d545cc8978977743	Description: Security group for all nodes in the nodeGroup to allow SSH access.	VPC ID: vpc-0e34519effea822e
Owner: 195275659054	Inbound rules count: 1 Permission entry	Outbound rules count: 1 Permission entry	

**Inbound rules (1)**

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-07e6955761c084132	-	-	SSH	TCP	22	sg-0e000265816a26be

**Actions**

Screenshot of the AWS Management Console showing the "Edit inbound rules" page for the selected security group.

**Edit inbound rules**

Inbound rules control the incoming traffic that's allowed to reach the instance.

**Inbound rules**

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-07e6955761c084132	SSH	TCP	22	Custom	sg-0e000265816a26be

**Add rule**

**Actions**

Screenshot of the AWS Management Console showing the same "Edit inbound rules" page, likely a duplicate or a later state of the previous screenshot.

The screenshot shows the AWS CloudShell interface with the following details:

- Header: AWS CloudShell Feedback
- Region: Europe (Stockholm)
- Page Title: ModifyInboundSecurityGroup
- URL: eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#ModifyInboundSecurityGroup?securityGroupId=sg-0d545cc897897743
- Content: "Edit inbound rules" section. Subtitle: "Inbound rules control the incoming traffic that's allowed to reach the instance." A message states: "This security group has no inbound rules." A "Add rule" button is present.
- Buttons: Cancel, Preview changes, Save rules.

The screenshot shows the AWS CloudShell interface with the following details:

- Header: AWS CloudShell Feedback
- Region: Europe (Stockholm)
- Page Title: ModifyInboundSecurityGroup
- URL: eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#ModifyInboundSecurityGroup?securityGroupId=sg-0d545cc897897743
- Content: "Edit inbound rules" section. Subtitle: "Inbound rules control the incoming traffic that's allowed to reach the instance." A message states: "This security group has no inbound rules." A "Add rule" button is present. A success message at the bottom says: "Rule added successfully." A warning message at the bottom says: "Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only."
- Buttons: Cancel, Preview changes, Save rules.

The screenshot shows the AWS CloudShell interface with the following details:

- Header: AWS CloudShell Feedback
- Region: Europe (Stockholm)
- Page Title: ModifyInboundSecurityGroup
- URL: eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#ModifyInboundSecurityGroup?securityGroupId=sg-0d545cc897897743
- Content: "Edit inbound rules" section. Subtitle: "Inbound rules control the incoming traffic that's allowed to reach the instance." A message states: "This security group has no inbound rules." A "Add rule" button is present. A success message at the bottom says: "Changes saved successfully."
- Buttons: Cancel, Preview changes, Save rules.

The screenshot shows the AWS CloudWatch Metrics interface. A single data series is plotted over time, starting from 0 and reaching a value of 1 at the current time. The chart has a light blue background with a white grid. The Y-axis is labeled 'Value' and ranges from 0 to 1. The X-axis is labeled 'Time'.

## Step 12: Project Ki GitHub Repository Clone Karen

### 1. Laptop Mein Documents Folder Mein Jao

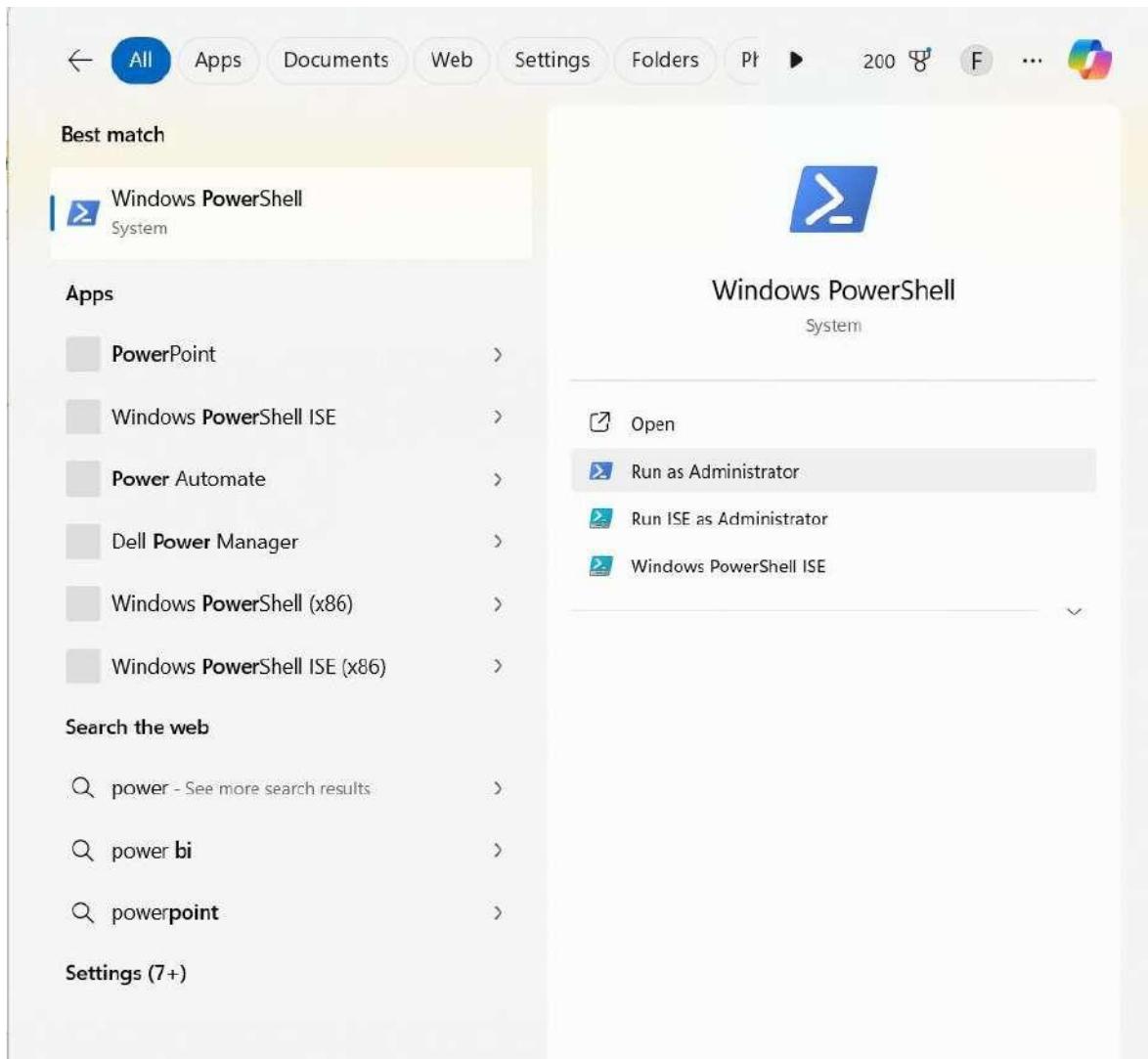
YE KUCH ISTARHA LEGAGA

The screenshot shows a Windows File Explorer window with the path 'OneDrive > Faisal - Personal > Documents'. The 'Documents' folder is expanded, showing its contents. The folder structure includes:

- ALL KUBERNETES NOTES
- commands
- controller
- Custom Office Templates
- database
- EKS-CLUSTER-LOAD-BALANCER
- EKS-CLUSTER-NODEPORTS
- fa1
- git
- KEY PAIRS
- models
- node\_modules
- edrone
- rclone-v1.68.1-windows-amd64
- routes
- Techstrime-backend
- templates
- TERRAFORM\_PROJECT
- test

## 2. PowerShell Ko "Run as Administrator" Open Karo

YE KUCH ISTARHA LAGEGA

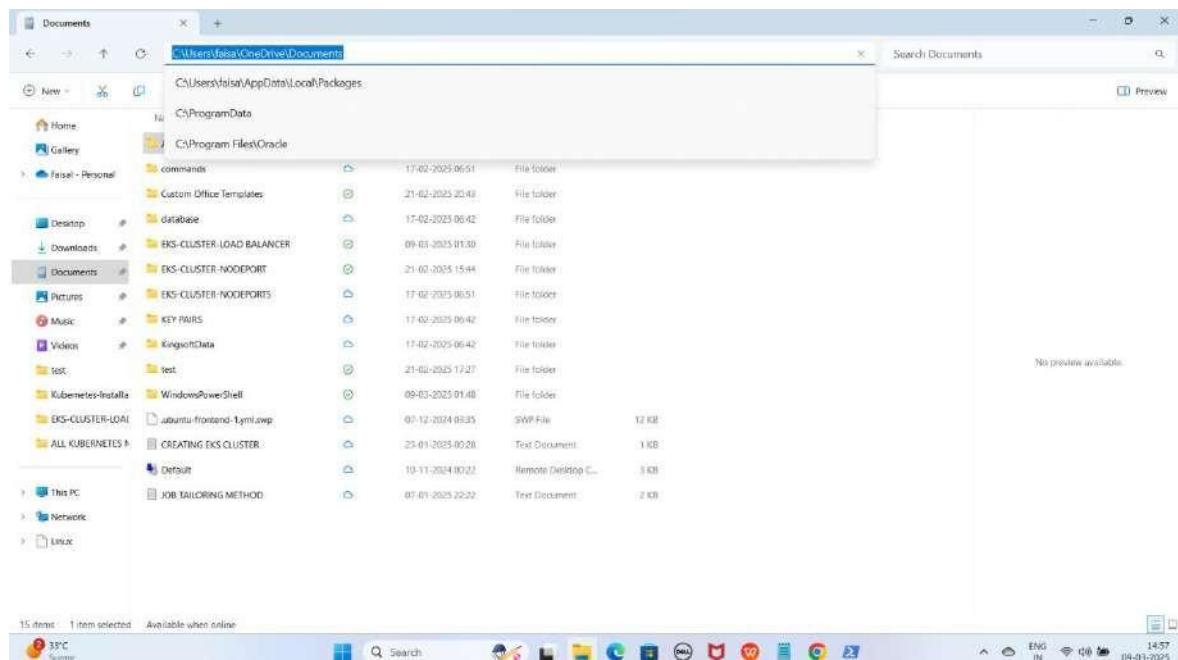


### 3. PowerShell me cd command ka use karke Documents folder ka path paste karein.

Jaisa ki mere case me, path kuch aisa hai

```
cd C:/Users/faisa/OneDrive/Documents
```

YE KUCH ISTARHA LAGEGA



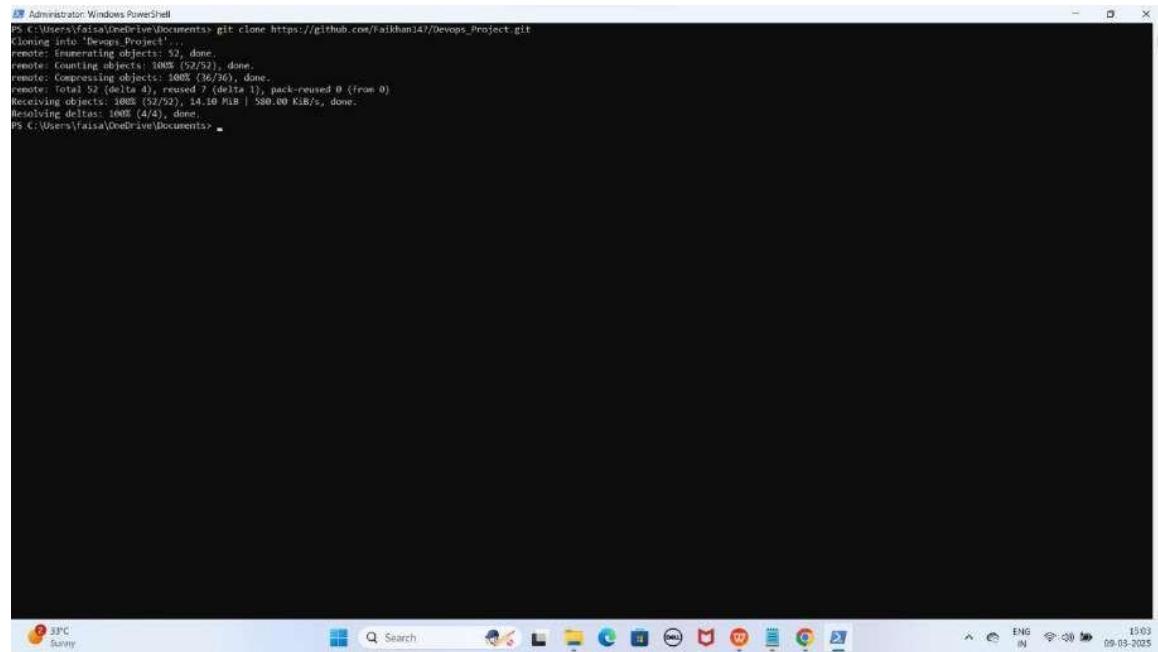
**NOTE: Isse aap Documents folder me Navigate ho jaoge.**

## 4. GitHub Repository Clone Karo

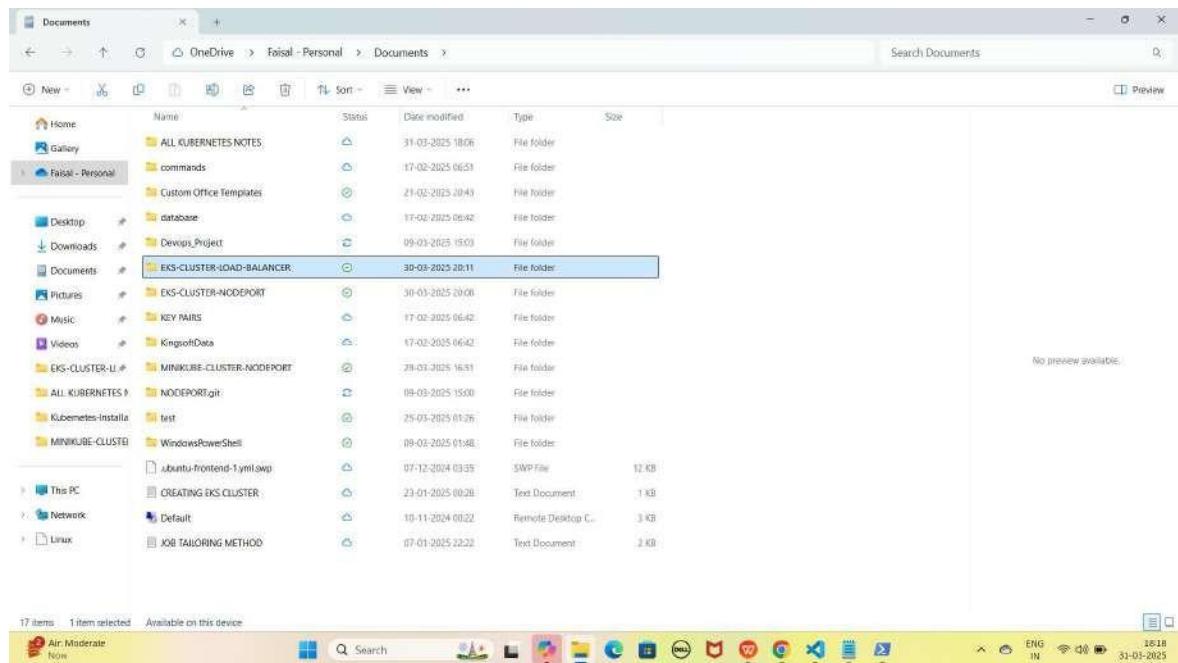
**As it is niche diya gaya URL copy karo aur PowerShell me paste karo**

```
git clone https://github.com/Faisalkhan45/EKS-CLUSTER-LOAD-BALANCER.git
```

YE KUCH ISTARHA LAGEGA



```
[PS] Admin:Windows PowerShell
PS C:\Users\faisal\OneDrive\Documents> git clone https://github.com/faisalkhan45/Devops_Project.git
Cloning into 'Devops_Project'...
remote: Enumerating objects: 1008 (93/93), done.
remote: Counting objects: 1008 (93/93), done.
remote: Compressing objects: 1008 (93/93), done.
remote: Total 1008 (delta 0), reused 9 (delta 1), pack-reused 0 (from 0)
receiving deltas: 1008 (4/4), done.
receiving deltas: 1008 (4/4), done.
receiving deltas: 1008 (4/4), done.
PS C:\Users\faisal\OneDrive\Documents>
```



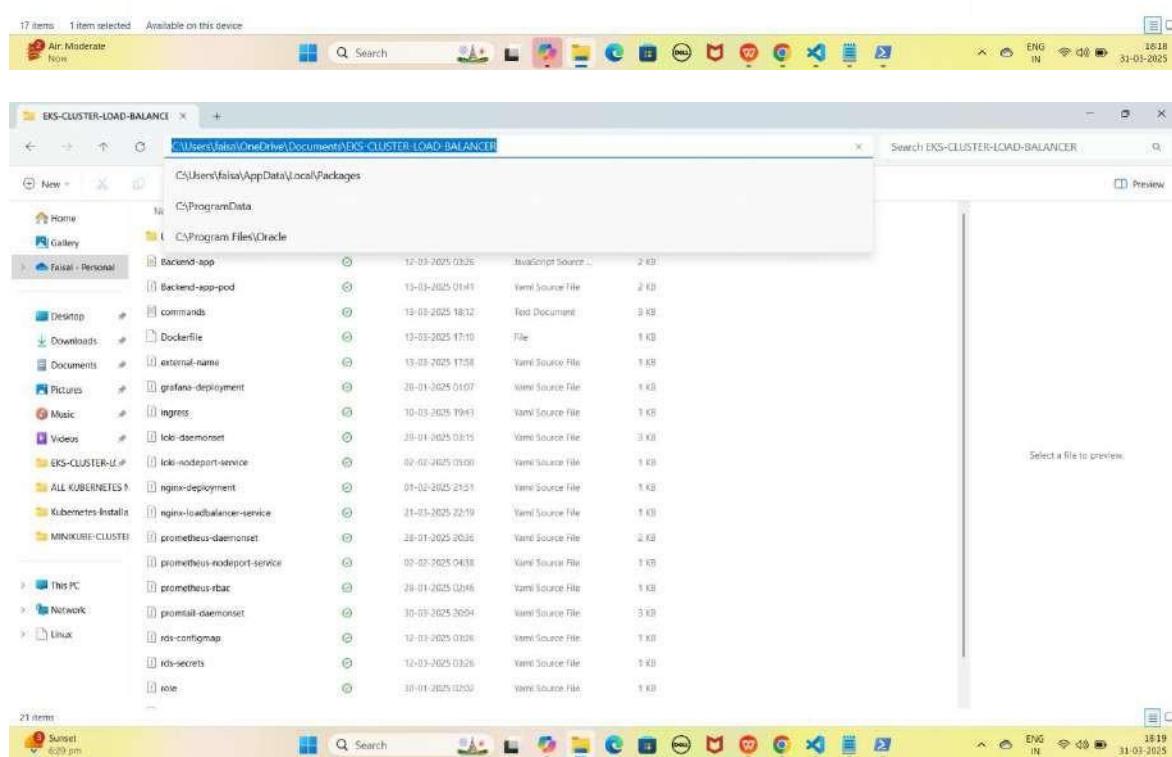
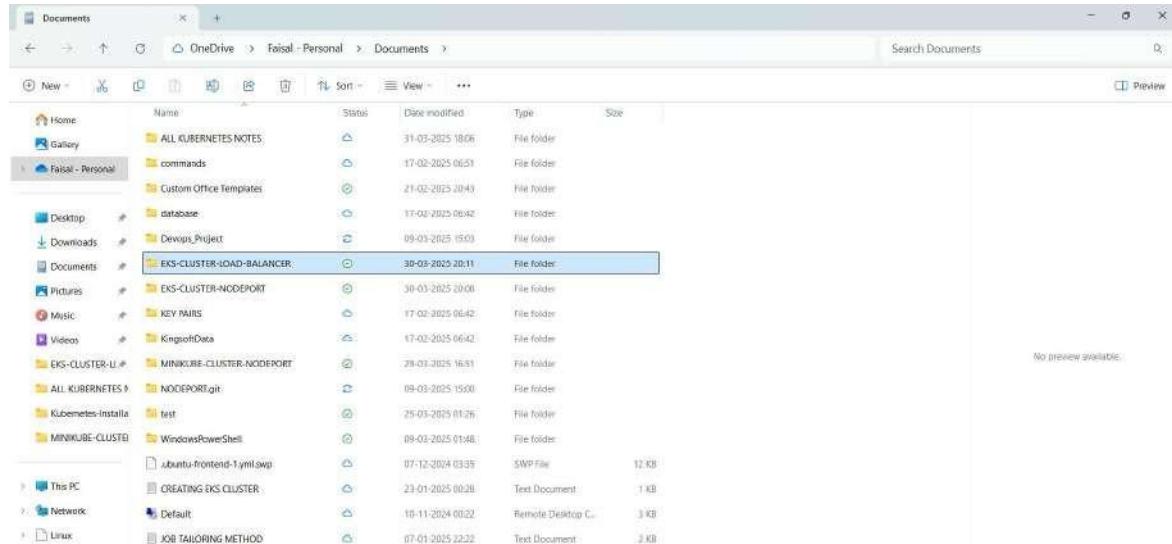
**NOTE: Repository clone hone ke baad, aapko "EKS-CLUSTER-LOAD-BALANCER" naam ka folder Documents me dikhega**

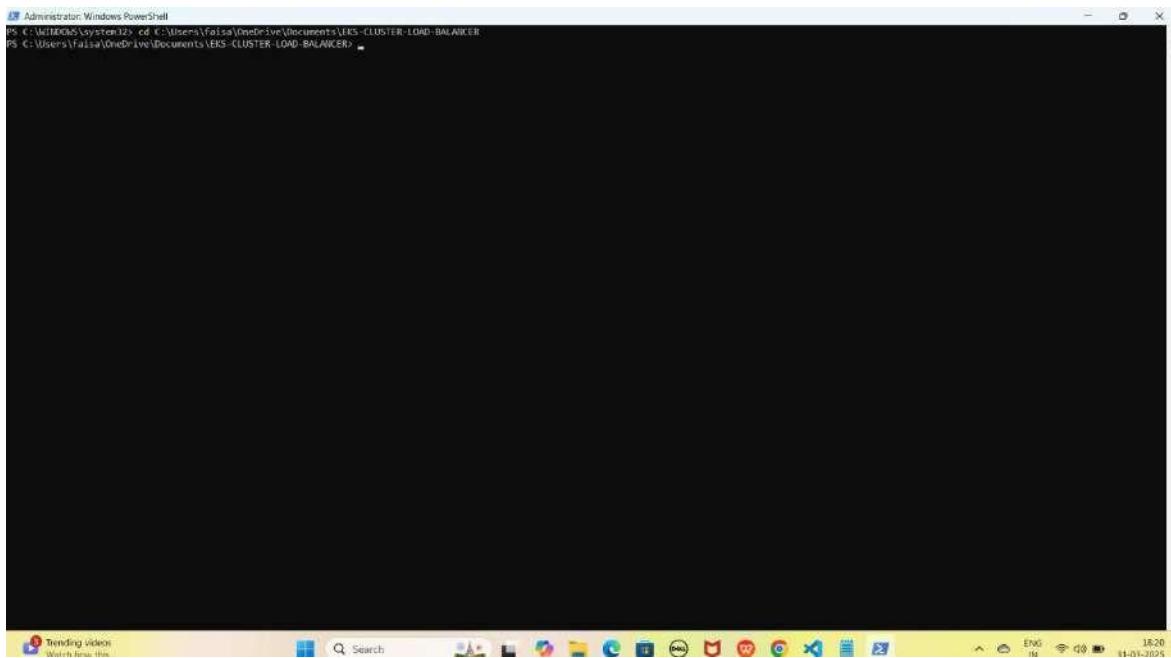
## 5. EKS-CLUSTER-LOAD-BALANCER Folder Me Jao PowerShell me cd command ka use karo aur apna path paste karo

Jaise ki mere case me path kuch aisa hogा

```
cd C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER
```

YE KUCH ISTARHA LAGEGA





```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> cd C:\Users\faizan\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER
PS C:\Users\faizan\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

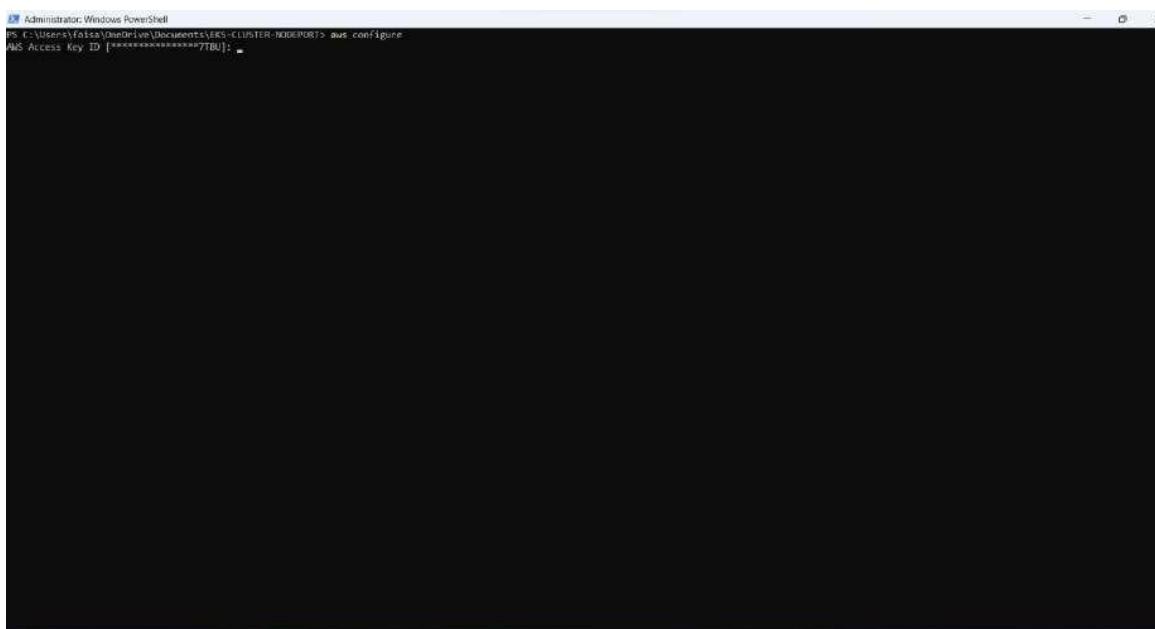
**NOTE: Is command se aap PowerShell me EKS-CLUSTER-LOAD-BALANCER folder me Navigate ho jaoge.**

## Step 13: AWS CLI Configure Karo

### 1. Configure karne ke liye ye command run karo

```
aws configure
```

YE KUCH ISTARHA LAGEGA

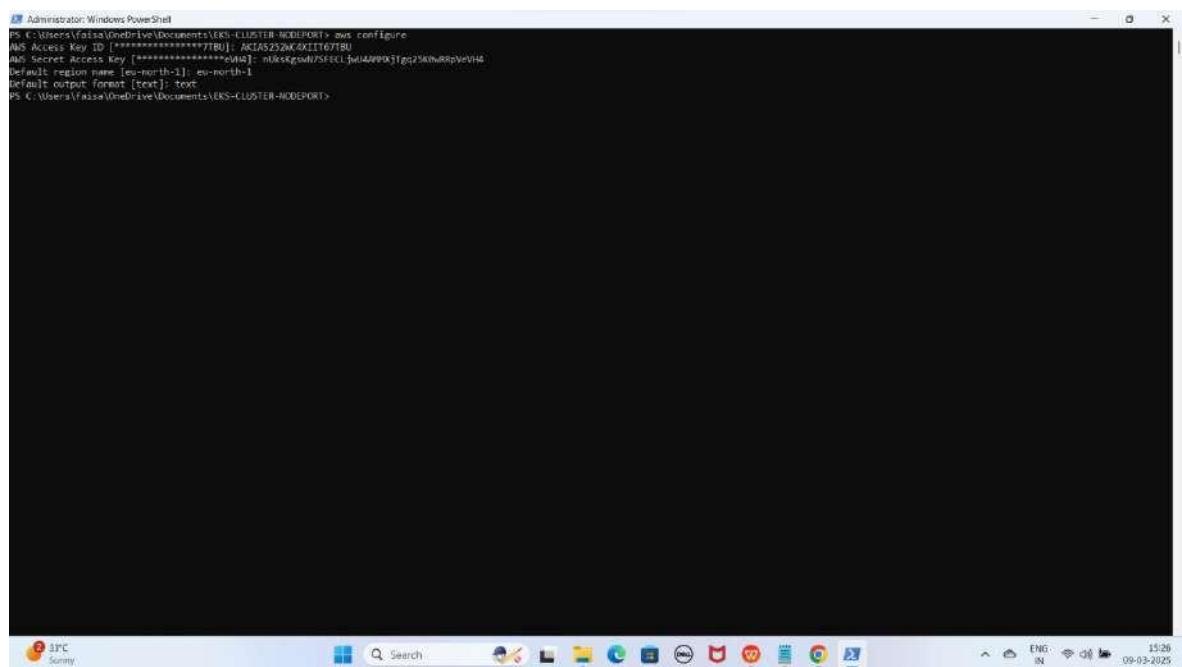


```
Administrator: Windows PowerShell
PS C:\Users\faizan\OneDrive\Documents\EKS-CLUSTER-NODEPORTS> aws configure
AWS Access Key ID [*****7TBU]:
```

## 2. Details Enter Karo

- **AWS Access Key ID:** XXXXXXXXXXXXXXXXXX
- **AWS Secret Access Key:** XXXXXXXXXXXXXXXXXXXXXXXXX
- **Default Region:** eu-north-1
- **Output Format:** text

YE KUCH ISTARHA LAGEGA

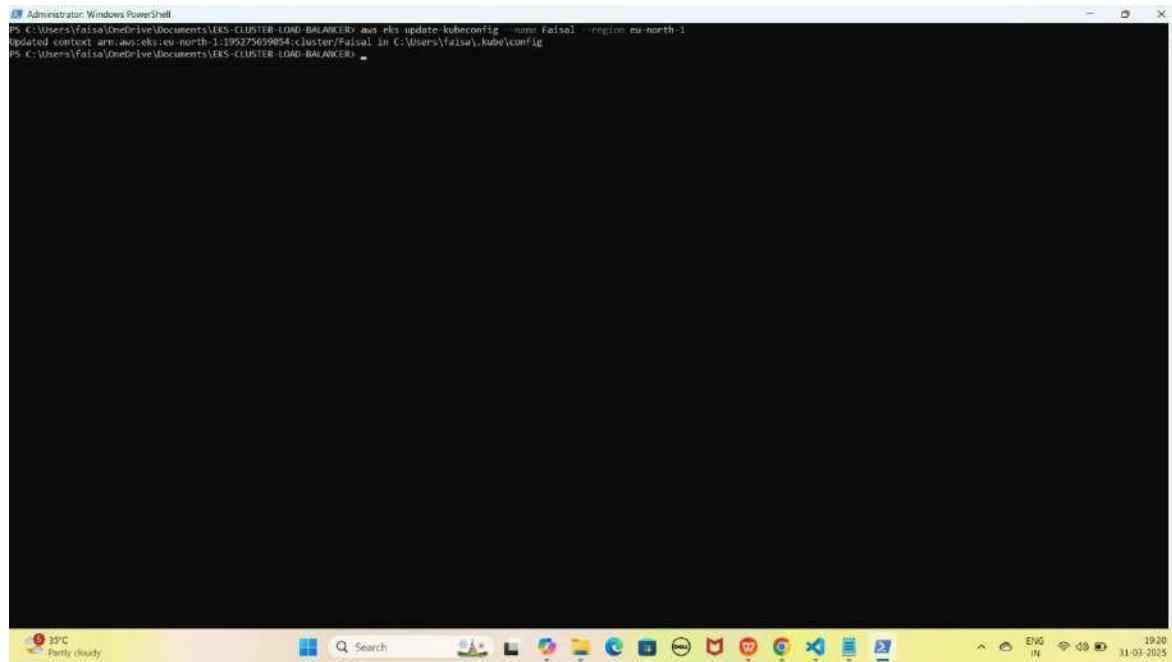


```
[Administrator: Windows PowerShell
PS C:\Users\faisal\OneDrive\Documents\EKS-CLUSTER-NODEPORT> aws configure
AWS Access Key ID [*****7IBU]: ACIAZ52XEXLT67IBU
AWS Secret Access Key [*****pM4]: m0ksgodl7SEELjwI4AWX3jgq2Kb88pVeVH4
Default region [eu-north-1]: eu-north-1
Default output format [text]: text
PS C:\Users\faisal\OneDrive\Documents\EKS-CLUSTER-NODEPORT>
```

## 3. EKS Clsuter se Connect karne ke liye ye command run kariye

```
aws eks update-kubeconfig --name Faisal --region eu-north-1
```

## YE KUCH ISTARHA LAGEGA



```
[Administrator: Windows PowerShell]
PS C:\Users\Faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> aws eks update-kubeconfig --name Faisal --region eu-north-1
Updated context arn:aws:eks:eu-north-1:199275659054:cluster/Faisal in C:\Users\Faisa\kube\config
PS C:\Users\Faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

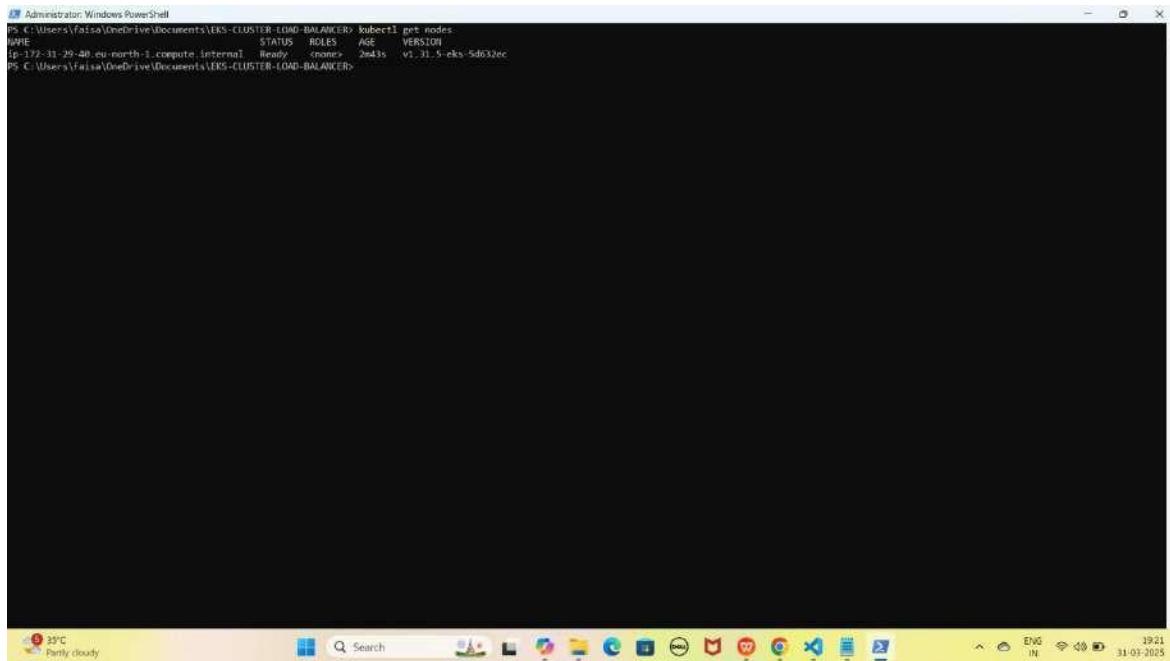
**NOTE:** Yaha par Faisal mere cluster ka name hai aap ke case me cluster  
ka name alag hosakta hai to according to your cluster name  
command aapke cluster name se Replace karo or ye hum  
Stockholm region me karrahe hai

## **Step 14: Nodes Check Karna**

### **1. Check Karo Ke Node Sahi Se Ready Hai Ya Nahi**

kubectl get nodes

YE KUCH ISTARHA LAGEGA



```
Administrator: Windows PowerShell
PS C:\Users\faiza\OneDrive\Documents\LEKS-CLUSTER-LOAD-BALANCER> kubectl get nodes
NAME           STATUS    ROLES   AGE     VERSION
ip-172-31-29-48.eu-north-1.compute.internal   Ready    master   2m43s   v1.31.5-eks-5d632ec
PS C:\Users\faiza\OneDrive\Documents\LEKS-CLUSTER-LOAD-BALANCER>
```

**NOTE: Agar Output Mein Node "Ready" STATUS Mein Hai To  
Setup Complete Hai.**

## Part 2: Nginx pods and Service Deployment

### Step 1: nginx-deployment.yaml File Kaam

Yeh **file NGINX ke 3 replicas create karne ke liye use hoti hai.** Isko **Kubernetes Deployment** ke andar likha jata hai.

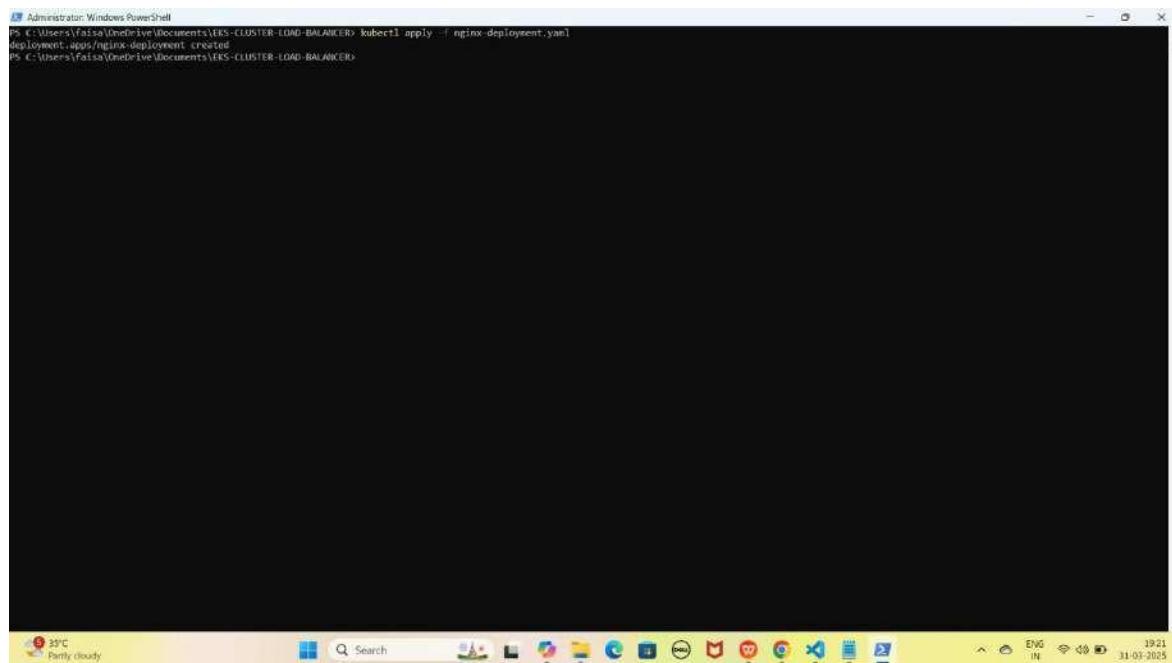
JAISE KI:-

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
```

### Deployment Apply Karo

kubectl apply -f nginx-deployment.yaml

## YE KUCH ISTARHA LAGEGA



```
Administrator: Windows PowerShell
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## Step 2: nginx-loadbalancer-service.yaml File Ka Kaam

Yeh **file NGINX** ko **LoadBalancer service** ke zariye **expose** karne ke liye use hoti hai, jisse hum usay **Load Balancer DNS** ke through **access** kar sakte hain.

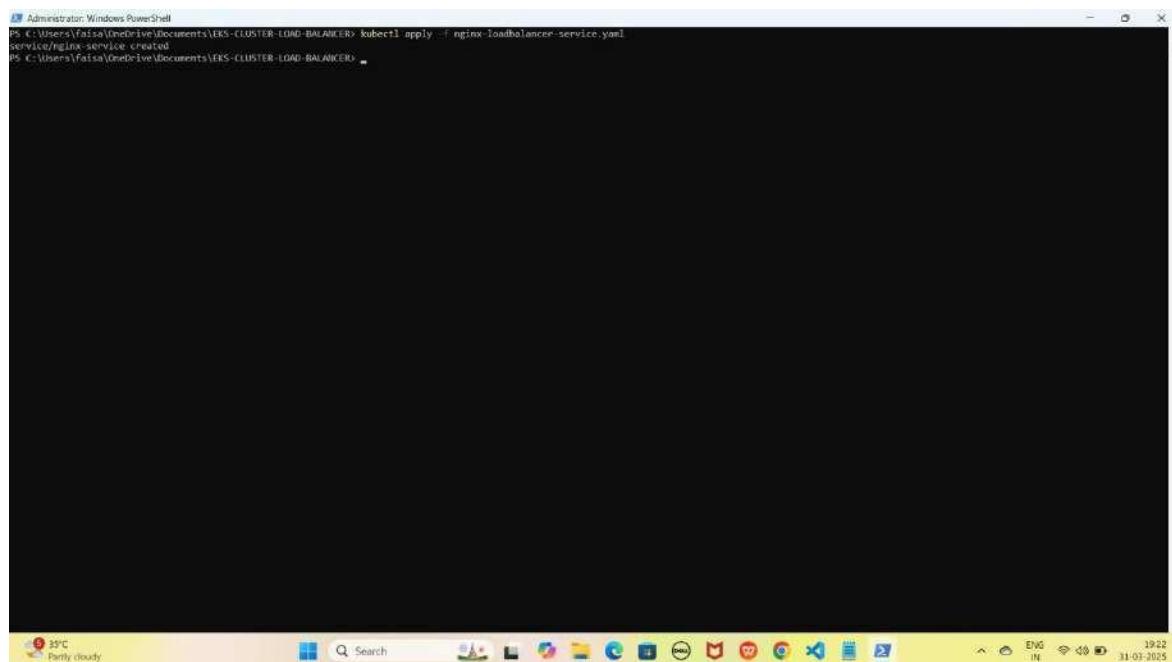
JAISE KI:-

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service-https
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

## Nginx Service Apply Karo

```
kubectl apply -f nginx-loadbalancer-service.yaml
```

## YE KUCH ISTARHA LAGEGA

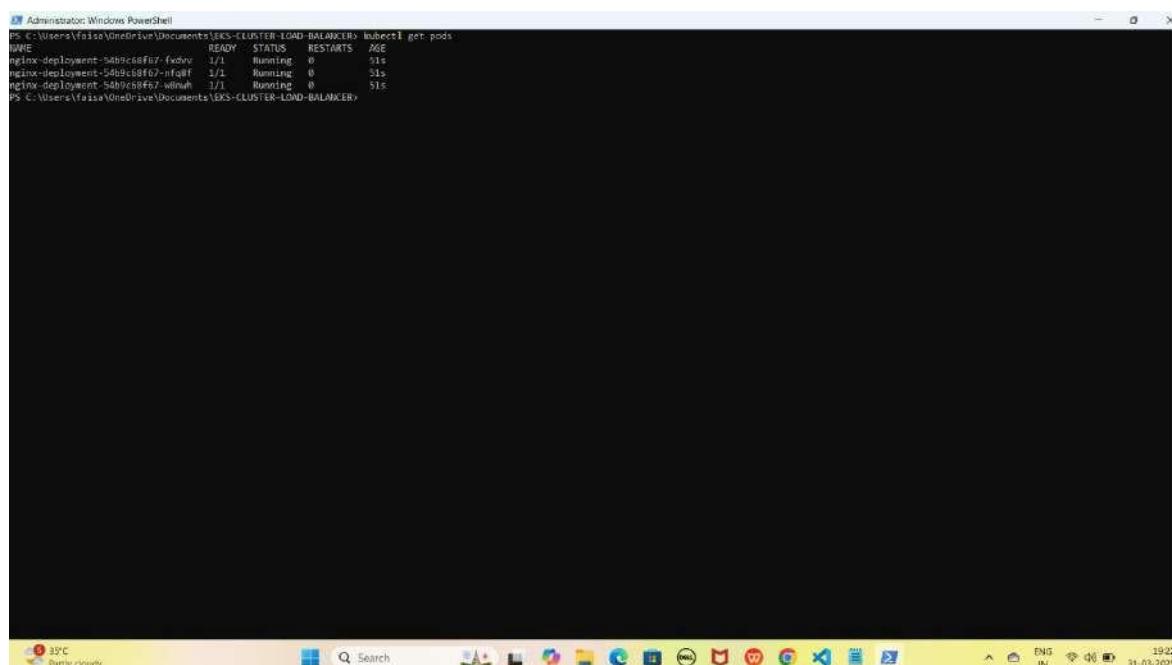


```
Administrator: Windows PowerShell
PS C:\Users\Visa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f nginx-loadbalancer-service.yaml
service/nginx-service created
PS C:\Users\Visa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

### 1. Pods check karne ke liye ye command run kariye

kubectl get pods

## YE KUCH ISTARHA LAGEGA



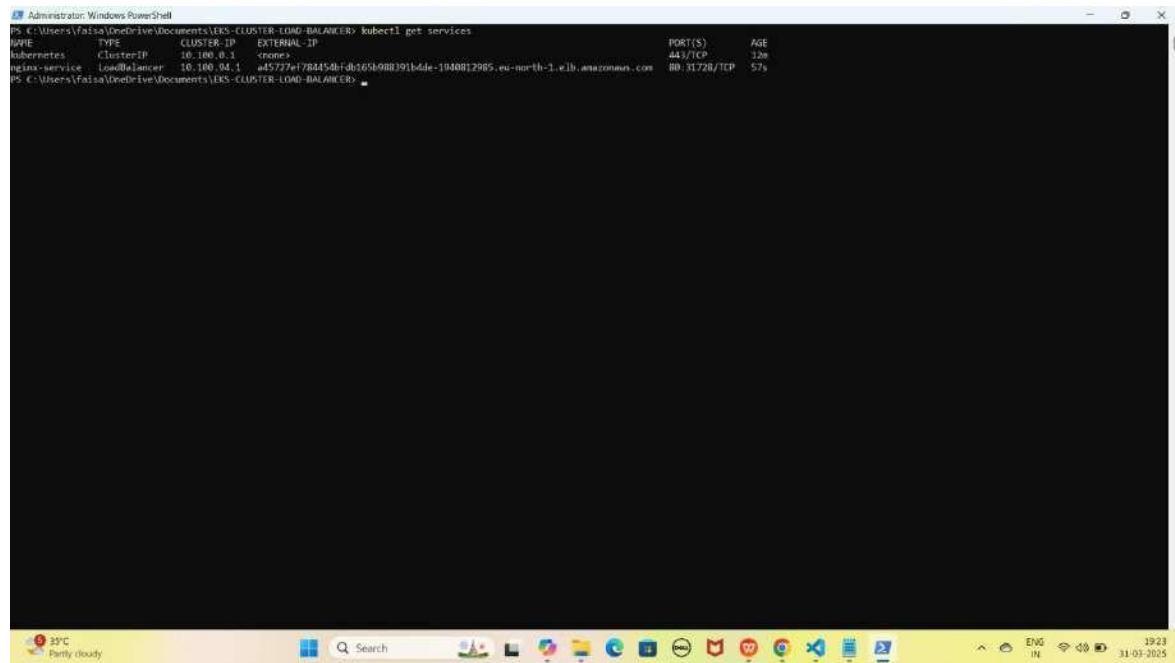
```
Administrator: Windows PowerShell
PS C:\Users\Visa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
nginx-deployment-54b9c68fb7-fxdrv  1/1     Running   0          51s
nginx-deployment-54b9c68fb7-nfqbf  1/1     Running   0          51s
nginx-deployment-54b9c68fb7-wtwhh  1/1     Running   0          51s
PS C:\Users\Visa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

**NOTE: Agar sabhi Pods ka STATUS Running show karraha hai to sab kuch sahi hai**

## 2. Services check karne ke liye ye commad run kariye

kubectl get services

YE KUCH ISTARHA LAGEGA



```
Administrator: Windows PowerShell
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl get services
NAME        TYPE      CLUSTER-IP  EXTERNAL-IP
kubernetes  ClusterIP  10.100.0.1   <none>
nginx-service  LoadBalancer  10.100.0.1   a5727ef78d5db165b988391bde-1940812985.eu-north-1.elb.amazonaws.com  80:31728/TCP  57s
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

**NOTE: Ab aapka Nginx expose ho chuka hai aapke Load Balancer ki DNS Name lekar browser me run kariye jaise ki kuch aisa hogi mere case me**

**NGINX Load Balancer DNS :**

<http://a1b2c3d4e5f6g7h8-123456789.ap-south-1.elb.amazonaws.com>

## YE KUCH AISA LAGEGA

The screenshot shows the AWS CloudShell interface with the following details:

- CloudShell** tab is active.
- Feedback** tab is visible.
- Load balancers | EC2 | eu-north-1** tab is open.
- The URL is [eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#LoadBalancers](https://eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#LoadBalancers).
- The AWS logo and search bar are at the top.
- The navigation bar includes links for Europe (Stockholm), Faisal Khan, and other account options.
- The left sidebar shows the navigation menu:
  - AMI Catalog
  - Elastic Block Store** (selected)
  - Volumes
  - Snapshots
  - Lifecycle Manager
  - Network & Security**
  - Security Groups
  - Elastic IPs
  - Placement Groups
  - Key Pairs
  - Network Interfaces
  - Load Balancing** (selected)
  - Load Balancers (selected)
  - Target Groups
  - Trust Stores
  - Auto Scaling**
  - Auto Scaling Groups
  - Settings
- The main content area displays the "Load balancers (1)" section. A message states: "Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic." Below this is a search bar labeled "Filter load balancers".

Name	DNS name	State	VPC ID	Availability Zones	Type	Date create
a45727ef784454bfb1...	a45727ef784454bfb165b...	-	vpc-0e345f19effbea822	3 Availability Zones	classic	March 31, 2024

Below the table, a message says "0 load balancers selected" and "Select a load balancer above."

The screenshot shows the AWS CloudFront console with the following details:

- Region:** Europe (Stockholm)
- Load Balancers:** eu-north-1#LoadBalancers
- Actions:** Create load balancer
- Load balancers (1/1):** a45727ef784454bfd165b988391b4de
- Details:** Load balancer type: Classic, Status: 1 of 1 instance in service, VPC: vpc-0e345f19effbea822, Date created: March 31, 2025, 19:22 (UTC+05:30), Scheme: Internet-facing, Hosted zone: 223TAZ6LKFNMNQ, Availability Zones: subnet-049ebbe2e309cd48f [eu-north-1a] (eu-north-1a)

Screenshot of the AWS Cloud Console showing the Load balancers page for the eu-north-1 region. The left sidebar shows navigation links for Clusters, EC2, Lambda, and other services. The main content area displays a table of load balancers with one entry:

Name	DNS name	State	VPC ID	Availability Zones	Type	Date create
a45727ef784454bfdb165b...	a45727ef784454bfdb165b...	-	vpc-0e345f19effbea822	3 Availability Zones	classic	March 31, 2024

The details pane for the selected load balancer shows its configuration, including its DNS name and associated subnets:

**Load balancer: a45727ef784454bfdb165b988391b4de**

subnet-048d5fb4ac1b0565e eu-north-1b (eu-n1-az2)  
subnet-03d77b3f5d3349800 eu-north-1c (eu-n1-az3)

**DNS name info**  
a45727ef784454bfdb165b988391b4de-1940812985.eu-north-1.elb.amazonaws.com (A Record)

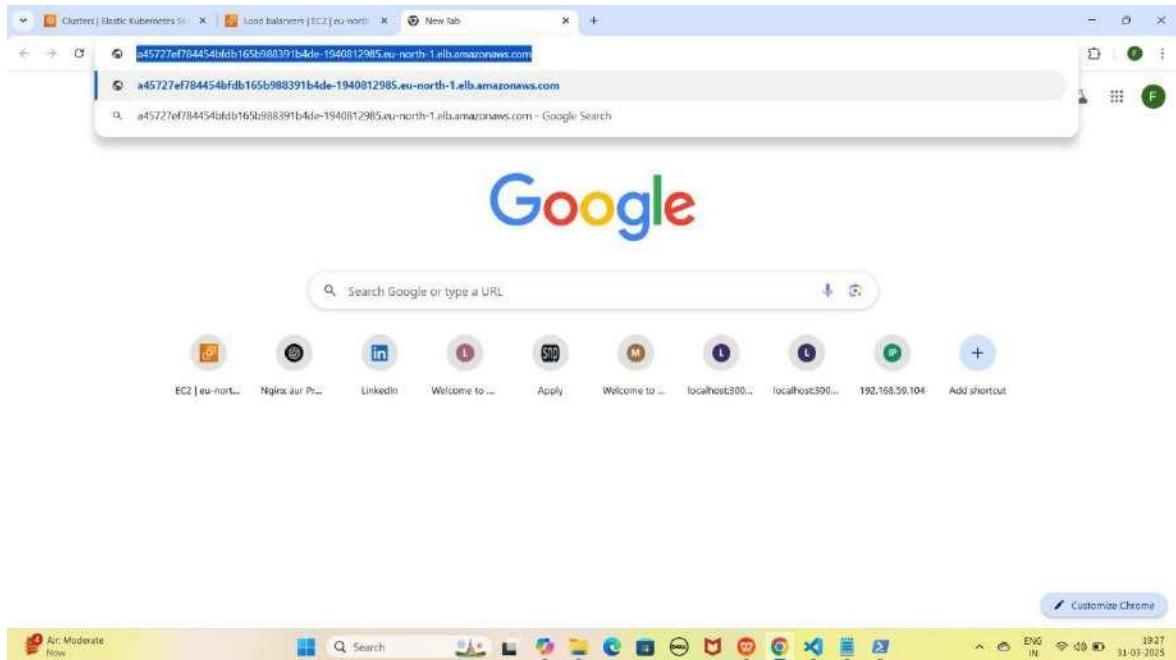
This Classic Load Balancer can be migrated to a next generation load balancer. Migration wizard uses your load balancer's current configurations to create a new load balancer. [Learn more](#) [Launch migration wizard](#)

Screenshot of the AWS Cloud Console showing the Load balancers page for the eu-north-1 region. The interface is identical to the first screenshot, but a message box is displayed indicating that the DNS name has been copied.

**DNS name copied**

a45727ef784454bfdb165b988391b4de-1940812985.eu-north-1.elb.amazonaws.com (A Record)

This Classic Load Balancer can be migrated to a next generation load balancer. Migration wizard uses your load balancer's current configurations to create a new load balancer. [Learn more](#) [Launch migration wizard](#)



# Part 3: Registering a Domain Name on Hostinger (From Sign-Up to Purchase)

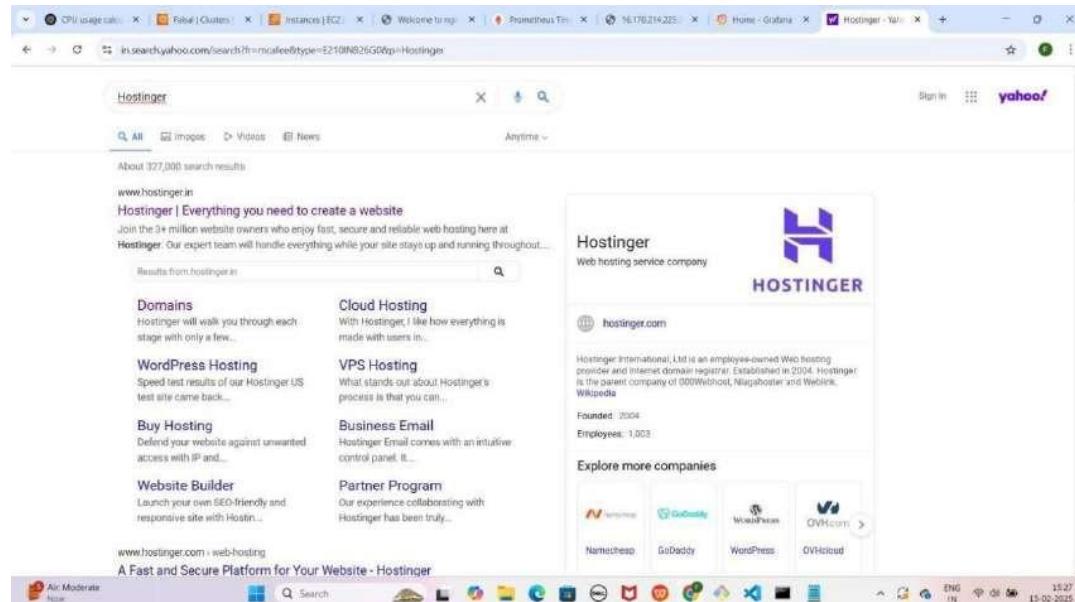
## Step 1: Domain Name Provider Choose Karo

- Tumhe **GoDaddy, AWS (Route 53), ya Hostinger** jaisa domain provider chahiye
- Main yahan **Hostinger** use kar raha hoon

## Step 2: Hostinger Pe Domain Search Karo

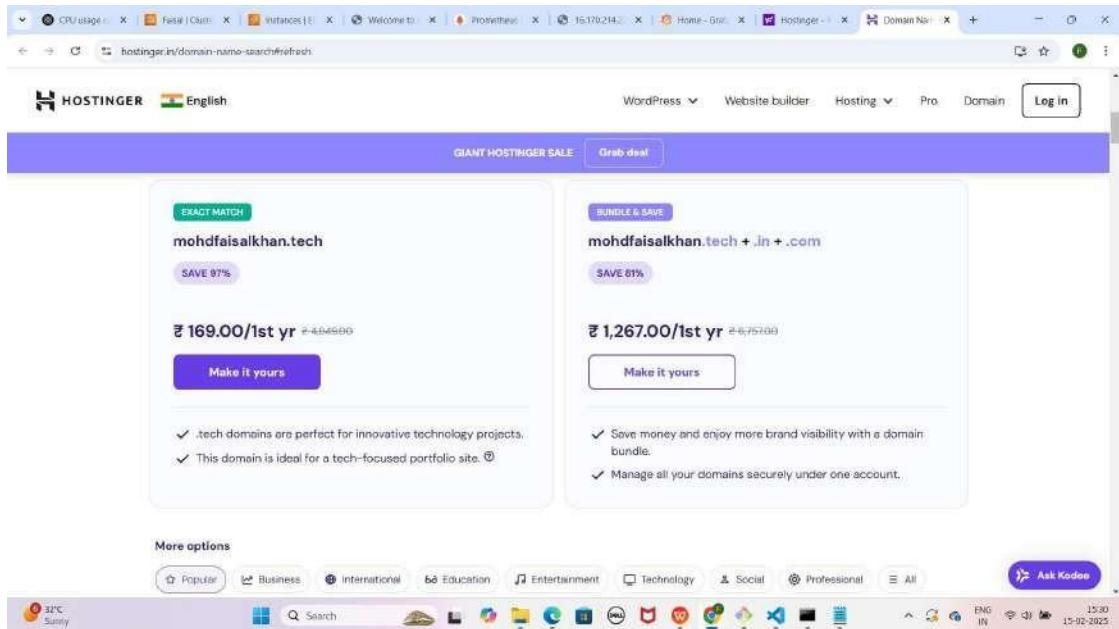
- **Google** pe jao aur "**Hostinger**" search karo
- "**Domains**" pe click karo
- Search bar me apna desired **domain** likho (Jaise: **mohdfaiskhan.tech**)
- (Aapke Name se Search Karo)
- **Search button** pe **click** karo

YE KUCH ISTARHA LAGEGA



The screenshot shows the Hostinger domain search interface. At the top, there's a purple header with the Hostinger logo, language selection (English), and navigation links for WordPress, Website builder, Hosting, Pro, Domain, and Log in. Below the header is a blue banner with the text "GIANT HOSTINGER SALE" and a "Grab deal" button. The main section has a purple background with the heading "Search and buy a domain in minutes". A sub-instruction "It's easy – simply enter your desired domain name and instantly check its availability. Register it before someone else will." is displayed. Below this is a search bar with the placeholder "Type in that perfect domain name" and a red "Search" button. To the right of the search bar are several domain extension boxes: ".in" (₹ 119.00), ".com" (₹ 499.00), ".online" (₹ 169.00), ".shop" (₹ 89.00), ".org" (₹ 699.00), and ".xyz" (₹ 179.00). A note at the bottom states "Free WHOIS privacy protection is included with every eligible domain registration." An "Ask Kodoo" link is also present.

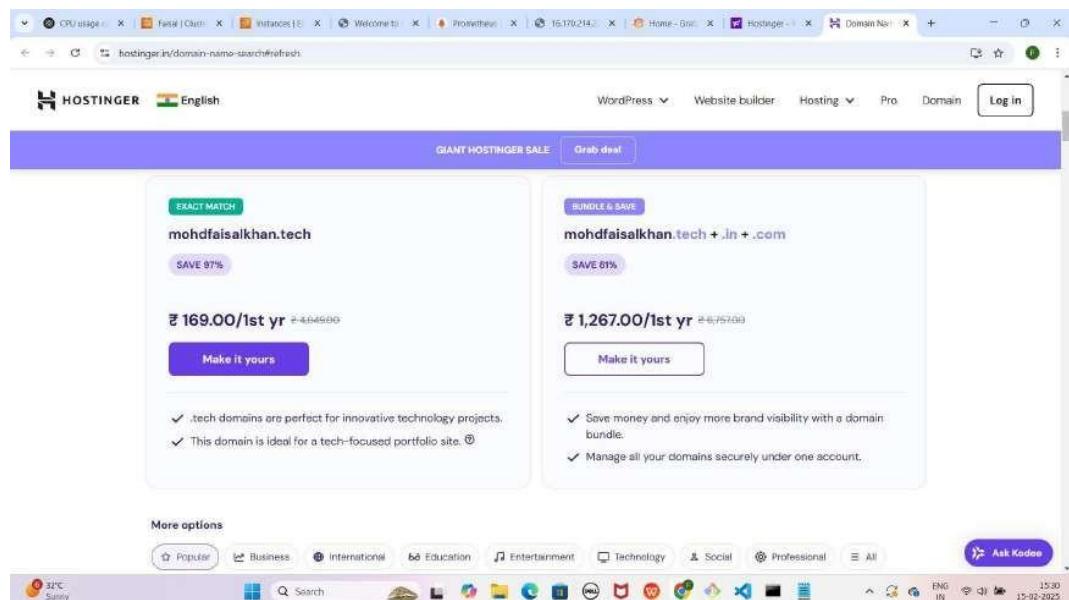
This screenshot is identical to the one above, but the search bar contains the specific domain name "mohdalsalikhan.tech". The rest of the interface, including the available domain extensions and promotional banners, remains the same.



## Step 3: Domain Select Karo

- "Make it Yours" pe click karo
- 1-year period select karo
- Continue button pe click karo

YE KUCH ISTARHA LAGEGA



Your cart

mohdfaikhan.tech  
.TECH domain registration

Period: 2 years **SAVE 48%** **£5,018.00** £9,696.00

Registration renewes at £4,849.00/year on 15/02/2027

FREE domain privacy protection included.

**Subtotal** £9,696.00 **£5,018.00**  
Subtotal does not include applicable taxes.  
Discount -48% -£4,680.00

Have a coupon code?

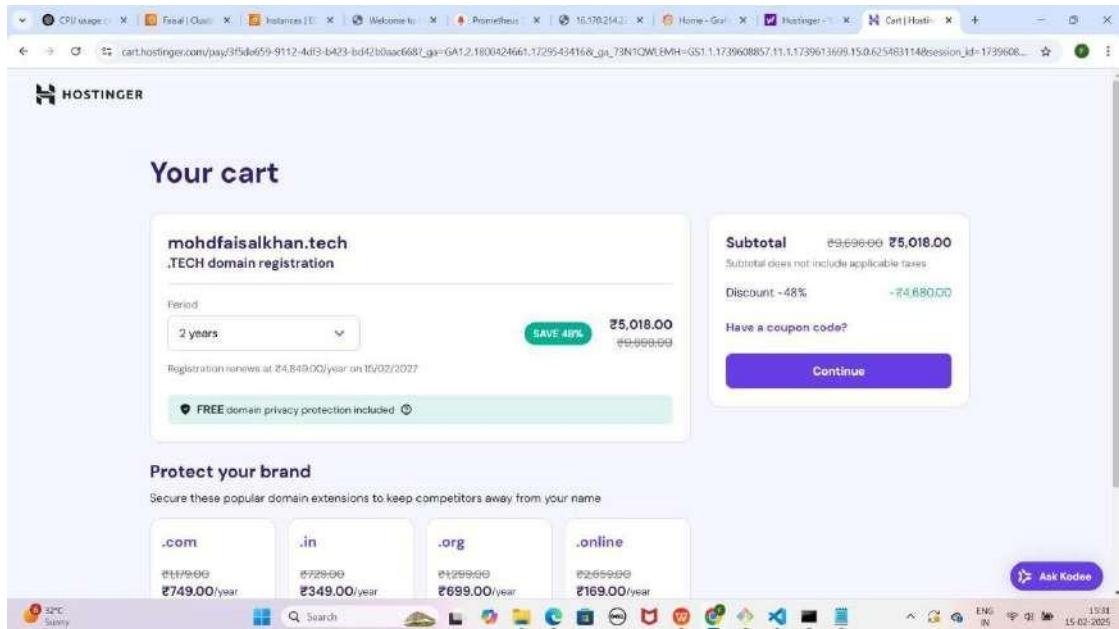
**Continue**

Protect your brand

Secure these popular domain extensions to keep competitors away from your name

.com	.in	.org	.online
£7,799.00	£7,299.00	£7,299.00	£2,699.00
<b>£749.00/year</b>	<b>£349.00/year</b>	<b>£699.00/year</b>	<b>£169.00/year</b>

Ask Kodoo



Your cart

mohdfaikhan.tech  
.TECH domain registration

Period: 1 year **SAVE 97%** **£169.00** £4,649.00

1 year 15/02/2026  
2 years Included  
3 years

**Subtotal** £4,649.00 **£169.00**  
Subtotal does not include applicable taxes.  
Discount -97% -£4,680.00

Have a coupon code?

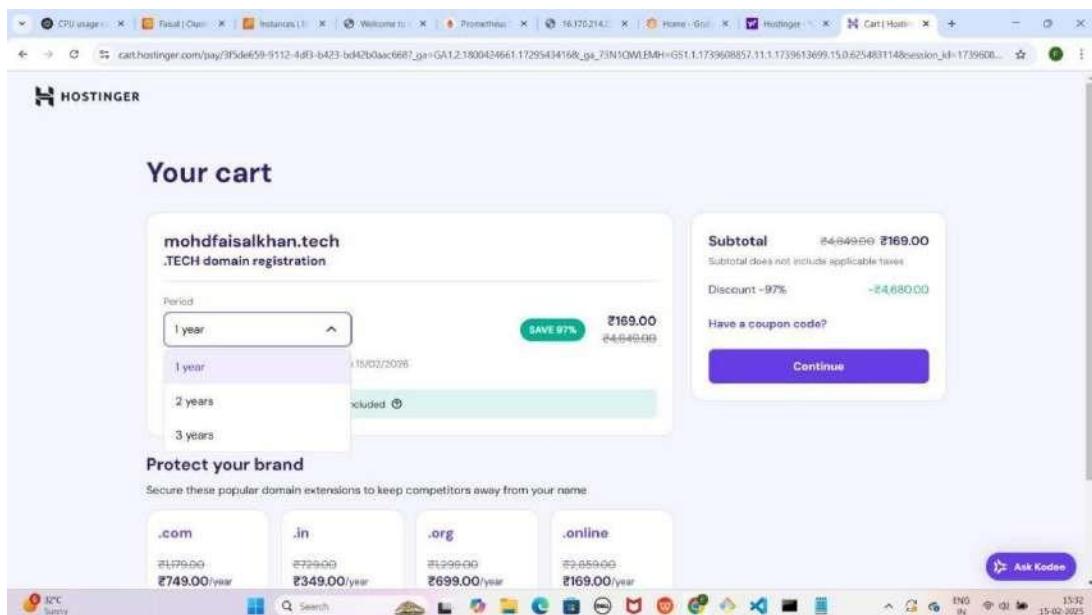
**Continue**

Protect your brand

Secure these popular domain extensions to keep competitors away from your name

.com	.in	.org	.online
£7,799.00	£7,299.00	£7,299.00	£2,699.00
<b>£749.00/year</b>	<b>£349.00/year</b>	<b>£699.00/year</b>	<b>£169.00/year</b>

Ask Kodoo



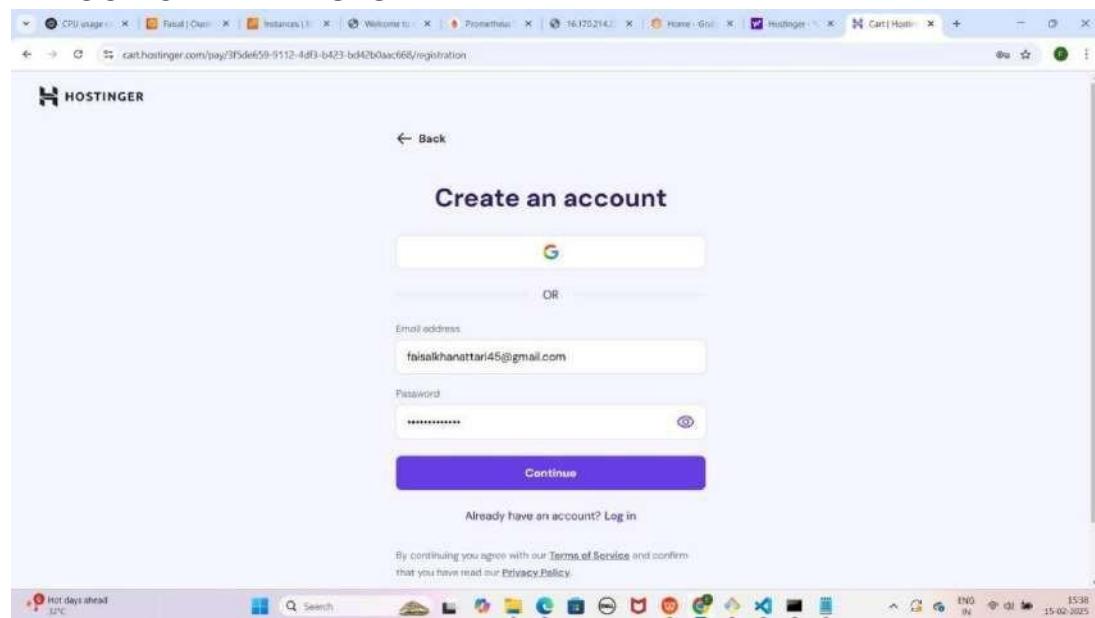
## Step 4: Account Create Karo

- **Create an Account Page** pe **Email** aur **Password** enter karo then continue pe click karo
- **Billing Address** details fill karo
- **Continue** button pe click karo

## Step 5: Payment Complete Karo Or Login Karke Domain Registration Complete Karo

- **UPI** option select karo Submit payment pe click karo or upi payment complete karo
- **Order Summary** dekho, phir **Continue** pe click karo
- **QR Code** scan karo aur **PhonePe, Paytm, ya Google Pay** se payment complete karo

YE KUCH ISTARHA LAGEGA



The screenshot shows the 'Billing address' section of the Hostinger checkout process. The form fields are populated with placeholder values: 'First name\*' contains ' ', 'Last name\*' contains ' ', 'Phone number' contains '+91 (India) 00000000', 'Country of residence\*' contains 'India', 'Address\*' contains ' ', 'City\*' contains ' ', 'State\*' contains ' ', and 'ZIP code\*' contains ' '. A checkbox for 'Add company details' is unchecked. To the right, the 'Order summary' panel shows a subtotal of ₹183.93, a discount of ₹4,680.00, and a final estimated total of ₹183.93.

The screenshot shows the same 'Billing address' section with specific values entered: 'First name\*' contains 'Faisal', 'Last name\*' contains 'Khan', 'Phone number' contains '+91 (India) 8374452927', 'Country of residence\*' contains 'India', 'Address\*' contains 'Falaknuma', 'City\*' contains 'Hyderabad', 'State\*' contains 'Telangana', and 'ZIP code\*' contains '500053'. The rest of the page, including the 'Order summary' panel, remains identical to the first screenshot.

[CPU usage](#) [Final Class](#) [Instances](#) [Welcome to](#) [Prometheus](#) [Home - Graf](#) [Hostinger](#) [Cart | Hosti](#)

carthostinger.com/pay/3f5de659-9112-4d3-b23-bd42b0aac668/checkout

**H HOSTINGER**

[Back](#)

**1 Billing address:**

First name*	Last name*
Faisal	Khan
Phone number*	
+91 (India)	8374452927
Country of residence*	India
Address*	Palaknuma
City*	Hyderabad
State*	Telangana
ZIP code*	600053

Add company details [?](#)

**Continue**

**2 Payment:**

Order summary

mohdalsikhan.tech	
Domain registration - 1 year	\$169.00
ICANN fee	\$14.83
Domain privacy protection <a href="#">?</a>	\$0.00
<b>Subtotal</b>	\$183.83
Discount -9%	-\$16.54
<b>Taxes</b> <small>(Estimated after billing information)</small>	\$0.00
<b>Ext. total</b>	\$163.93

Have a coupon code?

[Ask Kodex](#)

Finance headline India Wholesale...

Search

ENG IN 15:40 15-02-2025

[CPU usage](#) [Final Class](#) [Instances](#) [Welcome to](#) [Prometheus](#) [Home - Graf](#) [Hostinger](#) [Cart | Hosti](#)

carthostinger.com/pay/3f5de659-9112-4d3-b23-bd42b0aac668/checkout

**H HOSTINGER**

[Back](#)

**1 Billing address:**

First Name	•91 9374452927
India, Tel. Hyderabad	Palaknuma, 600053

**2 Payment:**

Payment method:

Card:

**Submit payment**

B. Discreet and secure payment

By checking out you agree with our [Terms of Service](#) and confirm that you have read our [Privacy Policy](#). You can cancel recurring payments at any time.

UPI

PayTM

Net Banking

PayPal

Order summary

mohdalsikhan.tech	
Domain registration - 1 year	\$169.00
ICANN fee	\$14.83
Domain privacy protection <a href="#">?</a>	\$0.00
<b>Subtotal</b>	\$183.83
Discount -9%	-\$16.54
<b>Taxes</b> <small>(Estimated after billing information)</small>	\$0.00
<b>Total</b>	\$163.93

Have a coupon code?

[Ask Kodex](#)

Finance headline India Wholesale...

Search

ENG IN 15:41 15-02-2025

The screenshot shows the Hostinger checkout process. On the left, there's a 'Billing address' section with a placeholder for a house number and street name. Below it is a 'Payment' section with a dropdown menu set to 'UPI'. Underneath are several other payment method options: PayTM, Net Banking, and PayPal. To the right is an 'Order summary' box containing the following details:

	Amount
Domain registration - 1 year	₹169.00 ₹169.00
ICANN fee	₹14.93 ₹14.93
Domain privacy protection	₹0.00 ₹0.00
<b>Subtotal</b>	₹169.00 ₹169.00
Discount - 8%	-₹13.52 ₹0.00
Taxes (0)	₹0.00 ₹0.00
<b>Total</b>	<b>₹217.04 ₹217.04</b>

At the bottom of the summary, there's a link 'Have a coupon code?' and a note about accepting and secure payments.

This screenshot shows the payment confirmation page. It features a 'Continue payment via UPI' form with fields for 'Phone' (8374452927) and 'UPI ID' (Enter your UPI ID). Below these fields is a note: 'UPI ID should be in the format user.name@yblpayid'. A 'Continue' button is centered below the input fields. To the right is an 'Order Summary' table:

	Amount
Request ID: rd_26018905	
TECH Domain (billed every year) - mohdaislahi.tech	₹169.00 ₹169.00
Domain WHOIS Privacy Protection	₹0.00 ₹0.00
ICANN fee (billed every year)	₹14.93 ₹14.93
GST 16%	₹33.11 ₹33.11
Credits	₹0.00 ₹0.00
<b>Total</b>	<b>₹ 217.04 ₹217.04</b>

**Note: Aapko neeche wala pop-up milega, tension mat lo, aap bas "Auto Pay" kar do, lekin baad mein aap isey PhonePe ke settings se cancel kar sakte ho. Abhi ke liye "Continue" pe click karo.**

YE KUCH ISTARHA LAGEGA

---

This is a UPI AutoPay payment

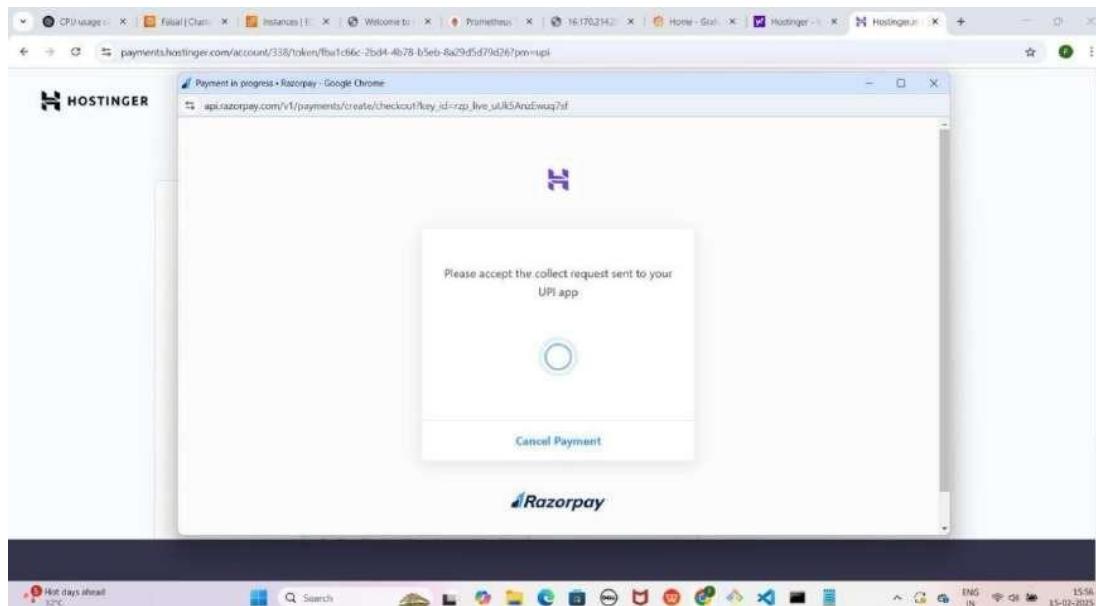
---

You'll be charged ₹ 217.04 today.

When you go to approve your payment, you will see the transaction limit of ₹ 7,500.00. **You won't be charged this amount.** The limit is set to automate your future payments (if any) and set up UPI AutoPay. Cancel AutoPay anytime.

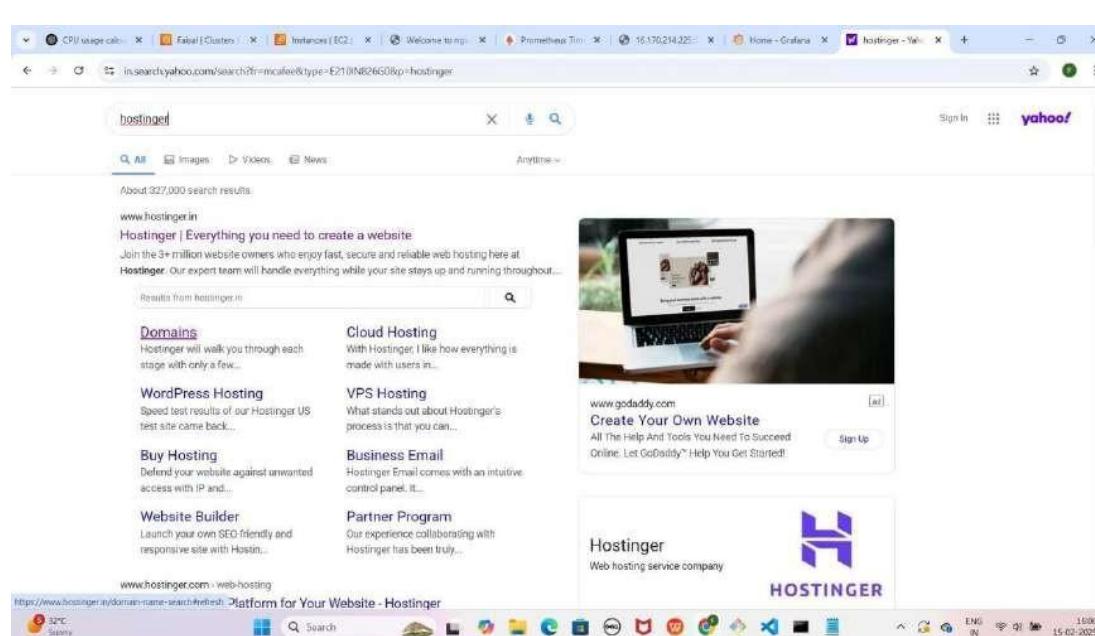
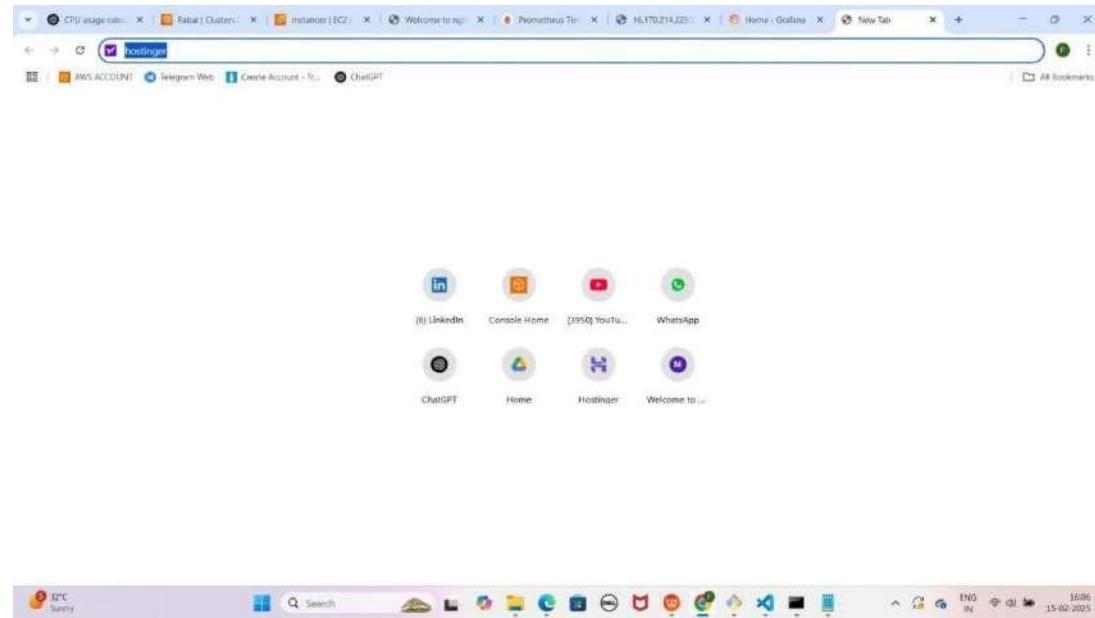
---

[Continue](#)



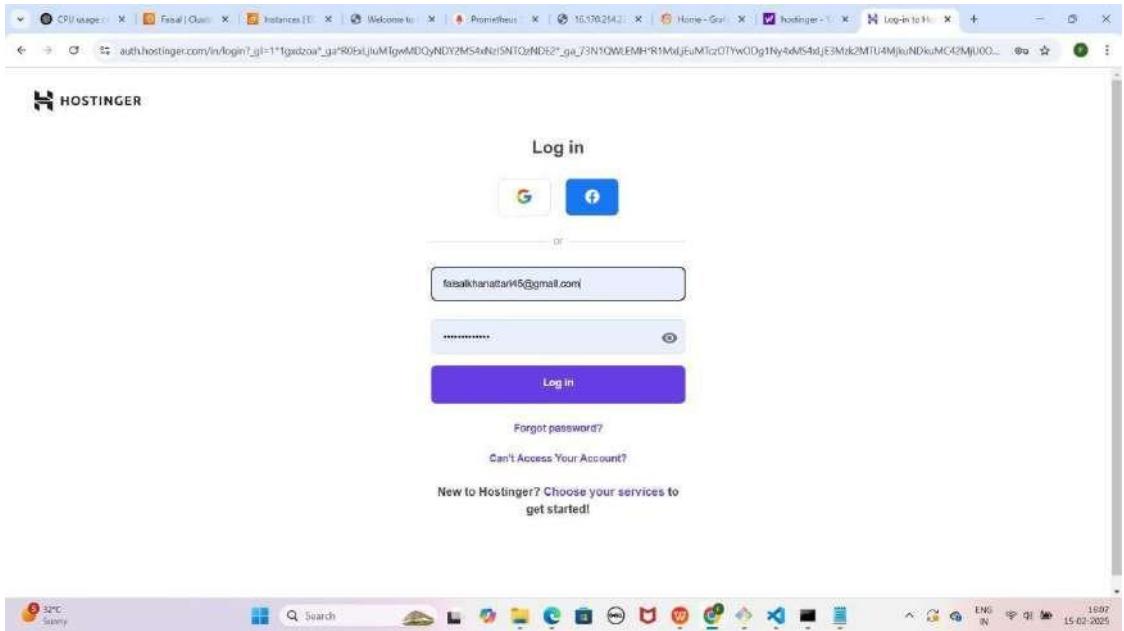
- **PhonePe** open karo aur **payment** complete karo
- **Browser** me new tab open karo aur **Google** me **Hostinger** search karo
- **Hostinger website** pe jao aur "**Domains**" pe **click** karo
- **Top right corner** pe "**Log in**" pe **click** karo
- **Email** aur **password** enter karo jo tumne pehle set kiya tha

## YE KUCH ISTARHA LAGEGA



The screenshot shows the Hostinger domain search interface. At the top, there's a navigation bar with links for CPU usage, Failai (Cloud), Instances, Welcome to, Prometheus, Home - Graf, Hostinger, and Domain Name. Below the navigation is a purple header with the Hostinger logo, a language switcher (English), and a 'Log in' button. The main heading is 'Search and buy a domain in minutes'. A sub-instruction reads: 'It's easy – simply enter your desired domain name and instantly check its availability. Register it before someone else will.' Below this are two buttons: 'Find new domain' and 'Generate domain using AI'. A search input field with placeholder text 'Type in that perfect domain name' is followed by a red 'Search' button. At the bottom of the search area, there are six boxes showing domain extensions and their prices: '.in' (₹ 799.00 / ₹ 119.00), '.com' (₹ 1799.00 / ₹ 499.00), '.online' (₹ 2059.00 / ₹ 169.00), '.shop' (₹ 2859.00 / ₹ 89.00), '.org' (₹ 1999.00 / ₹ 699.00), and '.xyz' (₹ 1299.00 / ₹ 179.00). A note states: 'Free WHOIS privacy protection is included with every eligible domain registration.' The status bar at the bottom shows the URL 'https://auth.hostinger.com/en/login', the Windows taskbar with various pinned icons, and system information like battery level (32°C, Sunny) and date/time (15-02-2023).

The screenshot shows the Hostinger login interface. The top navigation bar is identical to the previous screenshot. The main title is 'Log in'. It features social media login buttons for Google and Facebook. Below these are fields for 'Email' and 'Password', each with a 'Forgot password?' link. A large blue 'Log in' button is centered. Below the buttons are links for 'Forgot password?' and 'Can't Access Your Account?'. A promotional message at the bottom encourages users to 'New to Hostinger? Choose your services to get started!'. The status bar at the bottom shows the URL 'https://auth.hostinger.com/en/login?\_gl=1\*t1gxdzca\*\_ga\*80ExJiuMigwMDQyNDY2MS4vNzI5NTQzNDE2\*\_jsu/\_j3N1QWLEMH\*R1MeLjsuM1czOYwOdg1Ny4xMS4xLjE3M62MT4Mjk0NDIsaMCA2MjUOO...', the Windows taskbar with pinned icons, and system information like battery level (32°C, Sunny) and date/time (15-02-2023).



- **Login** karne ke baad tumhe "**Register your domain mohdfaiskhan.tech**" ka option milega
- "**Next Step**" pe click karo
- "**Finish Registration**" pe click karo
- Tumhara **domain registration complete ho jayega**

YE KUCH ISTARHA LAGEGA

Register your domain mohdfaikhan.tech

1 Select primary information

Country: India  
Personal (radio button selected)  
Company or organization

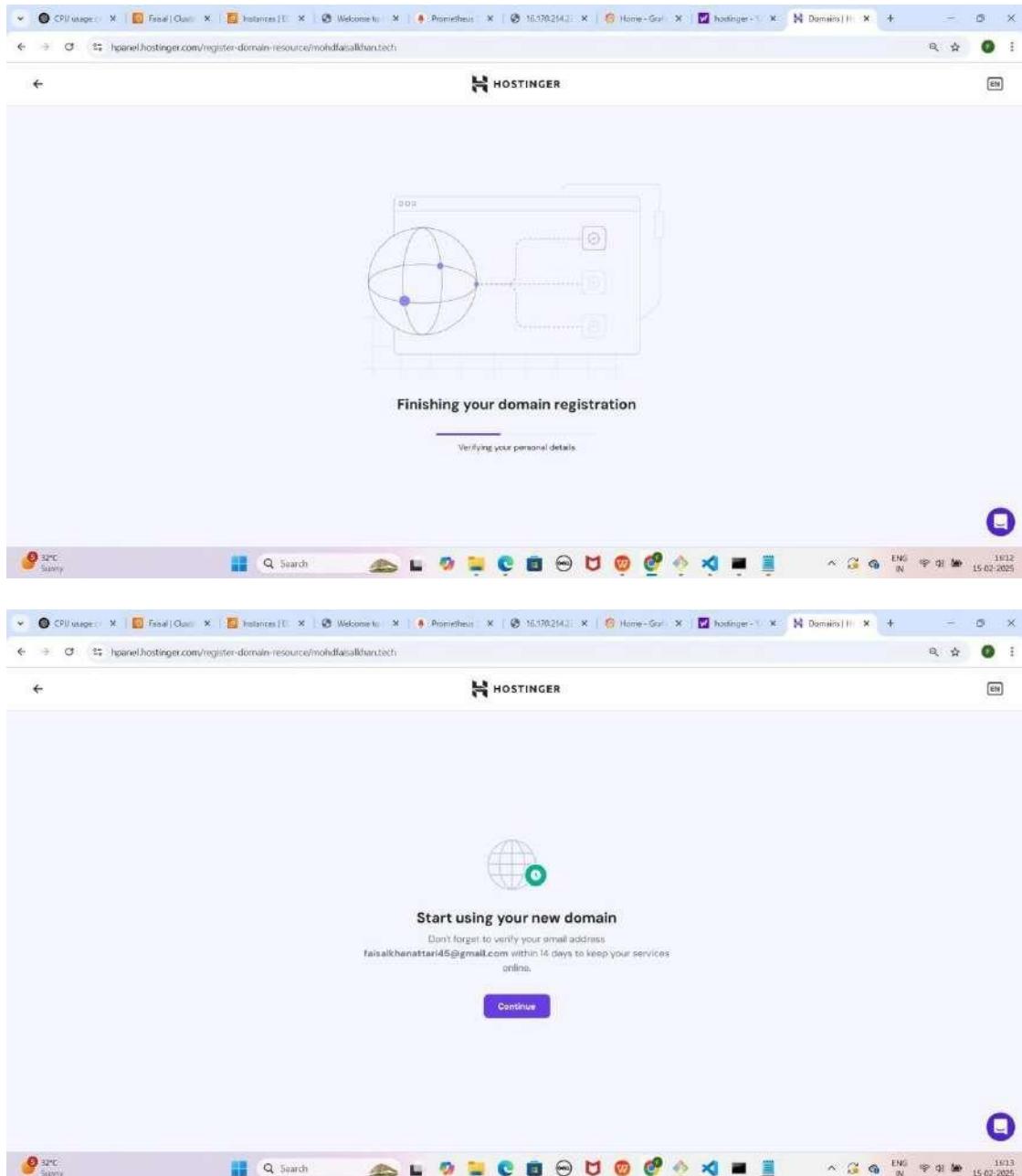
Next step

2 Enter contact details

First name: Faiz  
Last name: Khan  
Email: faizkhanmtech@gmail.com  
The email address will be used to verify your domain ownership.  
Reply-to Email Address: faizkhanmtech@gmail.com  
City: Hyderabad  
Address: Plot no 5000000  
ZIP code: 500000  
Country/Code of the Post Office:  
Mobile: +91 9876543210  
Phone number: +91 9876543210

Your domain will be registered using these details. Your email and phone number may be used to verify ownership. Providing incorrect details may lead to domain suspension.

Finish Registration



**NOTE: Continue pe click karo**

## YE KUCH ISTARHA LAGEGA

The screenshot shows a Windows desktop environment. A browser window is open to the Hostinger website, displaying a promotional offer for web hosting. The offer includes a 69.00 USD plan with various features like a No-Code Website builder, Malware Scanner, and Managed WordPress. The desktop taskbar at the bottom shows several pinned icons, including FileZilla, 32PC, and a search bar. The system tray indicates the date as 15.02.2025.

Everything You Need to Create a Website

Web Hosting

€ 69.00 /mo

No-Code Website builder  
Malware Scanner  
1 Email Account  
Free Automatic Website Migration  
Unlimited Free SSL

Unmetered traffic  
Weekly Backups  
50 GB SSD Storage  
Managed WordPress  
24/7 Customer Support

Explore plans

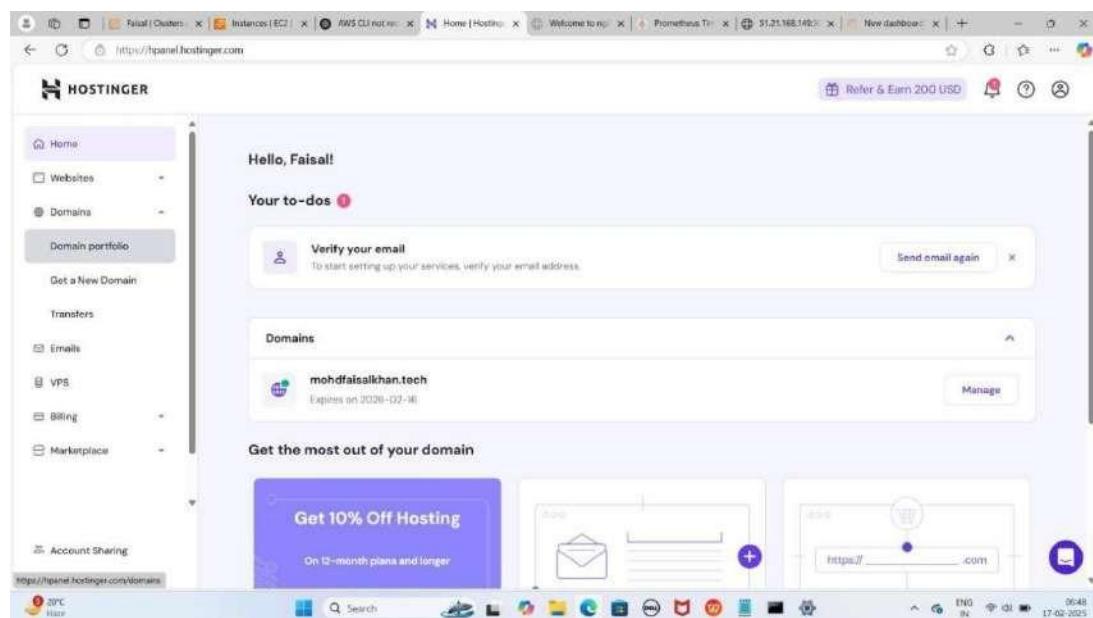
Skip, I don't need a website

FileZilla 32PC Search 15.02.2025

## Step 6: Hostinger Me Domains Section Me Navigate Karo

- **Hostinger** mein Dashboard page dikhega
- Phir **Domains** pe click karo.
- **Domain Portfolio** mein apni domain **mohdfaiskhan.tech** ko select karo.

YE KUCH ISTARHA LAGEGA



The screenshot shows the 'Domain portfolio' section of the Hostinger control panel. On the left sidebar, 'Domain portfolio' is selected. In the main area, a domain 'mohdfaikhan.com' is highlighted with a purple box. A search bar is present above a table listing domains. The table includes columns for 'Domain name', 'Status', 'Expiration date', and 'Auto-renewal'. One row in the table shows 'mohdfaikhan.tech' as active, expiring on 2026-02-16, with auto-renewal turned off.

## Step 7: Hostinger Me DNS/Nameservers Section Me Navigate Karo

- Apni **domain** ko select karne ke baad, **DNS/Nameservers** pe click karo.
- Yahan ek **page** khul jayega jahan tumhe **DNS records add** karna hoga.

YE KUCH ISTARHA LAGEGA

The screenshot shows the 'Domain Overview' page for 'mohdfaikhan.tech'. The 'DNS / Nameservers' tab is selected in the sidebar. It displays two nameservers: 'ns1.dns-parking.com' and 'ns2.dns-parking.com'. Below this, there's a 'Contact Information' section with fields for 'First Name' and 'Last Name', both set to 'Faisal'. A 'Your domain checklist!' section at the bottom indicates '1/3 completed'.

The screenshot shows the Hostinger DNS Management interface. On the left, there's a sidebar with a main menu containing "Domain Overview", "DNS / Nameservers" (which is selected and highlighted in purple), and "Domain Ownership". The main content area has tabs for "DNS records", "Child nameservers", "DNSSEC", "Forwarding", and "DNS history".

**Nameservers:** This section explains what nameservers are and how to change them. It lists two nameservers: "ns1.dns-parking.com" and "ns2.dns-parking.com". There is a "Change Nameservers" button.

**Manage DNS records:** This section allows users to define how their domain behaves. It includes fields for "Type" (set to "A"), "Name" (with a dropdown arrow), "Points to" (IP address field with value "14400"), and "TTL" (Time-to-Live). A "Add Record" button is also present. Below this, there's a search bar and a table header with columns: Type, Name, Priority, Content, and TTL.

## **Part 4: Configuring DNS on Hostinger for Load Balancer**

### **Step 1: CNAME Record Add Karna (Load Balancer Se Connect Karne Ke Liye)**

1. Pehle Load Balancer ka DNS Name Copy Karo.
2. Naya CNAME Record Add Karo:

- **Type:** CNAME
- **Name:** @
- **Target:** Tumhara Load Balancer ka DNS Name
- **TTL:** 14400

3. Add Record pe Click Karo.

**NOTE : Add Record pe Click karne ke baad DNS Record created successfully  
karke pop-up aayega**

### **Step 2: CNAME Record Add Karna (WWW ke liye)**

1. Pehle Load Balancer ka DNS Name Copy Karo.
2. Naya CNAME Record Add Karo:

- **Type:** CNAME
- **Name:** www
- **Target:** Tumhara Load Balancer ka DNS Name
- **TTL:** 14400

3. Add Record pe Click Karo.

**NOTE : Add Record pe Click karne ke baad DNS Record created successfully  
karke pop-up aayega**

## YE KUCH ISTRAHA LAGEGA

The screenshot shows the AWS CloudWatch Metrics interface. A line chart displays CPU usage percentage over a period of 1 hour. The Y-axis ranges from 0% to 100%, and the X-axis shows time intervals. The CPU usage fluctuates between 0% and 100% throughout the hour.

The screenshot shows the AWS CloudWatch Metrics interface. A line chart displays CPU usage percentage over a period of 1 hour. The Y-axis ranges from 0% to 100%, and the X-axis shows time intervals. The CPU usage fluctuates between 0% and 100% throughout the hour.

The screenshot shows the Hostinger DNS Management interface. On the left, a sidebar includes 'Main menu', 'Domain Overview', 'DNS / Nameservers' (which is selected and highlighted in purple), and 'Domain Ownership'. The main content area displays a domain overview for 'ns2.dns-parking.com'. A success message 'DNS Record created successfully.' is visible. Below it, a form for creating a new DNS record is shown, with 'Type' set to 'A', 'Name' set to '@', and 'Points to' set to 'a4572ef704454bfbdb65b98839lb4de-1940812985.eu-north-1.amazonaws.com'. The TTL is set to 14400. An 'Add Record' button is present. A table below lists existing records: one ALIAS record pointing to the same Amazon URL with TTL 14400, and one CAA record with the same details. The Windows taskbar at the bottom shows various pinned icons and the date 31-03-2025.

This screenshot is nearly identical to the first one, showing the Hostinger DNS Management interface for 'ns2.dns-parking.com'. The 'DNS / Nameservers' tab is still selected. A success message 'DNS Record created successfully.' is displayed. In the 'Manage DNS records' section, a new CNAME record is being added with 'Name' set to 'www' and 'Target' set to 'a4572ef704454bfbdb65b98839lb4de-1940812985.eu-north-1.amazonaws.com'. The TTL is 14400. The table of existing records remains the same. The Windows taskbar at the bottom shows the date 31-03-2025.

The screenshot shows the Hostinger HPanel interface. On the left, there's a sidebar with 'Main menu', 'Domain Overview', 'DNS / Nameservers' (which is selected and highlighted in purple), and 'Domain Ownership'. The main content area has a header 'Manage DNS records' with a sub-instruction: 'These records define how your domain behaves. Common uses include pointing your domain at web servers or configuring email delivery for your domain.' Below this is a search bar and a table for managing DNS records. The table has columns: Type, Name, Priority, Content, TTL, Delete, and Edit. There are two CNAME records listed: one for 'www' pointing to 'a45727e784454bfdb65b98839fb4de-1940812985.eu-north-1e.lb.amazonnews.com' and another for '@' pointing to the same URL. Both have a TTL of 14400. A success message box is visible at the top right: 'DNS Record created successfully'.

This screenshot is from the same Hostinger HPanel session as the first one, but it shows the records after they have been updated. The table now includes two additional CAA records: one for '@' with the content '0 issuewild "comodoca.com"' and another for '@' with the content '0 issuewild "sectigo.com"'. All other fields remain the same: Type (CNAME), Name (www and @), Priority (0), Content (the URLs), TTL (14400), and the 'Delete' and 'Edit' buttons.

**NOTE: Jab tum Load Balancer ke liye records add kar loge, toh kuch der baad apni application ko in URLs par access kar paoge. Ye process 15-20 minutes ya usse zyada bhi le sakta hai**

## Step 3: Records Add Karne Ke Baad Website Ko Domain Name Se Access Karna

### 1. Apni NGINX application ko Domain Name se access karo

- Nginx: <http://mohdfaiskhan.tech>

YE KUCH ISTARHA LAGEGA



### Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org). Commercial support is available at [nginx.com](http://nginx.com).

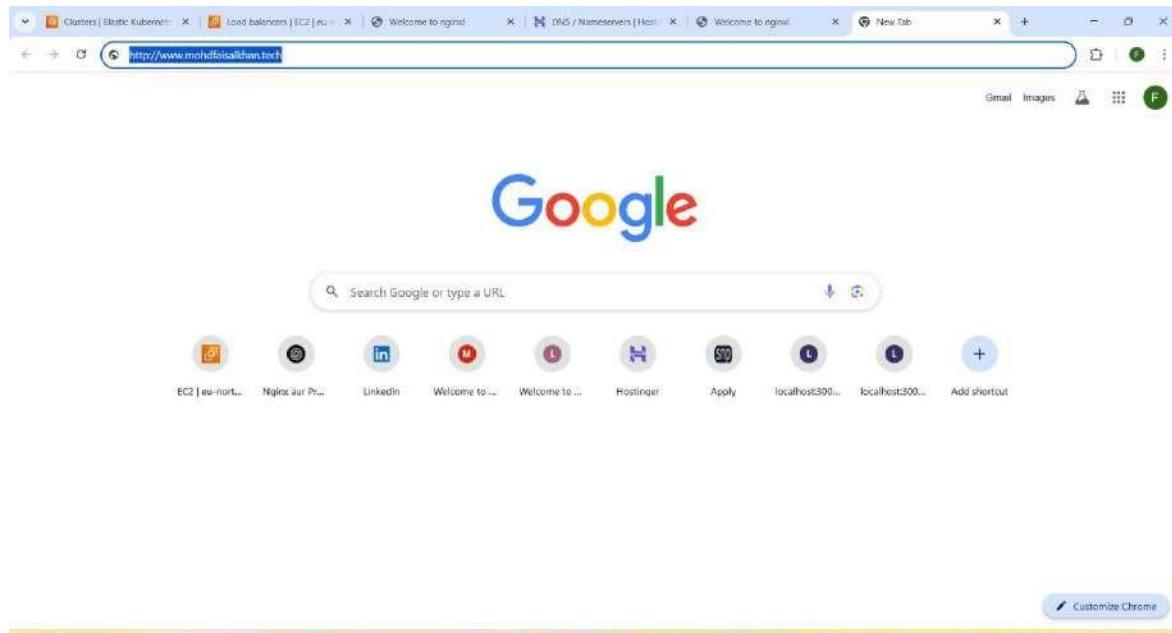
Thank you for using nginx.



## 2. Apni NGINX application ko www wale Domain Name se access karo

- Nginx: <http://www.mohdfaiskhan.tech>

YE KUCH ISTARHA LAGEGA



## Part 5: Enabling HTTPS on Load Balancer Using ACM

**ACM: AWS ACM** (AWS Certificate Manager) ek **AWS service** hai jo tumhari **website** ke liye **SSL/TLS certificates manage** karti hai. Yeh automatically **certificates issue, renew, aur manage** karta hai taake tumhari **website HTTPS** par securely chale

### Step 1: CAA Records Add Karo (SSL Certificate ke liye)

#### 1. Naya Record Add Karo:

- **Type:** CAA
- **Name:** @
- **Flag:** 0
- **Tag:** issue
- **CA domain:** amazon.com
- **TTL:** 14400

#### 2. Add Record pe Click Karo.

**NOTE : Add Record pe Click karne ke baad DNS Record created successfully karke pop-up aayega**

#### 1. Phir ek aur CAA Record Add Karo:

- **Type:** CAA
- **Name:** @
- **Flag:** 0
- **Tag:** issuewild
- **CA domain:** amazon.com
- **TTL:** 14400

#### 2. Add Record pe Click Karo.

**NOTE : Add Record pe Click karne ke baad DNS Record created successfully karke pop-up aayega**

## YE KUCH ISTARHALAGEGA

The screenshot shows the Hostinger DNS/Nameservers management interface. On the left sidebar, 'DNS / Nameservers' is selected. In the main area, under 'Nameservers', two entries are listed: 'ns1.dns-parking.com' and 'ns2.dns-parking.com'. Below this is a 'Change Nameservers' button. Under 'Manage DNS records', there's a form for creating a new record. The 'Type' dropdown is set to 'CAA', 'Name' is '@', 'Flag' is '0', 'Tag' is 'issue', and 'CA domain' is 'HINNAZOR.COM'. The 'TTL' field contains '14400'. A blue 'Add Record' button is visible. At the bottom of the interface, there's a search bar and a toolbar with various icons.

This screenshot is nearly identical to the one above, showing the same Hostinger DNS/Nameservers interface. The 'CAA' record has been successfully created, as indicated by a green success message 'DNS Record created successfully' at the top right of the 'Manage DNS records' section. The rest of the interface, including the nameservers list and the A record creation form, remains the same.

The screenshot shows the Hostinger DNS Management interface. On the left sidebar, under 'Main menu', 'DNS / Nameservers' is selected. The main content area has tabs for 'DNS records', 'Child nameservers', 'DNSSEC', 'Forwarding', and 'DNS history'. The 'Nameservers' section displays two entries: 'ns1.dns-parking.com' and 'ns2.dns-parking.com'. A purple 'Change Nameservers' button is located below these entries. Below this, the 'Manage DNS records' section is visible, featuring a form to add a new record. The form includes fields for 'Type' (set to 'CAA'), 'Name' (@), 'Flag' (0), 'Tag' (issuewild), 'CA domain' (set to 'HOSTINGER.COM'), and 'TTL' (set to 14400). A purple 'Add Record' button is at the bottom right of the form. A search bar labeled 'Search' is also present. The status bar at the bottom shows system information: 33°C, Party cloudy, ENG IN, 31-03-2025, 19:54.

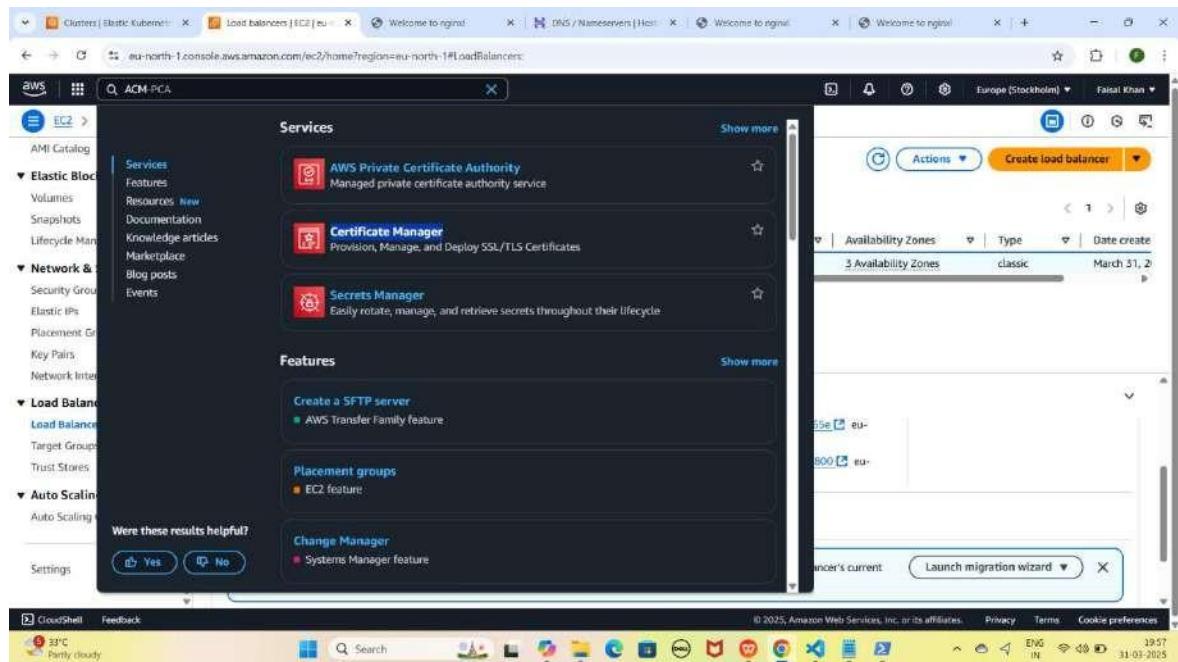
This screenshot is identical to the one above, but it includes a prominent success message in the top right corner: 'DNS Record created successfully' with a green checkmark icon. The rest of the interface and status bar are the same as the first screenshot.

**NOTE : Jab tum Amazon ke records add kar loge aur SSL certificate verify ho jayega, toh yeh records tumhari application ko HTTPS par access karne me madad karenge. Yeh process 15-20 minutes ya usse zyada bhi le sakta hai.**

## Step 2: ACM se SSL Certificate Request Karo

1. AWS Console me jao aur search bar me "ACM" likho.
2. "Certificate Manager" pe right click karo or open link in new tab se open karo .
3. "Request a certificate" pe click karo.
4. "Request a public certificate" select karo, phir "Next" pe click karo.
5. Apna domain name enter karo (e.g., [mohdfaiskhan.tech](https://mohdfaiskhan.tech)).
6. www ke liye "Add another name" pe click karo aur [www.mohdfaiskhan.tech](https://www.mohdfaiskhan.tech) add karo.
7. "Request" pe click karo.
8. Thodi der wait karo, ACM ek CNAME record dega.

YE KUCH IS TARHA LAGEGA



The screenshot shows the AWS EC2 service catalog. A context menu is open over the 'Certificate Manager' item in the 'Features' section. The menu options include:

- Open link in new tab
- Open link in new window
- Open link in incognito window
- Save link...
- Copy link address
- Copy (Ctrl+C)
- Copy link to highlight
- Search Google for 'Certificate Manager'
- Print...
- Translate selection to English
- Inspect...

The main catalog view shows the following services and features:

- Services**: AWS Private Certificate Authority, Certificate Manager, Secrets Manager.
- Features**: Create a SFTP server, AWS Transfer Family, Placement groups (EC2 feature).
- Change Manager**: Systems Manager feature.

A sidebar on the left lists categories like AMI Catalog, Elastic Block Store, Network & Content Delivery, Load Balancing, and Auto Scaling.

The screenshot shows the AWS Certificate Manager (ACM) service. The 'Certificates' page displays a table with the following columns: Certificate ID, Domain name, Type, and Status. The status is 'In use'. The table shows the message: "There are no certificates in your account.".

The left sidebar shows the following options:

- AWS Certificate Manager (ACM)**
- List certificates
- Request certificate
- Import certificate
- AWS Private CA

The browser status bar indicates the date as 31-03-2025.

The screenshot shows the 'Request certificate' page in the AWS Certificate Manager. In the 'Certificate type' section, the 'Request a public certificate' option is selected. A note below states: 'Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit AWS Private Certificate Authority.' There are 'Cancel' and 'Next Step' buttons at the bottom.

The screenshot shows the 'Request public certificate' page. In the 'Domain names' section, a fully qualified domain name 'mohdfaikhan.tech' is listed. Below it, there's a link to 'Add another name to this certificate'. In the 'Validation method' section, 'DNS validation - recommended' is selected. In the 'Key algorithm' section, 'RSA 2048' is selected. The browser status bar at the bottom indicates 'CloudShell Feedback' and the date '31-03-2025'.

**Domain names**  
Provide one or more domain names for your certificate.

**Fully qualified domain name** [Info](#)

mohdfaikhan.tech [Remove](#)

Add another name to this certificate [Info](#)

You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

**Validation method** [Info](#)  
Select a method for validating domain ownership.

DNS validation – recommended  
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

Email validation  
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

**Key algorithm** [Info](#)  
Select an encryption algorithm. Some algorithms may not be supported by all AWS services.

**Domain names**  
Provide one or more domain names for your certificate.

**Fully qualified domain name** [Info](#)

mohdfaikhan.tech [Remove](#)

www.mohdfaikhan.tech [Remove](#)

Add another name to this certificate [Info](#)

You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

**Validation method** [Info](#)  
Select a method for validating domain ownership.

DNS validation - recommended  
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

Email validation  
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

**Key algorithm** [Info](#)  
Select an encryption algorithm. Some algorithms may not be supported by all AWS services.

Screenshot of the AWS Certificate Manager console showing the 'Request public certificate' wizard.

**Domain names**  
Provide one or more domain names for your certificate.  
**Fully qualified domain name:**    
   
**Add another name to this certificate**  
You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

**Validation method** Info  
Select a method for validating domain ownership.  
 **DNS validation - recommended**  
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.  
 **Email validation**  
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

**Key algorithm** Info  
Select an encryption algorithm. Some algorithms may not be supported by all AWS services.  
 **RSA 2048**  
RSA is the most widely used key type.  
 **ECDSA P 256**  
Equivalent in cryptographic strength to RSA 3072.  
 **ECDSA P 384**  
Equivalent in cryptographic strength to RSA 7680.

**Tags** Info  
No tags associated with the resource.  
**Add new tag**  
You can add up to 50 tags.

**Cancel** **Previous** **Request**

Screenshot of the AWS Certificate Manager (ACM) console showing a successfully requested certificate.

**Certificate status:**

Identifier	Status
cf1961af-d633-4d11-9c2a-c7af2bac518b	Pending validation <a href="#">Info</a>

**Domains (2):**

Domain	Status	Renewal status	Type	CNAME name
mohdfaikhan.tech	Pending validation	-	CNAME	_298b1665b91981e95534639504ae29f4.mohdfaikhan.tech

**Details:**

In use	Serial number	Requested at	Renewal eligibility
No	N/A	March 31, 2025, 20:03:01 (UTC+05:30)	Ineligible
Domain name	Public key info	Issued at	
mohdfaikhan.tech	RSA 2048	N/A	

Screenshot of the AWS Certificate Manager (ACM) console showing a successfully requested certificate.

**Certificate status:**

Identifier	Status
c7af2bac518b	Pending validation

**Domains (2):**

Domain	Status	Renewal status	Type	CNAME name
mohdfaikhan.tech	Pending validation	-	CNAME	_298b1665b91981e95534639504ae29f4.mohdfaikhan.tech
www.mohdfaikhan.tech	Pending validation	-	CNAME	_875a9cb08d1d1902ba1e3d61741ba9cd.www.mohdfaikhan.tech

**Details:**

In use	Serial number	Requested at	Renewal eligibility
No	N/A	March 31, 2025, 20:03:01 (UTC+05:30)	Ineligible
Domain name	Public key info	Issued at	
mohdfaikhan.tech	RSA 2048	N/A	

### **Step 3: ACM CNAME Records Hostinger Me Add Karo**

#### **1. CNAME Record Add Karo (Root Domain - [mohdfaiskhan.tech](http://mohdfaiskhan.tech) ke liye)**

- **Type:** CNAME
- **Name:** ACM ka diya gaya full CNAME name generate hua Root Domain ke liye (**domain part hata ke**)
- **Target:** ACM ka diya gaya Root Domain ka full CNAME value
- **TTL:** 14400
- **Save karo.**

**NOTE : Add Record pe Click karne ke baad DNS Record created successfully  
karke pop-up aayega**

#### **2. CNAME Record Add Karo (WWW Domain - [www.mohdfaiskhan.tech](http://www.mohdfaiskhan.tech) ke liye)**

- **Type:** CNAME
- **Name:** ACM ka diya gaya full CNAME name generate hua www Domain ke liye (**domain part hata ke**)
- **Target:** ACM ka diya gaya www domain ka full CNAME value
- **TTL:** 14400
- **Save karo.**

**NOTE : Add Record pe Click karne ke baad DNS Record created successfully  
karke pop-up aayega**

## YE KUCH ISTARHA LAGEGA

The screenshot shows the AWS Certificate Manager interface. A certificate named 'c7af2bac518b' is selected, which is 'Amazon Issued'. Under the 'Domains' section, there are two entries, both marked as 'Pending validation':

Status	Renewal status	Type	CNAME value
Pending validation	-	CNAME	.298b1665b91981e95534639504ee28f4.mohdfaikhan.tech
Pending validation	-	CNAME	.875a9cb08d1d1902ba1e3d51741ba9cd.www.mohdfaikhan.tech.

In the bottom right corner of the table, there is a button labeled 'Copied to clipboard' with a clipboard icon.

The screenshot shows the Hostinger DNS management interface. On the left sidebar, 'DNS / Nameservers' is selected. In the main area, under 'Manage DNS records', a CNAME record is being configured:

Type	Name	Target	TTL	Add Record
CNAME	www.mohdfaikhan.tech	44572ef784454bfb65b980390b4de-1940812985.eu-north-1.amazonaws.com	14400	Add Record

Below this, a table lists existing DNS records:

Type	Name	Priority	Content	TTL	Delete	Edit
CNAME	www	0	a4572ef784454bfb65b980390b4de-1940812985.eu-north-1.amazonaws.com	14400	Delete	Edit

Screenshot of the AWS Certificate Manager (ACM) console showing a certificate named 'c7af2bac518b'. The 'Domains' section lists two domains: 'mohdfaikhan.tech.' and 'www.mohdfaikhan.tech.'. The 'Details' section provides certificate metadata.

In use	Serial number	Requested at	Renewal eligibility
No	N/A	March 31, 2025, 20:03:01 (UTC+06:30)	Ineligible
Domain name	Public key info	Issued at	
mohdfaikhan.tech.	RSA 2048	N/A	

Screenshot of the Hostinger domain management interface showing the DNS tab. It displays two nameservers: 'ns1.dns-parking.com' and 'ns2.dns-parking.com'. A CNAME record for 'www' is being added with the target '5a4494a798a.zfyfvmchrl.acm-validations.aws' and a TTL of 14400.

Type	Name	Target	TTL
CNAME	_298b1665b91981e95534639504ae28f4.mohdfaikhan.tech.	5a4494a798a.zfyfvmchrl.acm-validations.aws	14400
CNAME	www	845727ef784454bfdb765b98339bb4de-1940612985.eu-north-1.amazonaws.com	14400

The screenshot shows the Hostinger DNS Management interface. On the left sidebar, under 'Main menu', 'Domain Overview' is selected. Under 'DNS / Nameservers', there is a button 'Change Nameservers'. The main area displays two nameservers: 'ns1.dns-parking.com' and 'ns2.dns-parking.com'. A success message 'DNS Record created successfully.' is visible. Below this, the 'Manage DNS records' section shows a table with one entry: a CNAME record for 'www' pointing to 'a45722ef784454bf8b165b988391b4de-1040812985.eu-north-1.elb.amazonaws.com' with a TTL of 14400. The status bar at the bottom indicates the date as 31-03-2025.

The screenshot shows the AWS Certificate Manager (ACM) interface. On the left sidebar, 'AWS Certificate Manager (ACM)' is selected. The main area shows a certificate named 'c7af2bac518b' with the type 'Amazon Issued'. Below it, the 'Domains (2)' section lists two domains: 'mohdfaikhan.tech' and 'www.mohdfaikhan.tech'. Both domains are in a 'Pending validation' status. The 'CNAME name' column shows values like '381e05534639504ae28f4.mohdfaikhan.' and '875a9cb08d1d1902ba1e3d61741ba9cd.www.mohdfaikhan.tech'. A tooltip 'Copied to clipboard' is shown over the first value. The status bar at the bottom indicates the date as 31-03-2025.

The screenshot shows the Hostinger DNS / Nameservers management interface. On the left sidebar, under 'Domain Overview', the 'DNS / Nameservers' option is selected. In the main area, under 'Manage DNS records', a CNAME record is being added for 'www'. The 'Name' field contains 'cdwww.mohdfaikhan.tech' and the 'Target' field contains '985.eu-north-1.elb.amazonaws.com'. The TTL is set to 14400. An 'Add Record' button is visible. Below this, a table lists existing records, including one for 'www' with the same target and TTL.

The screenshot shows the AWS Certificate Manager (ACM) interface. A CNAME record named 'c7af2bac518b' has been created for the domain 'mohdfaikhan.tech'. The 'Type' is listed as 'Amazon Issued'. Under the 'Domains (2)' section, two entries are shown: one for 'mohdfaikhan.tech' with the CNAME value '110eacd76a44941a798a.zlyfvmchrlscm-validation.aws' and another for 'www.mohdfaikhan.tech' with the CNAME value '26aefcb3100ddca97265c4b177fe4e528.zlyfvmchrlscm-validation.aws'. Both entries have a status of 'Copied to clipboard'.

The screenshot shows the Hostinger DNS Management interface. On the left sidebar, under 'DNS / Nameservers', a purple button labeled 'Change Nameservers' is visible. The main area displays two sections: 'Manage DNS records' and 'Nameservers'. In the 'Manage DNS records' section, a table lists existing records, including a CNAME record for 'www' pointing to 'a45722ef784454bf(b)165b98539)14de-1040812985.eu-north-1.alb.amazonaws.com'. A new record is being added with the following details:

Type	CNAME	Name	875a9cb08d1d902ba1e3d6	Target	24b177fe4e528.ztylvmchr1acm-validations.aws	TTL	14400	Add Record
------	-------	------	------------------------	--------	---	-----	-------	------------

A success message 'DNS Record created successfully' is displayed in a toast notification.

**NOTE : Jab tum ACM ke records add kar loge, toh kuch der baad apni application ko HTTPS URLs par access kar paoge. Yeh process 15-20 minutes ya usse zyada bhi le sakta hai. ACM me status "Success" aane tak wait karo.**

**3. Reacords Add karne ke kuch der baad ACM par jaye or Refresh kare aapko Certificate Status ka Status Issued and Domains ka Status Success dikhega**

YE KUCH ISTARHA LAGEGA

The screenshot shows the AWS Certificate Manager (ACM) interface. On the left, a sidebar lists options like 'List certificates', 'Request certificate', 'Import certificate', and 'AWS Private CA'. The main area displays a certificate named 'cf1961af-d633-4d11-9c2a-c7af2bac518b'. The 'Certificate status' section shows the identifier 'cf1961af-d633-4d11-9c2a-c7af2bac518b' and ARN 'arnawsacmeu-north-1:195275659054:certificate/cf1961af-d633-4d11-9c2a-c7af2bac518b'. The status is 'Issued'. Below this, the 'Domains' section lists two entries:

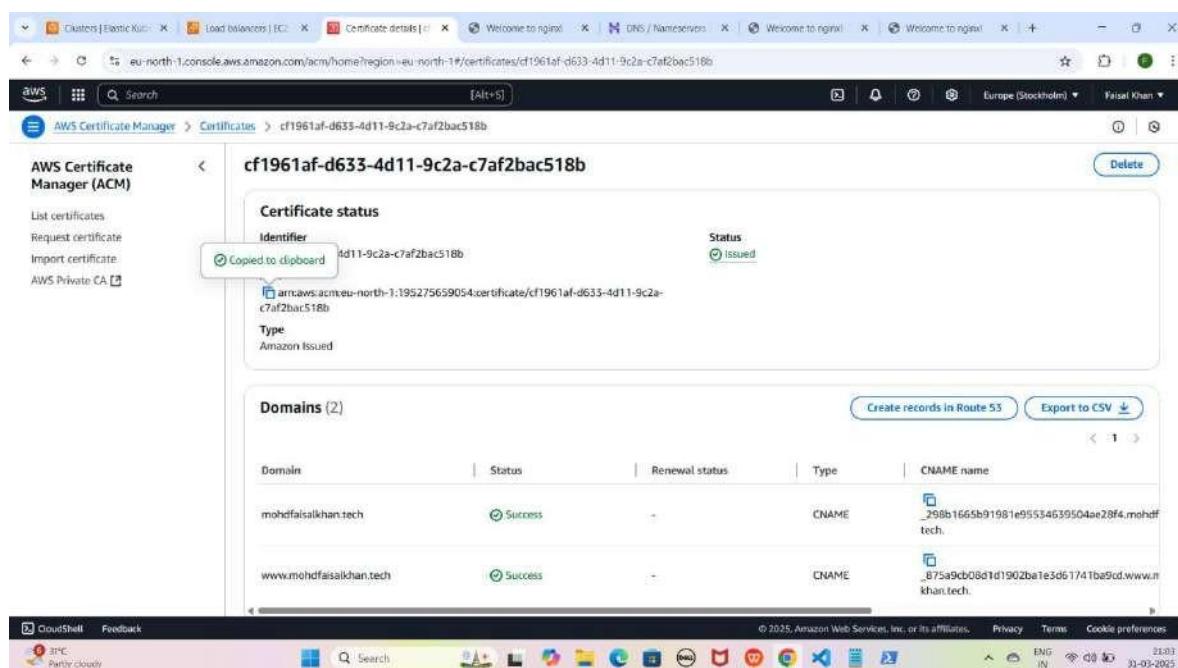
Domain	Status	Renewal status	Type	CNAME name
mohdfalsalkhan.tech	Success	-	CNAME	_298b1665b91981e95534639504ae28f4.mohdfalsalkhan.tech.
www.mohdfalsalkhan.tech	Success	-	CNAME	_875a9cb08d1d1902ba1e5961741ba9cd.www.mohdfalsalkhan.tech.

## Step 4: ingress.yaml file ka kaam

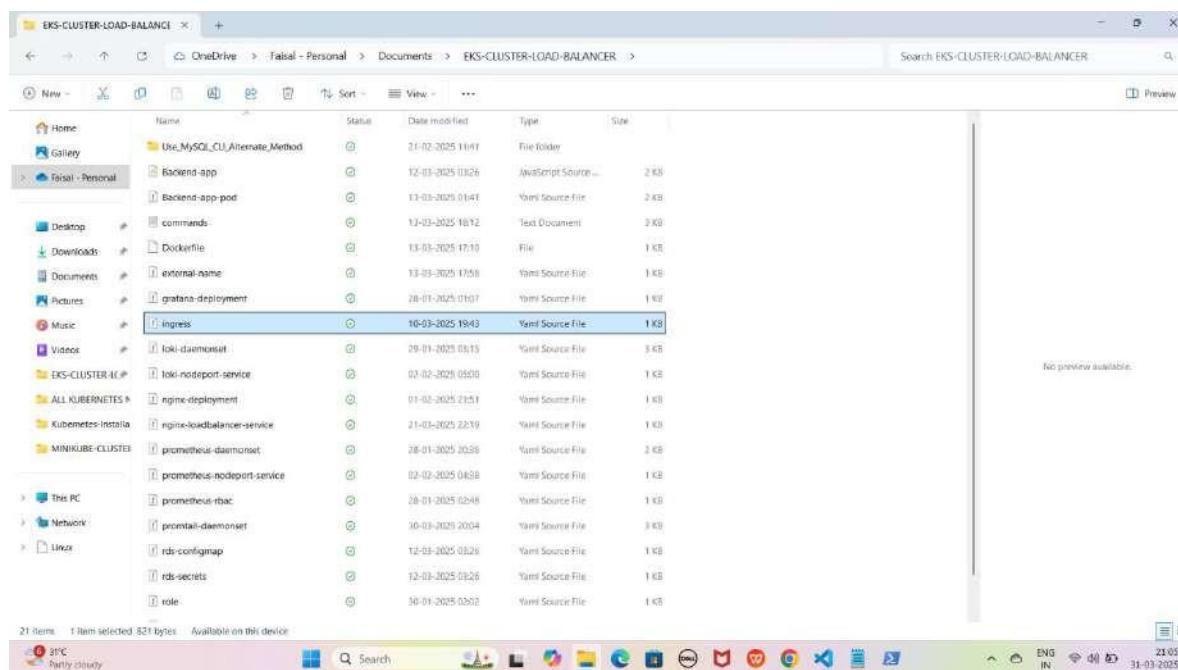
Ye file hum apne **domain name** ke liye **routing** provide karne ke liye use kar raha hain.

Lekin pehle humein **ACM** successfully issued hone ke baad, is **ingress.yaml** file me **ARN** Copy karke aur **Domain Name** update karna hoga.

YE KUCH ISTARHA LAGEGA



The screenshot shows the AWS Certificate Manager interface. A certificate named 'cf1961af-d633-4d11-9c2a-c7af2bac518b' is displayed. The status is 'Issued'. Two domains are listed: 'mohdfaikhan.tech' and 'www.mohdfaikhan.tech', both with a status of 'Success'. The 'Domains' table has columns: Domain, Status, Renewal status, Type, and CNAME name. The CNAME names are long random strings. There are buttons for 'Create records in Route 53' and 'Export to CSV'.



The screenshot shows a Windows file explorer window titled 'EKS-CLUSTER-LOAD-BALANCER'. The 'ingress' file is selected in the list. The file size is 1 KB. The file was modified on 10-03-2025 at 19:42. The file type is YAML Source File. The file path is 'OneDrive > Faisal - Personal > Documents > EKS-CLUSTER-LOAD-BALANCER > ingress'. The status bar at the bottom shows '21 items 1 item selected 821 bytes Available on this device'.

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-ingress
  annotations:
    alb.ingress.k8s.aws/certificate-arm: arn:aws:acm:eu-north-1:195275659954:certificate/cf1961af-d633-4d11-9c2a-c7a2bac510b
    alb.ingress.k8s.aws/scheme: internet-facing
spec:
  ingressClassName: nginx
  rules:
    - host: mohdfaiskhan.tech
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: nginx-service
                port:
                  number: 80
    - host: www.mohdfaiskhan.tech
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: nginx-service
                port:
                  number: 80
```

```
File Edit Selection View Go Run Terminal Help < > Search
File Edit Selection View Go Run Terminal Help < > Search
New Text File Ctrl+N
New File... Ctrl+Alt+Windows+N
New Window Ctrl+Shift+N
New Window with Profile
Open File... Ctrl+O
Open Folder... Ctrl+K Ctrl+O
Open Workspace from File...
Open Recent
Add Folder to Workspace...
Save Workspace As...
Duplicate Workspace
Save Ctrl+S
Save As... Ctrl+Shift+S
Save All Ctrl+Alt+S
Share > service
Auto Save
Preferences > an_tech
Revert File
Close Editor Ctrl+F4
Close Window Alt+F4
Exit > service
```

**NOTE: Yaha par mera Domain Name mohdfaiskhan.tech hai, jo humne pichle step me Hostinger se Domain Buy karte waqt dekha tha. Aapka Domain Name alag ho sakta hai or aapko update karna hoga aapke Domain name ke According**

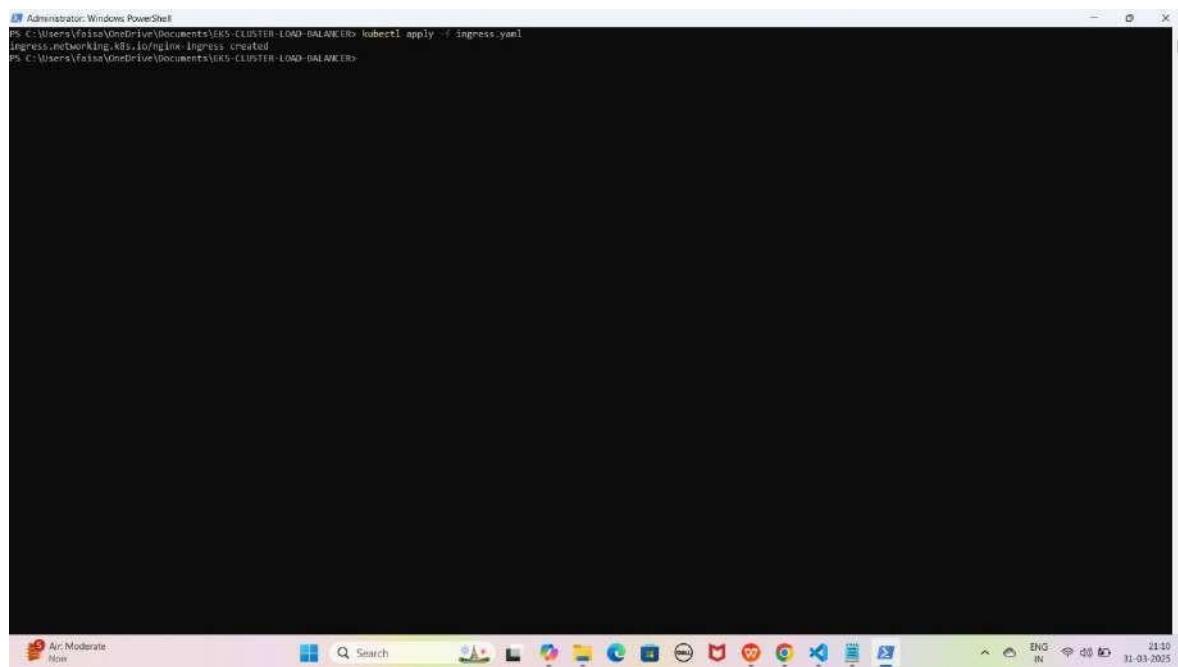
**JAISE KI:-**

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-ingress
  annotations:
    alb.ingress.k8s.aws/certificate-arn: arn:aws:acm:eu-north-1:195275659054:certificate/cf1961af-d633-4d11-9c2a-c7af2bac518b
    alb.ingress.k8s.aws/scheme: internet-facing
spec:
  ingressClassName: nginx
  rules:
    - host: mohdfaiskhan.tech
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: nginx-service
                port:
                  number: 80
    - host: www.mohdfaiskhan.tech
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: nginx-service
                port:
                  number: 80
```

**Ingress Service Apply Karo:**

```
kubectl apply -f ingress.yaml
```

## YE KUCH ISTARHA LAGEGA



```
Administrator: Windows PowerShell
PS C:\Users\fatima\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER> kubectl apply -f ingress.yaml
Ingress.networking.k8s.io/ingress created
PS C:\Users\fatima\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER>
```

# Part 6: Load Balancer Settings and Updates

## Step 1: Load Balancer ki Security Settings Update Karo

1. Load Balancer Page par jao
2. Load Balancer section me jao. Or Load Balancer select karo
3. "Security" tab me jao aur apne security group ko open karo. Or Edit inbound rules pe click karo
4. Sab existing rules remove karo.
5. Naya rule add karo:
  - Type: All Traffic
  - Source: Anywhere IPv4
  - Save karo.

YE KUCH ISTARHA LAGEGA

The screenshot shows the AWS EC2 Load Balancers interface. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances, Images, and Elastic Block Store. The main content area displays a table titled 'Load balancers (1/1)' with one item. Below the table, a detailed view of the load balancer 'a45727ef784454bfdb165b988391b4de' is shown, with the 'Details' tab selected. The details include the load balancer type (Classic), status (1 of 1 instance in service), VPC (vpc-0e345f19effbea822), and date created (March 31, 2025). The table below lists the load balancer's name, DNS name, state, VPC ID, availability zones, type, and date created.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date create
a45727ef784454bfdb165b...	a45727ef784454bfdb165b...	-	vpc-0e345f19effbea822	3 Availability Zones	classic	March 31, 2025

Screenshot of the AWS EC2 Load Balancers console showing a single load balancer named "a45727ef784454bfdb165b988391b4de".

The table lists the load balancer details:

Name	DNS name	State	VPC ID	Availability Zones	Type	Date create
a45727ef784454bfdb165b988391b4de	a45727ef784454bfdb165b988391b4de	-	vpc-0e345f19effbea822	3 Availability Zones	classic	March 31, 2025

The Security tab is selected, showing one security group assigned:

Security Group ID	Name	Description
sg-048a691d8e3ee6585	k8s-elb-a4572...	Security group for Kubernetes ELB a45727ef784454bfdb165b988391b4de (default/nginx-service)

Screenshot of the AWS EC2 Security Groups console showing a single security group named "sg-048a691d8e3ee6585 - k8s-elb-a45727ef784454bfdb165b988391b4de".

The Details tab shows the following information:

Security group name	Security group ID	Description	VPC ID
k8s-elb-a45727ef784454bfdb165b988391b4de	sg-048a691d8e3ee6585	Security group for Kubernetes ELB a45727ef784454bfdb165b988391b4de (default/nginx-service)	vpc-0e345f19effbea822

The Inbound rules tab shows two rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range
sgr-0b9e7ad7262e165e3	sgr-0b9e7ad7262e165e3	IPv4	HTTP	TCP	80
sgr-04b27e7476bd8495b	sgr-04b27e7476bd8495b	IPv4	Custom ICMP - IPv4	Destination Unreachable	fragmentation required

Clusters | Elastic Load balancers | Modify inbound | Certificate details | Welcome to nginx | DNS / Nameservers | Welcome to nginx | Welcome to nginx | +

eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#ModifyInboundSecurityGroupRules;securityGroupId=sg-048a691d8e3ee6585

aws Search [Alt+S] Europe (Stockholm) Fatal Khan

EC2 > Security Groups > sg-048a691d8e3ee6585 - k8s-elb-a45727ef784454bfb165b988391b4de > Edit inbound rules

## Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
sgr-0b9e7ad7262e165e3	HTTP	TCP	80	Custom	<input type="text" value="0.0.0.0/0"/> <input type="button" value="Delete"/>
sgr-04b27e747GbdB495b	Custom ICMP - IPv4	Desti...	frag...	Custom	<input type="text" value="0.0.0.0/0"/> <input type="button" value="Delete"/>

[Add rule](#)

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

The screenshot shows the AWS CloudShell interface with the following details:

- CloudShell** and **Feedback** buttons are at the top left.
- The top right features the text "© 2025, Amazon Web Services, Inc. or its affiliates.", **Privacy**, **Terms**, and **Cookie preferences**.
- A toolbar with various icons for file operations like Open, Save, Print, etc.
- The status bar at the bottom right shows "ENG IN", a signal strength icon, and the date "31-03-2025".
- The main browser window displays the URL "eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#ModifyInboundSecurityGroupRules;securityGroupId=sg-048a691d8e3ee65b5 - k8s-elb-a45227ef784454bfdb165b888391b4de > Edit inbound rules".
- The AWS logo and search bar are visible at the top of the browser window.
- The page content includes:
  - Edit inbound rules** (with an info link).
  - A note: "Inbound rules control the incoming traffic that's allowed to reach the instance."
  - A section titled "Inbound rules" (with an info link) containing the message: "This security group has no inbound rules."
  - A "Add rule" button.
  - Buttons for "Cancel", "Preview changes", and "Save rules".

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences Today's moment Elfit Tower Day ENG IN 21:45 31-03-2025

The screenshot displays the AWS CloudShell interface with two browser tabs open:

- Top Tab:** Shows the 'Edit inbound rules' page for a security group. It includes a table for defining rules, a note about using 0.0.0.0/0, and buttons for 'Cancel', 'Preview changes', and 'Save rules'.
- Bottom Tab:** Shows the EC2 dashboard with a success message: "Inbound security group rules successfully modified on security group (sg-048a691d8e3ee6585 | k8s-elb-a45727ef784454bfb165b988391b4de) | Details". It also lists security group details like name, ID, owner, and VPC ID.

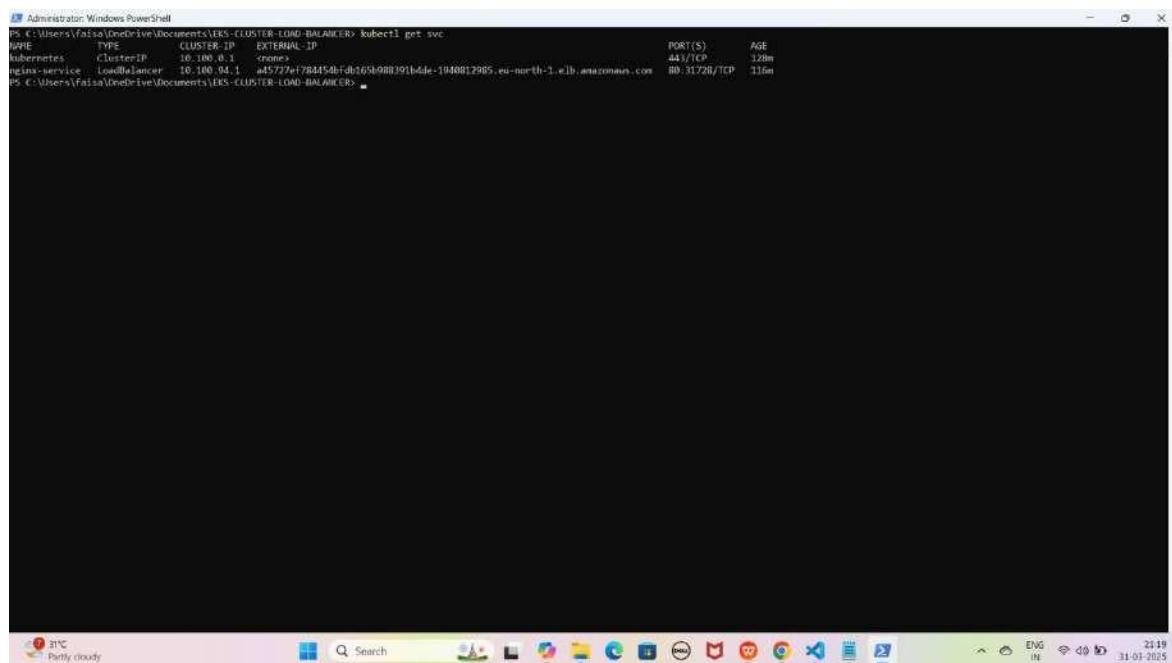
## Step 2: Load Balancer me Listeners Update Karo

### 1. PowerShell me command run karo:

```
kubectl get svc
```

### 2. Ye command tumhe Load Balancer ka port number dikhayega. Jase ki mere case me 80:31728/TCP

## YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
kubernetes     ClusterIP  10.100.0.1   <none>        443/TCP       128m
nginx-service  LoadBalancer 10.100.0.1   a45737ef78d454b1db165b988391bde-1940812985.eu-north-1.elb.amazonaws.com  80:31728/TCP   116m
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

3. Load Balancer page pe jao aur "Listeners" tab open karo.
4. "Manage Listeners" pe click karo.
5. Existing listener Remove karo and new Listeners Add karo Add Listener pe click karke

JAISE KI:-

- **HTTP (Port 80) → HTTP (Instance Port 31728)**  
Naya listener add karo:
- **HTTPS (Port 443) → HTTP (Instance Port 31728)**
- ACM ka SSL Certificate select karo.
- Save karo.

# YE KUCH ISTARHA LAGEGA

The screenshot shows the AWS EC2 console with the 'Manage listeners' page for a specific load balancer. The left sidebar shows navigation options like Dashboard, EC2 Global View, Instances, Images, and Elastic Block Store. The main content area is titled 'Manage listeners' and displays 'Load balancer details: a45727ef784454bfdb165b988391b4de'. Under the 'Listeners (1)' section, there is one listener configuration:

Listener protocol	Port	Instance protocol	Instance port	Security policy	Default SSL/TLS certificate	Cookie stickiness
TCP	80	TCP	31728	Not applicable	Not applicable	Not applicable

Below the table, there is a note: 'You can add up to 99 more.' At the bottom right of the page are 'Cancel' and 'Save changes' buttons.

This screenshot is identical to the one above, but it includes a message at the bottom of the 'Listeners (1)' section: 'This listener will be removed upon saving changes. [Link](#)'. The rest of the interface and data are the same as the first screenshot.

Screenshot of the AWS Management Console showing the 'Manage listeners' page for a Classic Load Balancer.

The left sidebar shows navigation links for EC2, including Dashboard, EC2 Global View, Events, Instances (selected), Images, and Elastic Block Store.

The main content area displays the 'Manage listeners' interface for a load balancer with ARN `a45727ef784454bfdb165b988391b4de`.

**Listeners (3)**

The listeners configured on your Classic Load Balancer (CLB) define how client requests and network traffic are routed within your application. Here you can add new listeners, modify existing listeners, or delete listeners as your needs change.

Listener protocol	Port	Instance protocol	Instance port	Security policy		
TCP	80	TCP	31728	This listener will be removed upon saving changes. <a href="#">Undo</a>		
HTTP	80	HTTP	31728	Not applicable	Not applicable	Disabled <a href="#">Edit</a>
HTTPS	443	HTTP	31728	ELBSecurityPolicy-2016-08	<a href="#">Choose a certificate</a> <a href="#">Edit</a>	Disabled <a href="#">Edit</a>

[Add listener](#)

You can add up to 97 more.

**Server-side tasks and status**

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 21:28 31-03-2025

Screenshot of the AWS Management Console showing the 'Manage listeners' page for a Classic Load Balancer.

The left sidebar shows navigation links for EC2, including Dashboard, EC2 Global View, Events, Instances (selected), Images, and Elastic Block Store.

The main content area displays the 'Manage listeners' interface for a load balancer with ARN `a45727ef784454bfdb165b988391b4de`.

**Listeners (3)**

The listeners configured on your Classic Load Balancer (CLB) define how client requests and network traffic are routed within your application. Here you can add new listeners, modify existing listeners, or delete listeners as your needs change.

Listener protocol	Port	Instance protocol	Instance port	Security policy		
TCP	80	TCP	31728	This listener will be removed upon saving changes. <a href="#">Undo</a>		
HTTP	80	HTTP	31728	Not applicable	Not applicable	Disabled <a href="#">Edit</a>
HTTPS	443	HTTP	31728	ELBSecurityPolicy-2016-08	<a href="#">Choose a certificate</a> <a href="#">Edit</a>	Disabled <a href="#">Edit</a>

[Add listener](#)

You can add up to 97 more.

**Server-side tasks and status**

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 21:28 31-03-2025

Screenshot of the AWS CloudFront console showing the 'Manage listeners' page for a specific load balancer. The 'Listeners' section shows three listeners: one for HTTP on port 80 and two for HTTPS on port 443. The HTTPS listener is selected, and a modal window titled 'Edit SSL certificate for HTTPS:443' is open. In the 'Default SSL/TLS certificate' dropdown, 'From ACM' is selected, and a list of available certificates is shown, including 'mohdfaikhan.tech' and 'mohdfaikhan.tech'. The 'Save changes' button at the bottom right of the modal is highlighted.

Screenshot of the AWS CloudFront console showing the 'Manage listeners' page for a specific load balancer. The 'Listeners' section shows three listeners: one for HTTP on port 80 and two for HTTPS on port 443. The HTTPS listener is selected, and a modal window titled 'Edit SSL certificate for HTTPS:443' is open. In the 'Default SSL/TLS certificate' dropdown, 'From ACM' is selected, and a list of available certificates is shown, including 'mohdfaikhan.tech' and 'mohdfaikhan.tech'. The 'Save changes' button at the bottom right of the modal is highlighted.

The screenshot shows the AWS EC2 management console with the 'Manage listeners' page open for a specific load balancer. The left sidebar lists various EC2 services like Dashboard, Global View, and Instances. The main content area displays 'Load balancer details: a45727ef784454bfd165b988391b4de'. Below this, a section titled 'Listeners (3)' provides information about how client requests and network traffic are routed. It includes a table with columns: Listener protocol, Port, Instance protocol, Instance port, and Security policy. The first row shows a listener for TCP port 80 to instance port 31728, which is set to be removed. The second row shows an HTTP listener on port 80 to instance port 31728 with no security policy. The third row shows an HTTPS listener on port 443 to instance port 31728 using the ELBSecurityPolicy-2016-08 policy. At the bottom, there's an 'Add listener' button and a note about adding up to 99 more.

Listener protocol	Port	Instance protocol	Instance port	Security policy
TCP	80	TCP	31728	This listener will be removed upon saving changes. <a href="#">Undo</a>
HTTP	80	HTTP	31728	Not applicable
HTTPS	443	HTTP	31728	ELBSecurityPolicy-2016-08 <a href="#">Edit</a>

eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#CLBManageListenersloadBalancerArn=a45727ef784454bfbdb165b988391b4de

Search [Alt+5]

EC2 > Load balancers > a45727ef784454bfbdb165b988391b4de > Manage listeners

**EC2**

Dashboard

EC2 Global View

Events

**Instances**

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

**Images**

- AMIs
- AMI Catalog

**Elastic Block Store**

- Volumes
- Snapshots
- Lifecycle Manager

CloudShell Feedback

Listeners (3)

The listeners configured on your Classic Load Balancer (CLB) define how client requests and network traffic are routed within your application. Here you can add new listeners, modify existing listeners, or delete listeners as your needs change.

Listener protocol	Port	Instance protocol	Instance port	Security policy			
TCP	80	TCP	31728	This listener will be removed upon saving changes. <a href="#">Undo</a>			
HTTP	80	HTTP	31728	Not applicable	Not applicable	Disabled <a href="#">Edit</a>	<a href="#">Remove</a>
HTTPS	443	HTTP	31728	ELBSecurityPolicy-2016-08	ACM: mohdfaikhan.tech	Disabled <a href="#">Edit</a>	<a href="#">Edit</a> <a href="#">Remove</a>

Add listener

You can add up to 81 more

▶ Server-side tasks and status

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Cancel [Save changes](#)

Screenshot of the AWS Cloud Console showing the Load Balancer details for a Classic Load Balancer.

**Load balancer type:** Classic

**Status:** 1 of 1 instance in service

**VPC:** vpc-0e345f19effbea622

**Date created:** March 31, 2025, 19:22 (UTC+05:30)

**Hosted zone:** Z23TA26LKFMNIO

**Availability Zones:**

- subnet-0d3bebe2909cd48f eu-north-1a (eu-n1-az1)
- subnet-04bd5fb1ac41b856c eu-north-1b (eu-n1-az2)
- subnet-03d77b5f5d3349800 eu-north-1c (eu-n1-az3)

**DNS name:** a45727ef784454bfdb165b988391b4de-1940812985.eu-north-1.elb.amazonaws.com (A Record)

**Migration Wizard:** This Classic Load Balancer can be migrated to a next generation load balancer. Migration wizard uses your load balancer's current configurations to create a new load balancer. [Learn more](#) [Launch migration wizard](#)

**Distribution of targets by Availability Zone (AZ):** For each enabled Availability Zone, you can view the number of registered instances and their current health states. Selecting any values here will apply the corresponding filter to the Target instances table.

Screenshot of the AWS Cloud Console showing the Load Balancer details for a Classic Load Balancer.

**Listeners:**

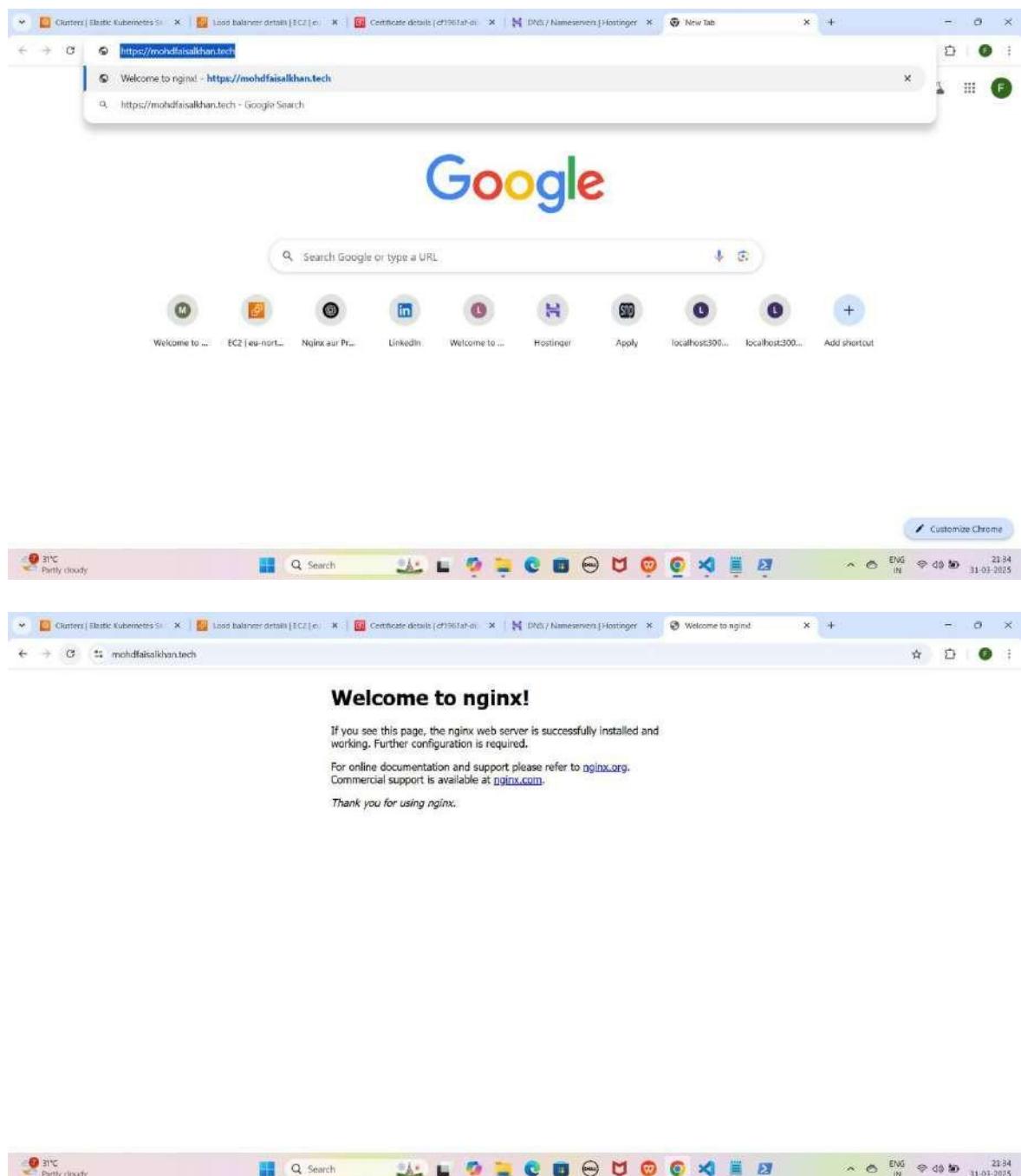
Protocol/Port	Instance Protocol	Security policy	Default SSL/TLS certificate	Cookie stickiness
HTTP:80	HTTP:31728	Not applicable	Not applicable	Not applicable
HTTPS:443	HTTP:31728	ELBSecurityPolicy-2016-08	mohdfaikhan.tech (Certificate ID: cf1961af...)	Not applicable

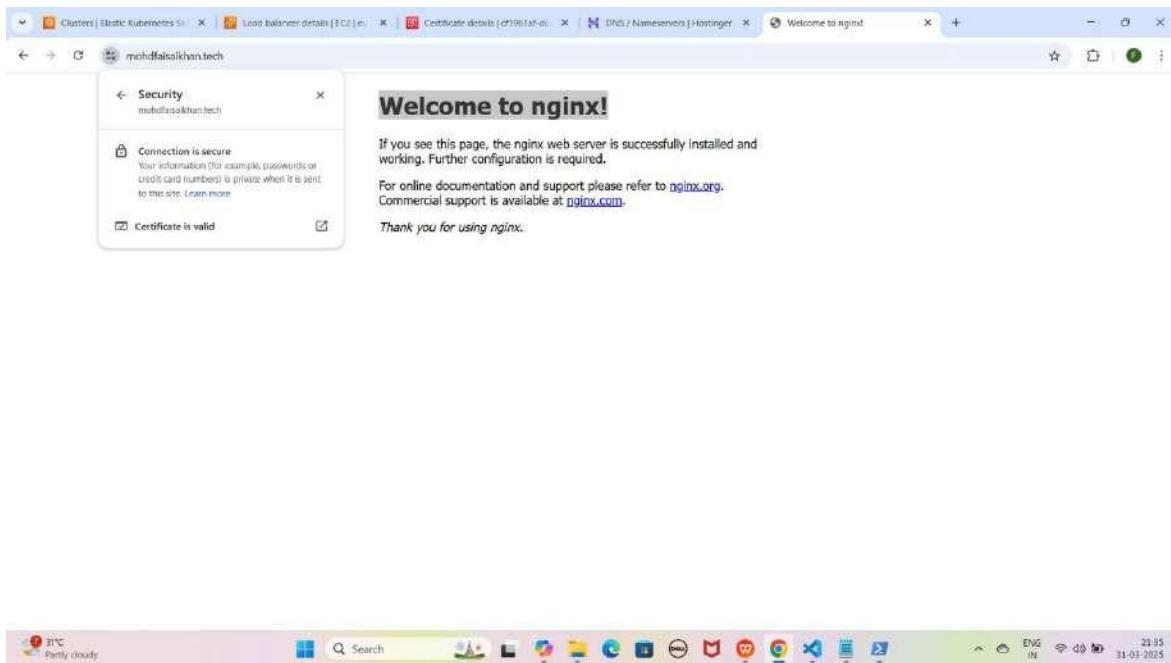
## Step 3: Listeners Update Karne Ke Baad Website HTTPS Domain Se Access Karna

### 1. Apni NGINX application ko Domain name se access karo:

- Nginx: <https://mohdfaikhan.tech>

YE KUCH ISTARHA LAGEGA

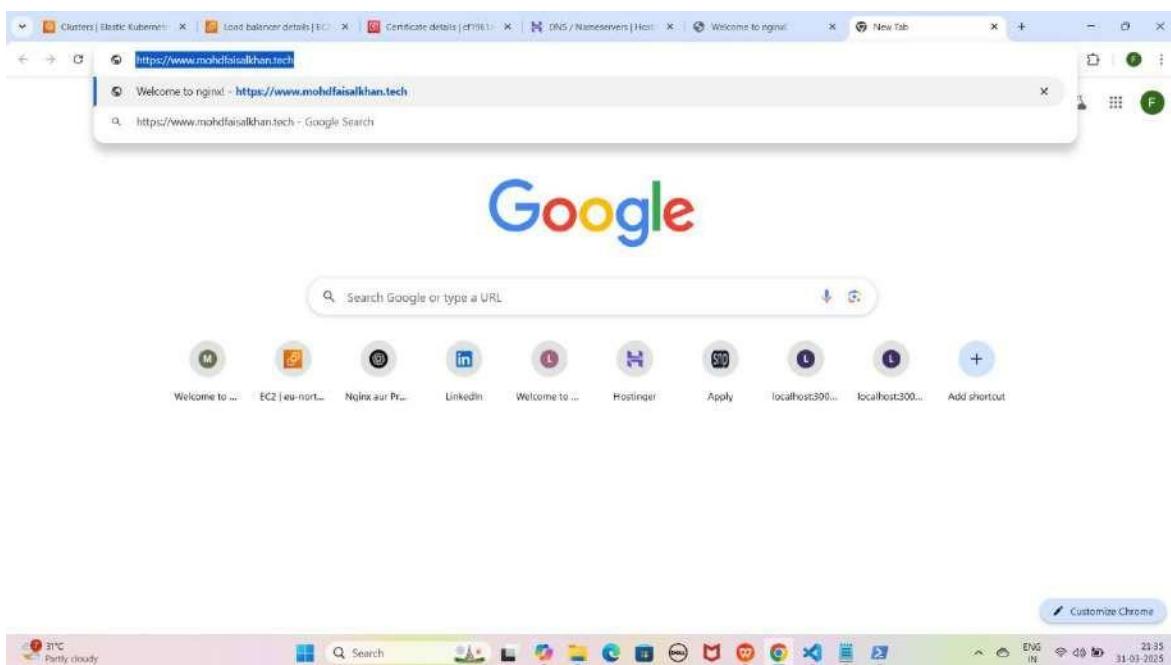




## 2. Apni NGINX application ko www wale Domain se access karo:

- Nginx: <https://www.mohdfaiskhan.tech>

YE KUCH ISTARHA LAGEGA





## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*



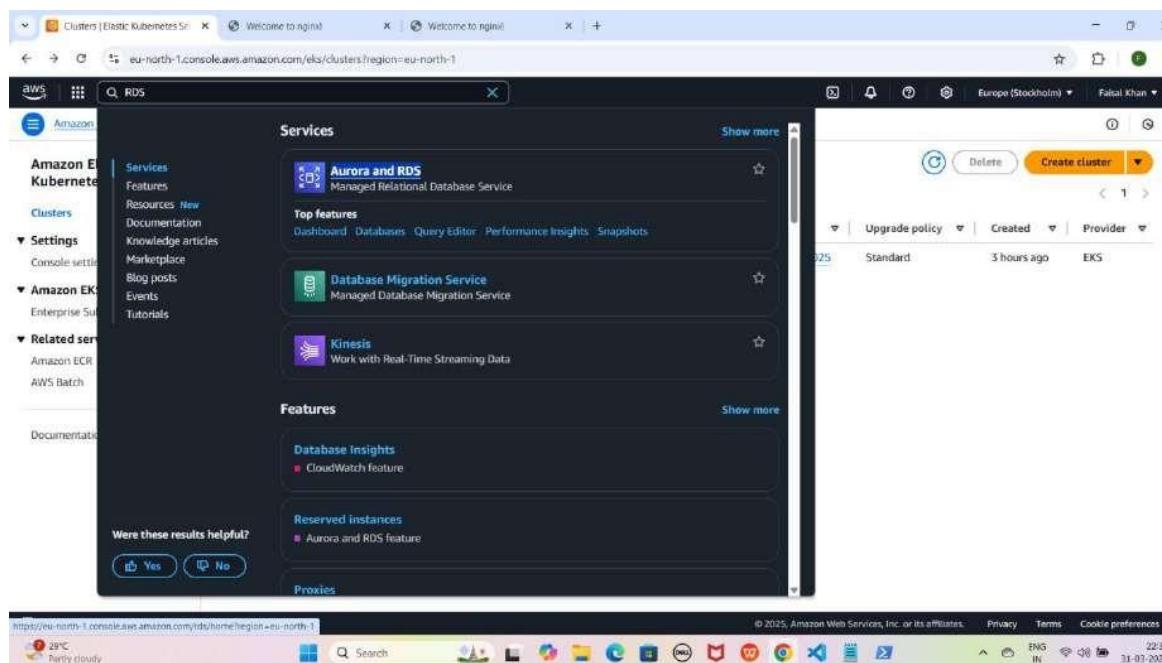
## Part 7: Amazon RDS (MySQL, Oracle, PostgreSQL) Setup and Creation

MySQL, Oracle, aur PostgreSQL databases AWS RDS per create karne ke liye yeh simple steps follow karein:

### Step 1: Subnet Group Create Karna

- AWS Console open karein, **RDS** search karein aur naye tab mein open karein.
- **Subnet Groups → Create DB Subnet Group**
- **Name:** faisal-subnet
- **Description:** Faisal Databases (required hai warna error aayega)
- **Default VPC** select karein.
- **Availability Zones:** eu-north-1a, eu-north-1b, eu-north-1c
- **Subnets :** eu-north-1a, eu-north-1b
- **Create** par click karein.
- Pop-up aayega: "**Successfully created faisal-subnet**"

YE KUCH ISTARHA LAGEGA



Screenshot of the AWS RDS Dashboard showing resources and recommended services.

**Resources**

- DB Instances (0/40)
  - Allocated storage (0 TB/100 TB)
  - Instances and storage include Neptune and DocumentDB. Increase DB instances limit [\[?\]](#)
- DB Clusters (0/40)
  - Reserved instances (0/40)
  - Snapshots (0)
    - Manual
      - DB Cluster (0/100)
      - DB Instance (0/100)
    - Automated
      - DB Cluster (0)
      - DB Instance (0)
  - Recent events (0)
  - Event subscriptions (0/20)
- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations [New](#)

**Create database**

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

You can use a backup from Amazon S3 to restore and create a new Aurora MySQL and MySQL database.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 22:44 31-03-2025

Screenshot of the AWS RDS Subnet Groups page showing a single subnet group.

**Subnet groups (1)**

Name	Description	Status	VPC
default-vpc-0e345f19effbea822	Created from the RDS Management Console	Complete	vpc-0e345f19effbea822

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 22:44 31-03-2025

Screenshot of the AWS RDS console showing the creation of a DB subnet group named "faisal-subnet". The subnet group is associated with the "Default VPC (vpc-De345f19effbea822)" which contains 3 subnets across 3 availability zones.

**Create DB subnet group**

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

**Subnet group details**

**Name**  
You won't be able to modify the name after your subnet group has been created.  
 Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

**Description**

**VPC**  
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.  
 3 Subnets, 3 Availability Zones

**Add subnets**

**Availability Zones**  
Choose the Availability Zones that include the subnets you want to add.

**Subnets**

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 22:52 31-03-2025

Screenshot of the AWS RDS console showing the selection of three availability zones (eu-north-1a, eu-north-1b, eu-north-1c) for the DB subnet group. A note indicates that for Multi-AZ DB clusters, 3 subnets must be selected in 3 different Availability Zones.

**Add subnets**

**Availability Zones**  
Choose the Availability Zones that include the subnets you want to add.  
  
 eu-north-1a  
 eu-north-1b  
 eu-north-1c  
  
For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

**Subnets selected (0)**

Availability zone	Subnet name	Subnet ID	CIDR block
No subnets added to this group			

Cancel **Create**

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 22:52 31-03-2025

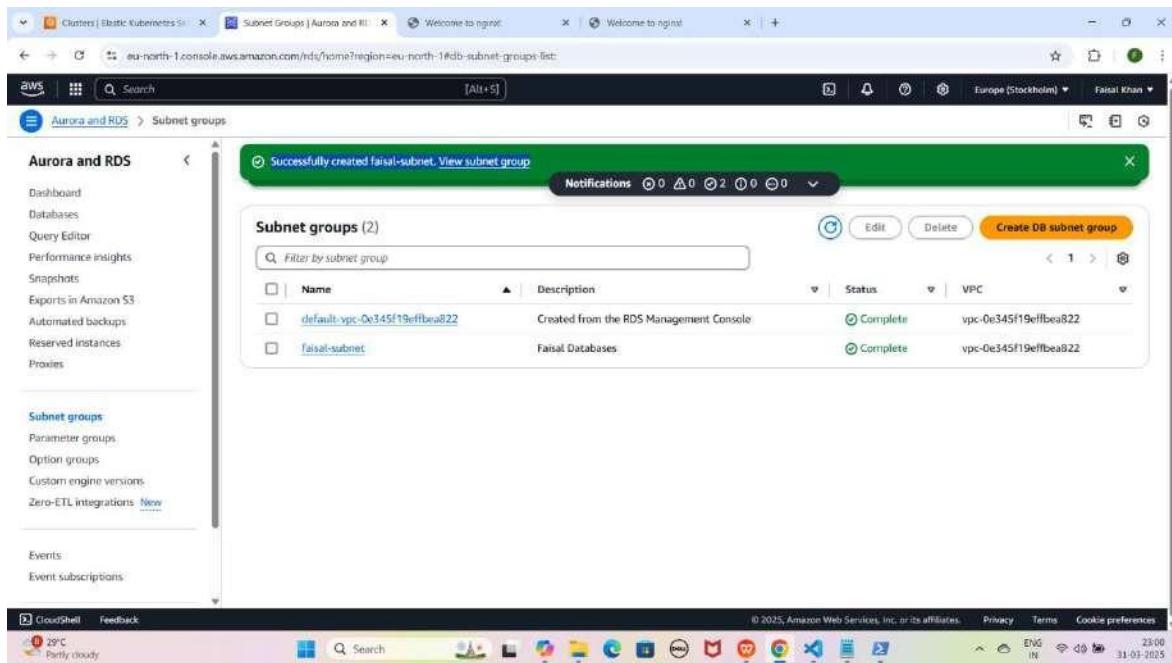
Screenshot of the AWS RDS Subnet Groups creation interface:

The page shows the "Add subnets" section. Under "Availability Zones", three zones are selected: eu-north-1a, eu-north-1b, and eu-north-1c. Under "Subnets", a dropdown menu is open, showing several subnet options. A note at the bottom states: "For Multi-AZ DB clusters, you must select 5 subnets in 3 different Availability Zones." The "Subnets selected (0)" table has four columns: Availability zone, Subnet name, Subnet ID, and CIDR block. The table currently displays: "No subnets added to this group".

Screenshot of the AWS RDS Subnet Groups creation interface:

The page shows the "Add subnets" section. Under "Availability Zones", three zones are selected: eu-north-1a, eu-north-1b, and eu-north-1c. Under "Subnets", a dropdown menu is open, showing several subnet options. The "Subnets selected (0)" table has four columns: Availability zone, Subnet name, Subnet ID, and CIDR block. The table currently displays: "No subnets added to this group".

At the bottom right of the interface, there are "Cancel" and "Create" buttons.



## Step 2: MySQL Database Create Karna

- **Databases → Create Database**
- **Creation Method:** Standard Create
- **Engine Type:** MySQL
- **Template:** Free Tier
- **DB Name:** faisal-mysql-db
- **Username:** FaisalKhan
- **Password:** Faisalkhan35\$
- **Storage Type:** General Purpose SSD (gp3)
- **Allocated Storage:** 20 GiB
- **Subnet Group:** faisal-subnet
- **Public Access:** No
- **VPC Security Group:** ALL-TRAFFIC-ALLOW (Default hata dein)
- **Availability Zone:** eu-north-1a
- **AWS KMS Key:** Faisal-Keys
- **Create Database** par click karein.
- Pop-up aayega: "**Creating database faisal-MYSQL-db**"

## YE KUCH ISTARHA LAGEGA

The screenshot shows the AWS RDS Databases page. The left sidebar is titled "Aurora and RDS" and includes sections for Dashboard, Database, Query Editor, Performance Insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, and Event subscriptions. The main content area is titled "Databases (0)" and features a search bar labeled "Filter by database". Below the search bar are columns for DB identifier, Status, Role, Engine, Region ..., Size, and Recommendations. A message at the bottom states "No instances found". The top navigation bar shows the URL "eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases:" and the AWS logo. The bottom status bar includes "CloudShell Feedback", "29°C Partly cloudy", and system icons.

The screenshot shows the "Create database" wizard. The first step, "Choose a database creation method", offers two options: "Standard create" (selected) and "Easy create". The "Standard create" option is described as setting all configuration options, including ones for availability, security, backups, and maintenance. The "Easy create" option is described as using recommended best-practice configurations, noting that some options can be changed after creation. The second step, "Engine options", allows selecting an engine type: MySQL (selected), Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), PostgreSQL, MariaDB, and Oracle. Each engine has a corresponding icon. To the right, a "MySQL" section provides a brief overview and a bulleted list of features. The top navigation bar shows the URL "eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-database:" and the AWS logo. The bottom status bar includes "CloudShell Feedback", "29°C Partly cloudy", and system icons.

Screenshot of the AWS RDS MySQL creation wizard:

### Templates

Choose a sample template to meet your use case.

- Production**: Use defaults for high availability and fast, consistent performance.
- Dev/Test**: This instance is intended for development use outside of a production environment.
- Free tier**: Use MySQL Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info

### Availability and durability

Deployment options: Info

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the Amazon RDS service level agreement (SLA) Info

- Multi-AZ DB cluster deployment (3 instances)**: Creates a primary DB instance with two read-only standby instances across multiple Availability Zones. This setup provides:
  - 99.99% uptime
  - Redundancy across Availability Zones
  - Increased read capacity
  - Reduced write latency
- Multi-AZ DB instance deployment (2 instances)**: Creates a primary DB instance with a read-redundant instance in a separate Availability Zone. This setup provides:
  - 99.95% uptime
  - Redundancy across Availability Zones
- Single-AZ DB instance deployment (1 instance)**: Creates a single DB instance without standby instances. This setup provides:
  - 99.9% uptime
  - No data redundancy

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Screenshot of the AWS RDS MySQL creation wizard (continued):

### Settings

**DB instance identifier**: Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "myDBinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

### Credentials Settings

**Master username**: Info

Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. The first character must be a letter.

### Credentials management

You can use AWS Secrets Manager to manage your master user credentials.

**Managed in AWS Secrets Manager - most secure**

RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

**Auto generate password**

AWS RDS can generate a password for you, or you can specify your own password.

**Master password**: Info

Password strength: Very strong

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Screenshot of the AWS RDS 'Create database' wizard showing 'Master password' configuration.

**Credentials management**

You can use AWS Secrets Manager or manage your master user credentials.

- Managed in AWS Secrets Manager - most secure**  
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.
- Self managed**  
Create your own password or have RDS create a password that you manage.

**Auto generate password**  
Amazon RDS can generate a password for you, or you can specify your own password.

**Master password** [Info](#)  
\*\*\*\*\*

**Password strength** Very strong

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ ^ @

**Confirm master password** [Info](#)  
\*\*\*\*\*

**Instance configuration**

The DB instance configuration options below are limited to those supported by the engine that you selected above.

**DB instance class** [Info](#)

**Hide filters**

Show instance classes that support Amazon RDS Optimized Writes [Info](#)

**CloudShell** **Feedback**

29°C Partly cloudy

Search

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:05 31-03-2025

Screenshot of the AWS RDS 'Create database' wizard showing 'Storage' configuration.

**Storage**

**Storage type** [Info](#)  
Provisioned IOPS-SSD (io2) storage volumes are now available.

**General Purpose SSD (gp3)**  
Performance scales independently from storage

**Allocated storage** [Info](#)  
20 GiB  
Minimum: 20 GiB, Maximum: 6,144 GiB

**Provisioned IOPS** [Info](#)  
5000 IOPS  
Baseline IOPS of 5,000 IOPS is included for allocated storage less than 400 GiB.

**Storage throughput** [Info](#)  
125 MiBps  
Baseline storage throughput of 125 MiBps is included for allocated storage less than 400 GiB.

To provision additional IOPS and throughput, increase the allocated storage to 400 GiB or greater.

**Additional storage configuration**

**CloudShell** **Feedback**

29°C Partly cloudy

Search

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:07 31-03-2025

The screenshot shows the 'Create database' wizard for MySQL. The first step, 'DB subnet group', lists three options: 'default-vpc-Ge45f19effbea822' (selected), 'default-vpc-Ge45f19effbea822' (disabled), and 'faisal-subnet' (disabled). The 'No' option under 'Public access' is selected. The 'VPC security group (firewall)' section shows 'Choose existing' selected, with 'Choose existing VPC security groups' highlighted. The 'Existing VPC security groups' dropdown shows 'default' selected. The 'Availability Zone' dropdown shows 'No preference'. On the right, a 'MySQL' summary panel highlights its popularity and key features.

This screenshot is identical to the one above, showing the 'Create database' wizard for MySQL. It displays the same configuration options: selecting 'faisal-subnet' for the DB subnet group, choosing 'No' for public access, selecting 'Choose existing' for the VPC security group, and having 'default' selected in the Existing VPC security groups dropdown. The MySQL summary panel is also present on the right.

Clusters | Elastic Kubernetes Service | Create database | Aurora and | Welcome to nginx | Welcome to nginx

eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#Launch-dbinstance

Aurora and RDS > Create database

VPC security group (firewall) [Info](#)  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing  
Choose existing VPC security groups

Create new  
Create new VPC security group

Existing VPC security groups  
Choose one or more options

ALL-TRAFFIC-ALLOW [X](#)

Availability Zone [Info](#)  
eu-north-1a

RDS Proxy  
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resilience, and security.

Create an RDS Proxy [Info](#)  
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing [\[?\]](#)

Certificate authority - optional [Info](#)  
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)  
Expiry: May 15, 2061

If you don't select a certificate authority, RDS chooses one for you.

► Additional configuration

CloudShell Feedback

AWS Moderate Row

Q Search [Alt + S] Europe (Stockholm) Fatal Khan

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

ENG IN 23:10 31-01-2025

Clusters | Elastic Kubernetes Sa... Create database | Aurora and... Welcome to nginx! Welcome to nginx!

eu-north-1.console.aws.amazon.com/rds/?idm?region=eu-north-1#launch-dbinstance

aws Search [Alt+S]

Aurora and RDS > Create database

**Additional configuration**

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

**Database options**

Initial database name [Info](#)

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)

defaultmysql8-0

Option group [Info](#)

defaultmysql-8-0

**Backup**

Enable automated backups

Creates a point-in-time snapshot of your database.

**Backup retention period** [Info](#)

The number of days (1-35) for which automatic backups are kept.

1 day

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:12 31/01/2025

Screenshot of the AWS RDS 'Create database' configuration page for an Aurora MySQL database.

**AWS KMS key:** Faisal-Keys

**Account:** 195275659054

**KMS key ID:** d80ba9f9-b99e-4878-a392-bf744e75fe6c

**Maintenance**

**Auto minor version upgrade:** Enabled

**Maintenance window:** Choose a window (radio button selected)

**Deletion protection:** Enabled

**MySQL Information:**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Screenshot of the AWS RDS 'Databases' page showing the creation of a new database.

**Aurora and RDS**

**Databases**

**Databases (1)**

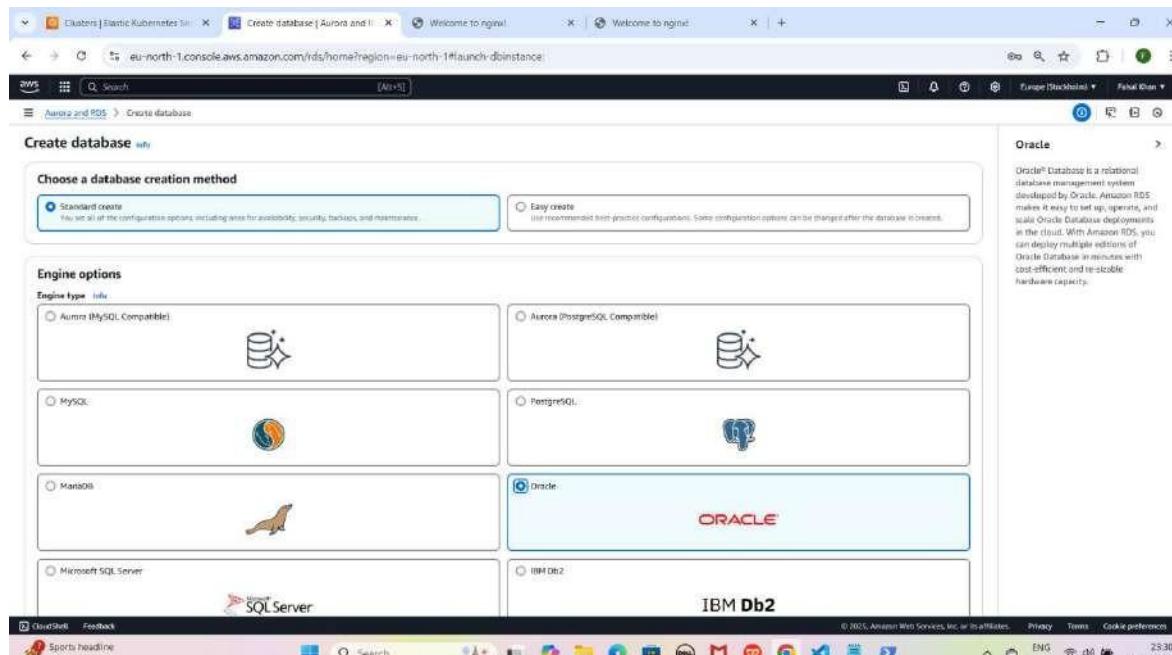
DB identifier	Status	Role	Engine	Region	Size	Recommendations
faisal-mysql-db	Creating	Instance	MySQL Com...	eu-north-1a	db.t4g.micro	

**CloudShell Feedback**

## Step 3: Oracle Database Create Karna

- **Databases → Create Database**
- **Creation Method:** Standard Create
- **Engine Type:** Oracle
- **Template:** Dev/Test
- **DB Name:** faisal-oracle-db
- **Username:** FaisalKhan
- **Password:** Faisalkhan35\$
- **Storage Type:** General Purpose SSD (gp3)
- **Allocated Storage:** 20 GiB
- **Subnet Group:** faisal-subnet
- **Public Access:** No
- **VPC Security Group:** ALL-TRAFFIC-ALLOW
- **Availability Zone:** eu-north-1a
- **AWS KMS Key:** Faisal-Keys
- **Create Database** par click karein.
- Pop-up aayega: "**Creating database faisal-Oracle-db**"

YE KUCH ISTARHA LAGEGA



Screenshot of the AWS RDS 'Create database' wizard step 1: Set instance type and configuration.

**Edition**

- Oracle Enterprise Edition
- Oracle Standard Edition Two

Affordable and full-featured database management system.

Affordable and full-featured database management system supporting up to 16 vCPUs. Oracle Database Standard Edition Two is a replacement for Standard Edition and Standard Edition One.

**License**

Bring Your Own License (BYOL)

**Engine version**

Oracle 19.0.0.0.ru-2025-01.rur-2025-01.r2

**Templates**

Choose a sample template to meet your use case.

- Production
- Dev/Test

This instance is intended for development use outside of a production environment.

**Settings**

**DB instance identifier** Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

**CloudShell** **Feedback**

**Oracle**

Oracle® Database is a relational database management system developed by Oracle. Amazon RDS makes it easy to set up, operate, and scale Oracle Database deployments in the cloud. With Amazon RDS, you can deploy multiple editions of Oracle Database in minutes with cost-efficient and re-usable hardware capacity.

Screenshot of the AWS RDS 'Create database' wizard step 2: Configure security and connectivity.

**Settings**

**DB instance identifier** Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

**Master username** Info

Type a login ID for the master user of your DB instance.

**Credentials management**

You can use AWS Secrets Manager or manage your master user credentials.

- Managed in AWS Secrets Manager – most secure
- Self managed

RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

If you manage the master user credentials in AWS Secrets Manager, additional charges apply. See [AWS Secrets Manager pricing](#). Additionally, some RDS features aren't supported. See limitations [here](#).

**Select the encryption key** Info

You can encrypt using the KMS key that Secrets Manager creates or a customer-managed KMS key that you create.

**CloudShell** **Feedback**

**Oracle**

Oracle® Database is a relational database management system developed by Oracle. Amazon RDS makes it easy to set up, operate, and scale Oracle Database deployments in the cloud. With Amazon RDS, you can deploy multiple editions of Oracle Database in minutes with cost-efficient and re-usable hardware capacity.

Screenshot of the AWS RDS 'Create database' wizard step 1: Set instance configuration.

**Master password:**  **Password strength:** Very strong

**Instance configuration:**

**DB instance class:**  Standard classes (includes m classes)

**Storage:**  General Purpose SSD (gp3)

**Allocated storage:** 20 GiB

**Provisioned IOPS:** 3000 IOPS

**Storage throughput:** 125 MiBps

**Additional storage configuration:** To provision additional IOPS and throughput, increase the allocated storage to 200 GiB or greater.

**Oracle:** Oracle Database is a relational database management system developed by Oracle. Amazon RDS makes it easy to set up, operate, and scale Oracle Database deployments in the cloud. With Amazon RDS, you can deploy multiple editions of Oracle Database in minutes with cost-efficient and re-usable hardware capacity.

Screenshot of the AWS RDS 'Create database' wizard step 2: Set instance configuration.

**Master password:**  **Password strength:** Very strong

**Instance configuration:**

**DB instance class:**  Standard classes (includes m classes)

**Storage:**  General Purpose SSD (gp3)

**Allocated storage:** 20 GiB

**Provisioned IOPS:** 3000 IOPS

**Storage throughput:** 125 MiBps

**Additional storage configuration:** To provision additional IOPS and throughput, increase the allocated storage to 200 GiB or greater.

**Oracle:** Oracle Database is a relational database management system developed by Oracle. Amazon RDS makes it easy to set up, operate, and scale Oracle Database deployments in the cloud. With Amazon RDS, you can deploy multiple editions of Oracle Database in minutes with cost-efficient and re-usable hardware capacity.

Clusters | Elastic Kubernetes Service | Create database | Aurora and RDS | Welcome to nginx! | Welcome to nginx! | - | x

eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-dbinstance

Aurora and RDS > Create database

After a database is created, you can't change its VPC.

**DB subnet group** Info  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.  
 default-vpc-0e345f19efbea822  
3 Subnets, 3 Availability Zones  
 default-vpc-0e345f19efbea822  
3 Subnets, 3 Availability Zones  
 falsal-subnet  
2 Subnets, 2 Availability Zones

**No**  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall)** Info  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.  
 Choose existing  
Choose existing VPC security groups  
 Create new  
Create new VPC security group

**Existing VPC security groups**  
Choose one or more options  
default

**Availability Zone** Info

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:33 31-03-2025

Oracle

Oracle® Database is a relational database management system developed by Oracle. Amazon RDS makes it easy to set up, operate, and scale Oracle Database deployments in the cloud. With Amazon RDS, you can deploy multiple editions of Oracle Database in minutes with cost-efficient and re-usable hardware capacity.

Clusters | Elastic Kubernetes Service | Create database | Aurora and RDS | Welcome to nginx! | Welcome to nginx! | - | x

eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-dbinstance

Aurora and RDS > Create database

After a database is created, you can't change its VPC.

**DB subnet group** Info  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.  
 falsal-subnet  
2 Subnets, 2 Availability Zones

**Public access** Info  
 Yes  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database.  
 No  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall)** Info  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.  
 Choose existing  
Choose existing VPC security groups  
 Create new  
Create new VPC security group

**Existing VPC security groups**  
Choose one or more options  
default

**Availability Zone** Info

Sports headline  
Other Mammals... CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:34 31-03-2025

Oracle

Oracle® Database is a relational database management system developed by Oracle. Amazon RDS makes it easy to set up, operate, and scale Oracle Database deployments in the cloud. With Amazon RDS, you can deploy multiple editions of Oracle Database in minutes with cost-efficient and re-usable hardware capacity.

Screenshot of the AWS RDS 'Create database' configuration page for Oracle.

**Existing VPC security groups:** Choose existing VPC security groups. Option: ALL-TRAFFIC-ALLOW.

**Availability Zone:** eu-north-1a

**Certificate authority - optional:** rds-ca-rsa2048-g1 (default) (Expiry: May 25, 2061)

**Tags - optional:** No tags associated with the resource.

**Monitoring:** Database Insights pricing is separate from RDS monthly estimates. See Amazon CloudWatch pricing.

**Performance Insights:** Enable Performance insights (checkbox checked). With Performance Insights dashboard, you can visualize the database load on your Amazon RDS DB instance load and filter the load by waits, SQL statements, hints, or users.

**Retention period:** 7 days (free tier)

**AWS KMS key:** Faisal-Keys

**Account:** 195275659054

**KMS key ID:** d80ba919-b99e-4878-a392-bf744e75fe6c

Screenshot of the AWS RDS 'Create database' configuration page for Oracle.

**Existing VPC security groups:** Choose existing VPC security groups. Option: ALL-TRAFFIC-ALLOW.

**Availability Zone:** eu-north-1a

**Certificate authority - optional:** rds-ca-rsa2048-g1 (default) (Expiry: May 25, 2061)

**Tags - optional:** No tags associated with the resource.

**Monitoring:** Database Insights pricing is separate from RDS monthly estimates. See Amazon CloudWatch pricing.

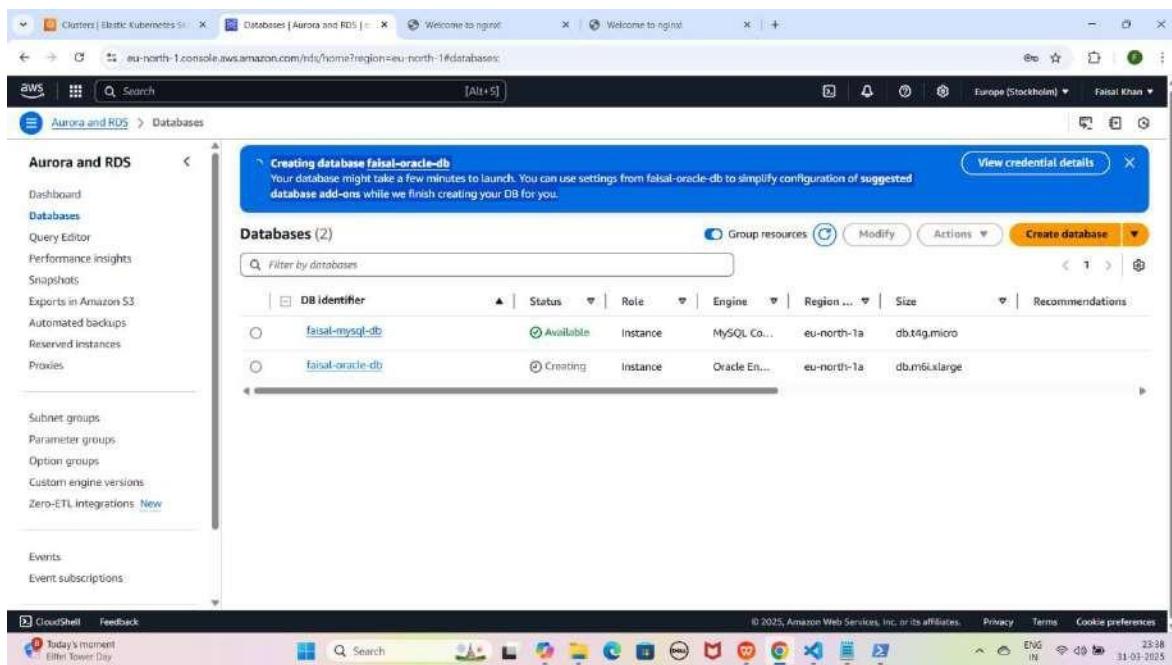
**Performance Insights:** Enable Performance insights (checkbox checked). With Performance Insights dashboard, you can visualize the database load on your Amazon RDS DB instance load and filter the load by waits, SQL statements, hints, or users.

**Retention period:** 7 days (free tier)

**AWS KMS key:** Faisal-Keys

**Account:** 195275659054

**KMS key ID:** d80ba919-b99e-4878-a392-bf744e75fe6c



## Step 4: PostgreSQL Database Create Karna

- **Databases → Create Database**
- **Creation Method:** Standard Create
- **Engine Type:** PostgreSQL
- **Template:** Free Tier
- **DB Name:** faisal-postgresql-db
- **Username:** FaisalKhan
- **Password:** Faisalkhan35\$
- **Storage Type:** General Purpose SSD (gp3)
- **Allocated Storage:** 20 GiB
- **Subnet Group:** faisal-subnet
- **Public Access:** No
- **VPC Security Group:** ALL-TRAFFIC-ALLOW
- **Availability Zone:** eu-north-1a
- **AWS KMS Key:** Faisal-Keys
- **Create Database** par click karein.
- Pop-up aayega: "Creating database faisal-PostgreSQL-db"

# YE KUCH ISTARHA LAGEGA

[eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-dbinstance](https://eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-dbinstance)

Aurora and RDS > Create database

### Create database Info

**Choose a database creation method**

- Standard create You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- Easy create Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

Engine type Info

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible) Selected
- MySQL
- PostgreSQL Selected
- MariaDB
- Oracle

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:40 31-03-2025

[eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-dbinstance](https://eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-dbinstance)

Aurora and RDS > Create database

### Templates

Choose a sample template to meet your use case.

- Production Use defaults for high availability and fast, consistent performance.
- Dev/Test This instance is intended for development use outside of a production environment.
- Free tier Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

**Availability and durability**

Deployment options Info

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

- Multi-AZ DB cluster deployment (5 instances) Create a primary DB instance with five read/write readReplica instances in separate Availability Zones. This setup provides:
  - 99.99% uptime
  - Redundancy across Availability Zones
  - Increased read capacity
  - Reduced write latency
- Multi-AZ DB instance deployment (2 instances) Create a primary DB instance with a failover standby instance in a separate Availability Zone. This setup provides:
  - 99.99% uptime
  - Redundancy across Availability Zones
- Single-AZ DB instance deployment (1 instance) Create a single DB instance without standby instances. This setup provides:
  - 99.95% uptime
  - No data redundancy

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:40 31-03-2025

The screenshot shows the 'Create database' wizard in the AWS RDS console. The current step is 'Settings'. The 'DB instance identifier' field contains 'faisal postgresql db'. The 'Master username' field contains 'Faisalkhan'. Under 'Credentials management', the 'Managed in AWS Secrets Manager - most secure' option is selected. A note states: 'RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.' The 'Self managed' option is also present. A note for 'Auto generate password' says: 'Amazon RDS can generate a password for you, or you can specify your own password.' On the right, a sidebar titled 'PostgreSQL' provides an overview of the database engine.

The screenshot shows the 'Create database' wizard in the AWS RDS console. The current step is 'Instance configuration'. It displays the DB instance configuration options, which are limited by the engine selected above. The 'DB instance class' dropdown is set to 'Info'. Under 'Hide filters', the 'Include previous generation classes' checkbox is checked. On the right, a sidebar titled 'PostgreSQL' provides an overview of the database engine.

Screenshot of the AWS Cloud console showing the creation of a PostgreSQL database. The storage configuration is being set up, including allocated storage (20 GiB), provisioned IOPS (3000), and storage throughput (125 MiBps). A note indicates that additional IOPS and throughput require 400 GiB or more.

**Storage**

**Allocated storage**: 20 GiB

**Provisioned IOPS**: 3000 IOPS

**Storage throughput**: 125 MiBps

To provision additional IOPS and throughput, increase the allocated storage to 400 GiB or greater.

**PostgreSQL**

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation of reliability, stability, and correctness.

- High reliability and stability in a variety of workloads.
- Advanced features to perform in high-volume environments.
- Vibrant open-source community that releases new features multiple times per year.
- Supports multiple extensions that add even more functionality to the database.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- The most Oracle-compatible open-source database.

Screenshot of the AWS Cloud console showing the creation of a PostgreSQL database. The VPC configuration is being set up, including selecting a DB subnet group (default-vpc-0e345f19effbea822) and choosing VPC security groups (Choose existing: default).

**DB subnet group**: default-vpc-0e345f19effbea822

**VPC security group (firewall)**: Choose existing: default

**Existing VPC security groups**: default

**Availability Zone**: No preference

**PostgreSQL**

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation of reliability, stability, and correctness.

- High reliability and stability in a variety of workloads.
- Advanced features to perform in high-volume environments.
- Vibrant open-source community that releases new features multiple times per year.
- Supports multiple extensions that add even more functionality to the database.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- The most Oracle-compatible open-source database.

**Create database | Aurora and RDS**

eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-dbinstance

Welcome to nginx!

Welcome to nginx!

Aurora and RDS > Create database

After a database is created, you can't change its VPC.

**DB subnet group** Info

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

**faishal-subnet** 2 Subnets, 2 Availability zones

**Public access** Info

Yes: RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No: RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall)** Info

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing: Choose existing VPC security groups.

Create new: Create new VPC security group.

**Existing VPC security groups**

Choose one or more options: default

**Availability Zone** Info

No preference

**CloudShell** **Feedback**

28°C Partly cloudy

Search

PostgreSQL

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation of reliability, stability, and correctness.

- High reliability and stability in a variety of workloads.
- Advanced features to perform in high-volume environments.
- Vibrant open-source community that releases new features multiple times per year.
- Supports multiple extensions that add even more functionality to the database.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- The most Oracle-compatible open-source database.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:43 31-03-2025

**Create database | Aurora and RDS**

eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-dbinstance

Welcome to nginx!

Welcome to nginx!

Aurora and RDS > Create database

Choose existing VPC security group: ALL-TRAFFIC-ALLOW

Creates new VPC security group.

**Existing VPC security groups**

Choose one or more options: ALL-TRAFFIC-ALLOW

**Availability Zone** Info

eu-north-1a

**RDS Proxy**

RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

Create an RDS Proxy

RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

**Certificate authority - optional** Info

Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-na2048-g1 (default) Expiry: May 25, 2061

If you don't select a certificate authority, RDS chooses one for you.

**Additional configuration**

**Tags - optional**

CloudShell Feedback

28°C Partly cloudy

Search

PostgreSQL

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation of reliability, stability, and correctness.

- High reliability and stability in a variety of workloads.
- Advanced features to perform in high-volume environments.
- Vibrant open-source community that releases new features multiple times per year.
- Supports multiple extensions that add even more functionality to the database.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- The most Oracle-compatible open-source database.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:44 31-03-2025

Screenshot of the AWS RDS 'Create database' wizard for PostgreSQL.

**Monitoring**

Choose monitoring tools for this database. Database Insights provides a combined view of Performance Insights and Enhanced Monitoring for your fleet of databases. Database Insights pricing is separate from RDS monthly estimates. See Amazon CloudWatch pricing.

Database Insights - Advanced

- Retains 15 months of performance history
- Fleet-level monitoring
- Integration with CloudWatch Application Metrics

Database Insights - Standard

- Retains 7 days of performance history, with the option to pay for the retention of up to 24 months of performance history

**Performance Insights**

Enable Performance insights With Performance Insights dashboard, you can visualize the database load on your Amazon RDS DB instance load and filter this load by waits, SQL statements, hosts, or users.

**Retention period**

7 days (free tier)

**AWS KMS key info**

Faisal-Keys

**Account**

195275659054

**KMS key ID**

d80b919-b99e-4878-a392-bf744e75fe6c

**CloudShell Feedback**

28°C Partly cloudy

Search bar: eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#launch-dbinstance

Right panel: **PostgreSQL**

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation of reliability, stability, and correctness.

- High reliability and stability in a variety of workloads.
- Advanced features to perform in high-volume environments.
- Vibrant open-source community that releases new features multiple times per year.
- Supports multiple extensions that add even more functionality to the database.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- The most Oracle-compatible open-source database.

Bottom status bar: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:44 31-03-2025

Screenshot of the AWS RDS 'Databases' page showing the creation of a PostgreSQL database.

**Aurora and RDS**

- Dashboard
- Databases
- Query Editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL Integrations: New

**Databases (3)**

Creating database **faisal-postgresql-db**. Your database might take a few minutes to launch. You can use settings from **faisal-postgresql-db** to simplify configuration of suggested database add-ons while we finish creating your DB for you.

**Create database**

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations
<a href="#">faisal-mysql-db</a>	Available	Instance	MySQL Co...	eu-north-1a	db.t4g.micro	
<a href="#">faisal-oracle-db</a>	Creating	Instance	Oracle En...	eu-north-1a	db.m6x.large	
<a href="#">faisal-postgresql-db</a>	Creating	Instance	PostgreSQL	eu-north-1a	db.t4g.micro	

**CloudShell Feedback**

28°C Partly cloudy

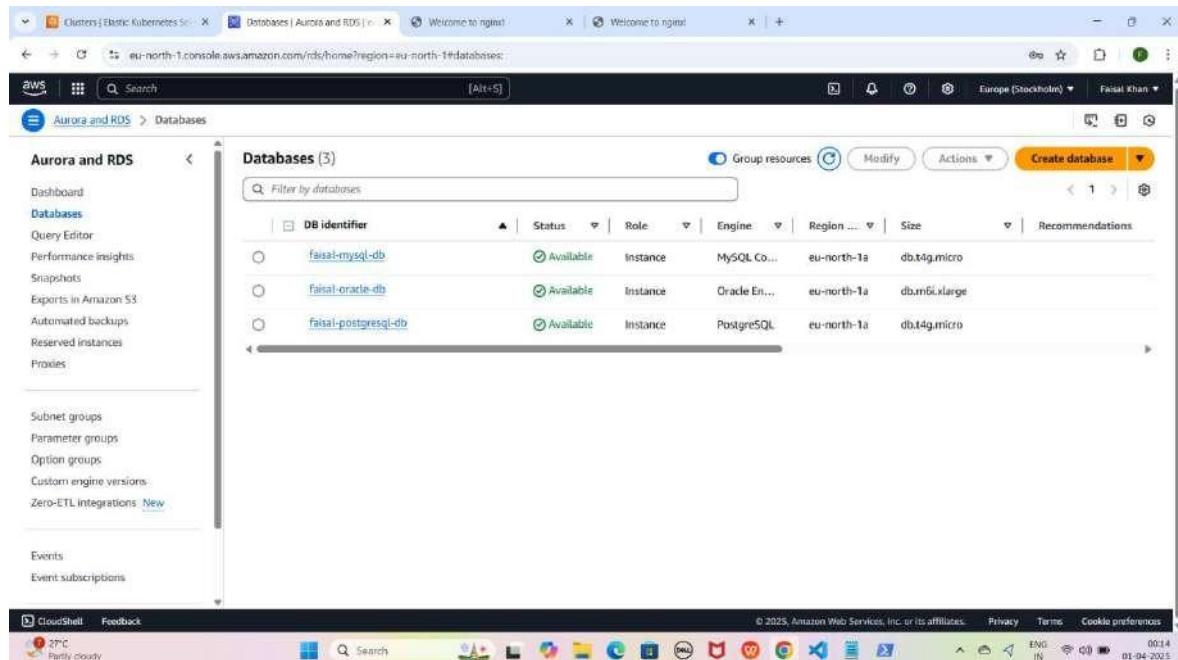
Search bar: eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases

Bottom status bar: © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:44 31-03-2025

## Step 5: Database Status Check Karna

- **Databases** section open karein.
- Thodi der wait karein, jab tak sab databases ka **Status = Available** na ho jaye.

YE KUCH ISTARHA LAGEGA



The screenshot shows the AWS RDS console interface. On the left, there's a sidebar with options like Dashboard, Databases, Query Editor, etc. The main area is titled 'Databases (3)' and lists three database instances:

DB identifier	Status	Role	Engine	Region	Size
faisal-mysql-db	Available	Instance	MySQL Co...	eu-north-1a	db.t4g.micro
faisal-oracle-db	Available	Instance	Oracle En...	eu-north-1a	db.m6i.xlarge
faisal-postgresql-db	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro

**NOTE:-**

1. Amazon RDS password me Special characters bhi allow hain, lekin "/" , "" aur '@' allow nahi hain.
2. gp3 storage → gp2 se 20% sasta aur better performance deta hai.
3. Public Access = No rakhein taake security maintain ho.
4. MySQL, Oracle, aur PostgreSQL ka Status = Available ho jaye, iska matlab AWS RDS successfully create ho gaya!

## **Part 8: Accessing Amazon RDS (MySQL, Oracle, PostgreSQL) from an EKS Cluster (With Endpoint)**

**Amazon RDS ko Access Karne ke 2 Methods Hai:-**

**Method 1: Application ke through Database Access Karna (With Endpoint)**

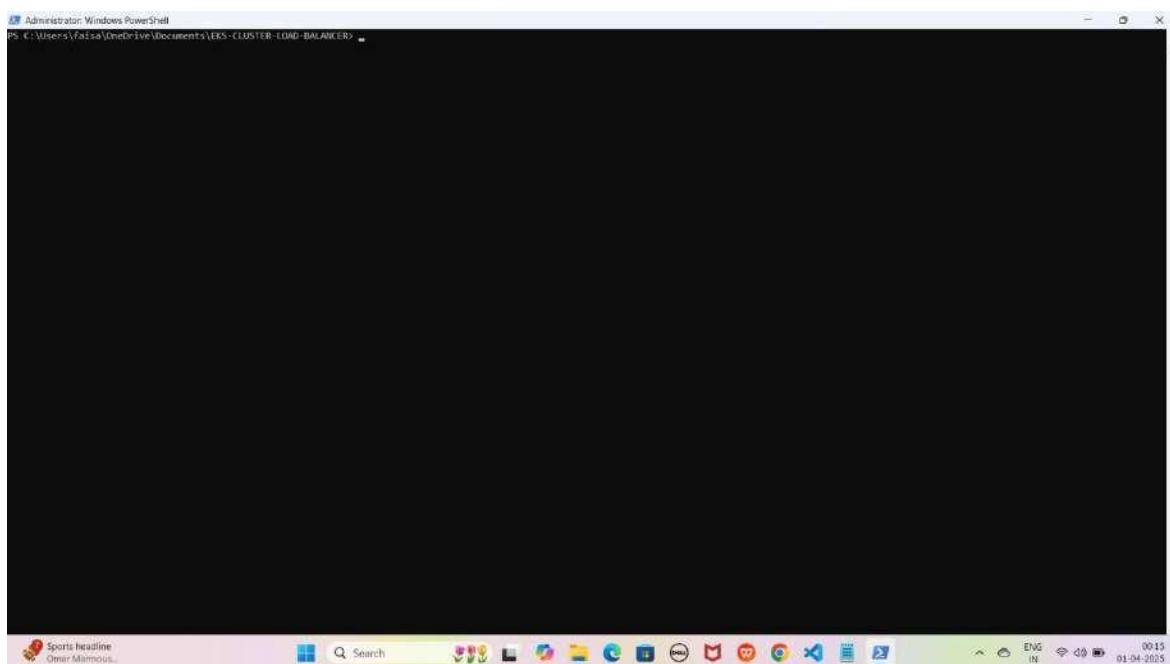
**Application image** ek pre-packaged file hoti hai jo tumhari **application ka code, dependencies, aur configurations** contain karti hai. Ye image **Docker** ya kisi aur **containerization tool** se banayi jati hai. Isko use karke tum apni **application ko kisi bhi environment me easily deploy** kar sakte ho.

**Step 1: Dependencies Install Karo**

**1. Project Folder ki Location par jao jaise ki mere case me kuch aisi hogi**

```
cd C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER
```

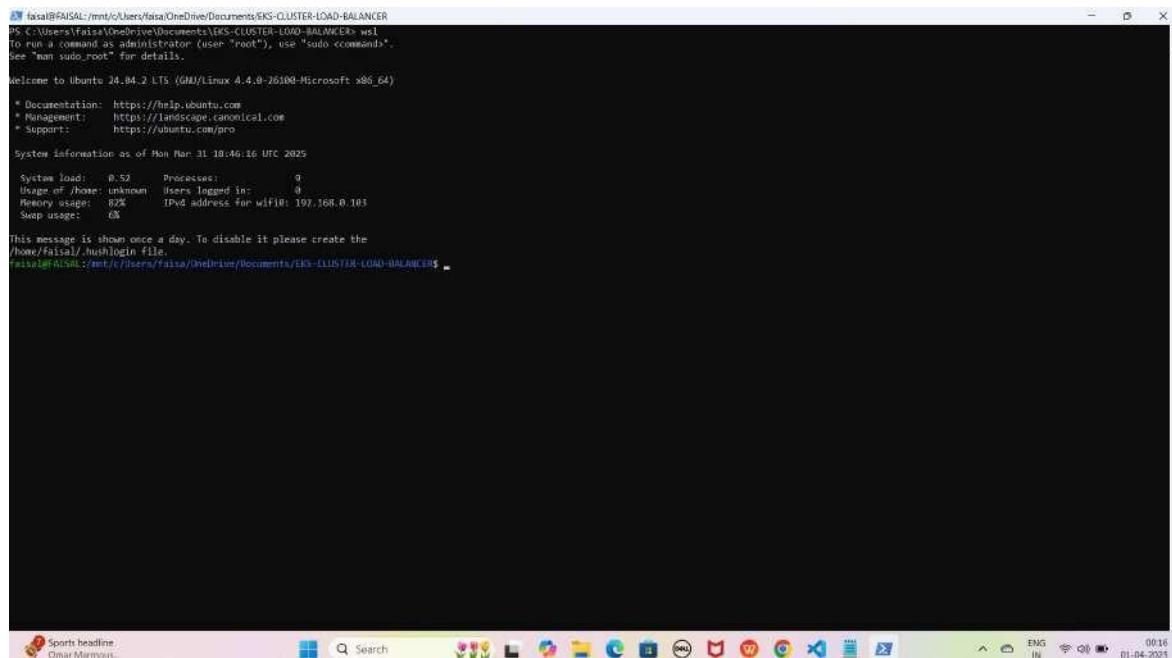
**YE KUCH IS TARHA LAGEGA**



## 2. Ab WSL me enter karo enter hone ke liye wsl type karo:

wsl

### YE KUCH IS TARHA LAGEGA



```
faisal@FAISAL:~$ /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER
PS C:\Users\faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> wsl
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 4.4.0-26500-Microsoft x86_64)

 * Documentation:  http://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Mar 31 18:46:16 UTC 2025

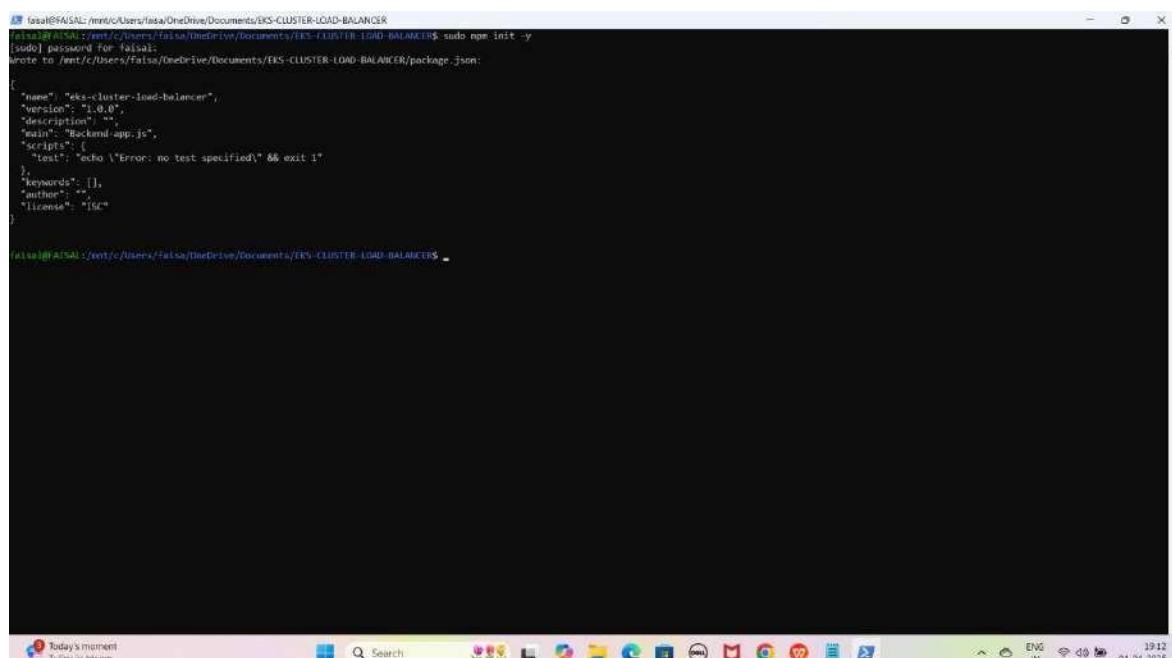
 System load: 0.52   Processes:          9
 Usage of /home: unknown  Users logged in:  0
 Memory usage: 88%    IPv6 address for wifi0: 197.168.0.183
 Swap usage:   0M

This message is shown once a day. To disable it please create the
/home/faisal/.hushlogin file.
faisal@FAISAL:~$ /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER
```

## 3. Backend-app.js initialize karne ke liye ye command run kariye:

sudo npm init -y

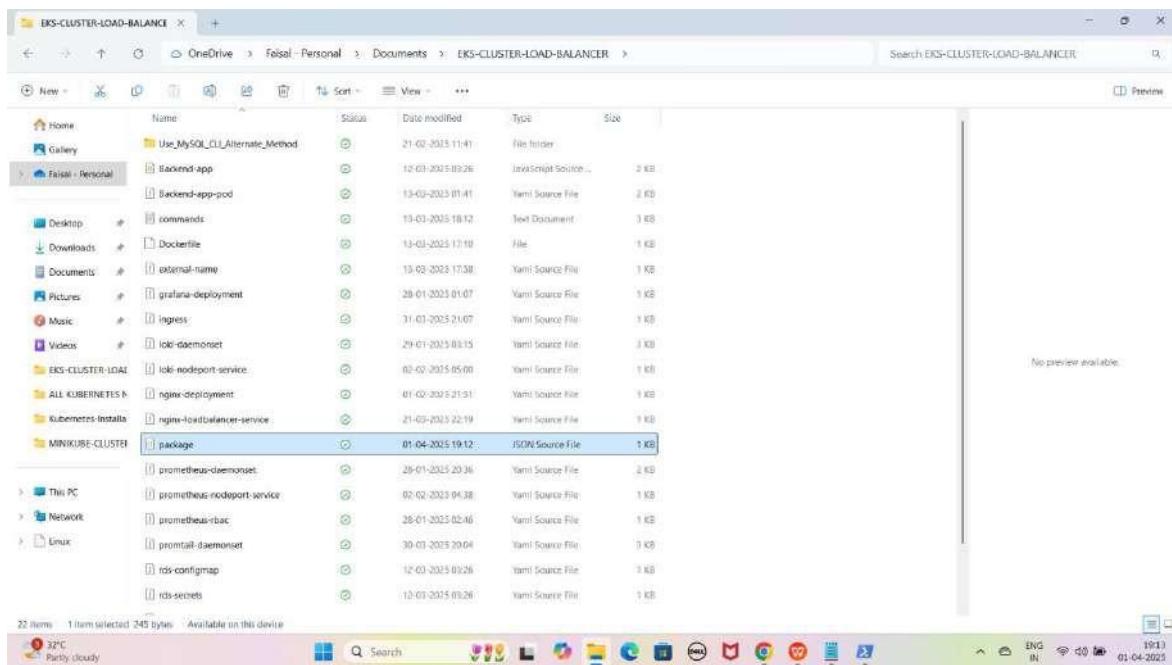
### YE KUCH ISTARHA LAGEGA



```
faisal@FAISAL:~$ /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER
faisal@FAISAL:~$ /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER> sudo npm init -y
[sudo] password for faisal:
wrote to /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER/package.json

{
  "name": "eks-cluster-load-balancer",
  "version": "1.0.0",
  "description": "",
  "main": "Backend-app.js",
  "scripts": {
    "test": "echo 'Error: no test specified' && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

faisal@FAISAL:~$ /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER
```

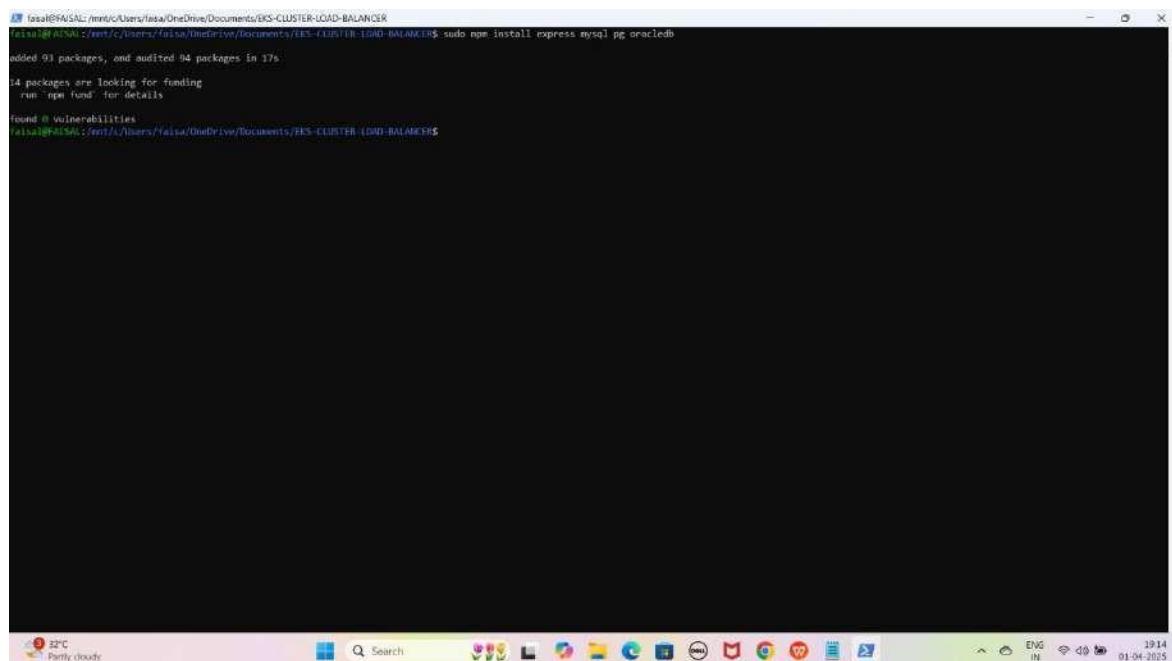


**NOTE: Yeh command Backend-aap.js initialize karti hai aur package.json create karta hai with default settings**

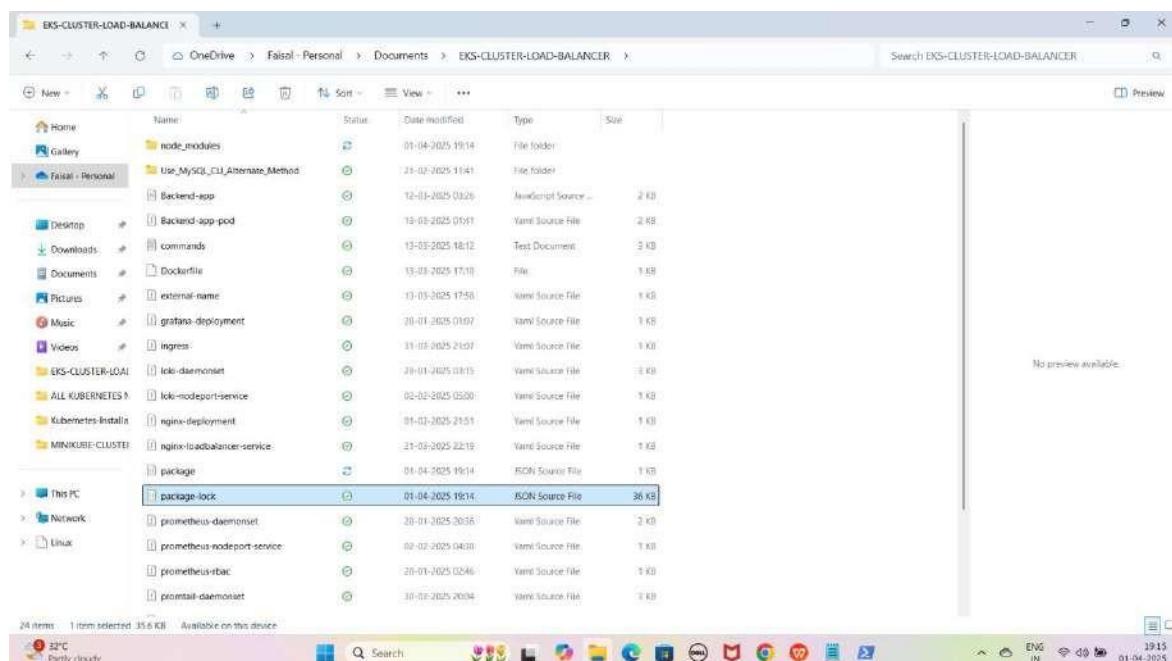
#### 4. Required Dependencies install karne ke liye ye command run kariye:

```
sudo npm install express mysql pg oracledb
```

YE KUCH ISTARHA LAGEGA



```
faisal@F450AL:~/mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$ sudo npm install express mysql pg oracledb
added 93 packages, and audited 94 packages in 17s
14 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
faisal@F450AL:~/mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$
```



**NOTE: Yeh command required dependencies (express, mysql, pg, oracledb) ke liye install karti hai or package-lock.json file bhi creraste karta hai .Saath hi package.json file me inko "dependencies" section me add karti hai**

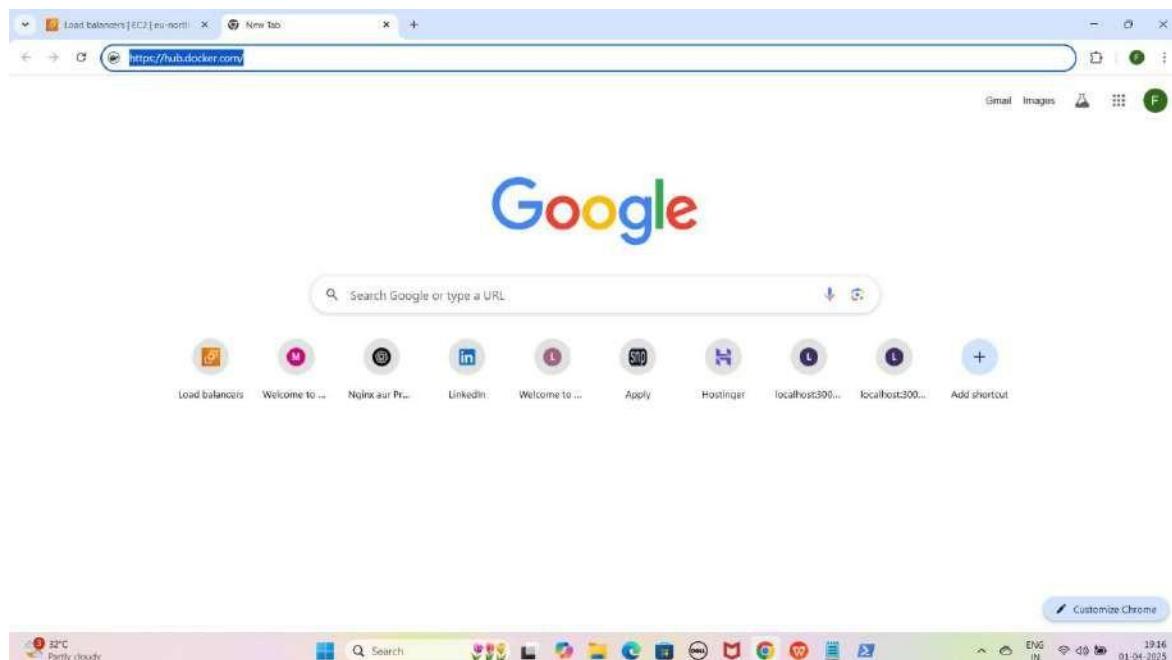
## Step 2: Application Image ki Files and Tools jo hum use karenge

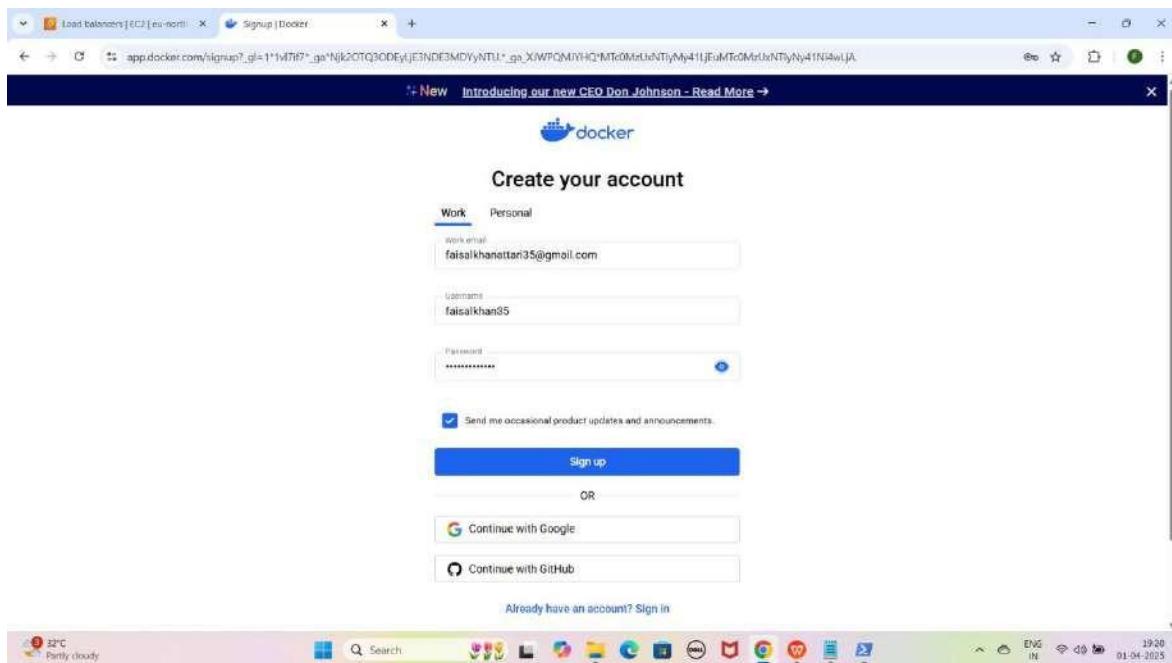
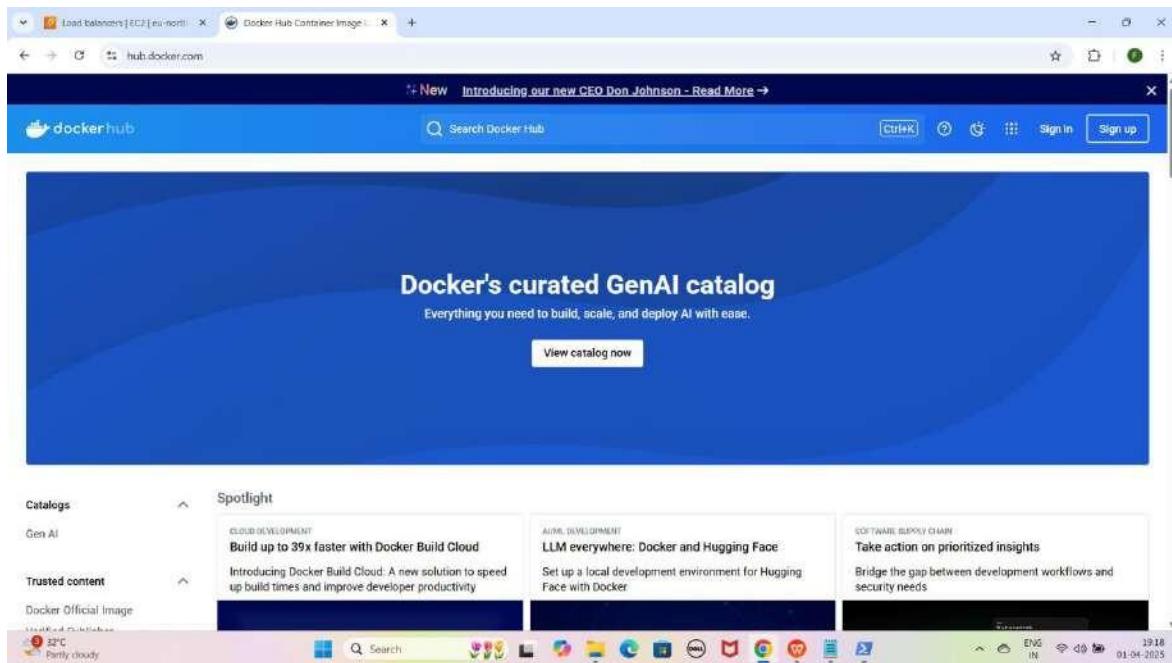
1. Docker Hub
2. Backend-app.js
3. Dockerfile
4. rds-secrets.yaml
5. rds-configmap.yaml
6. service-account.yaml
7. role.yaml
8. rolebinding.yaml
9. Backend-app-pod.yaml

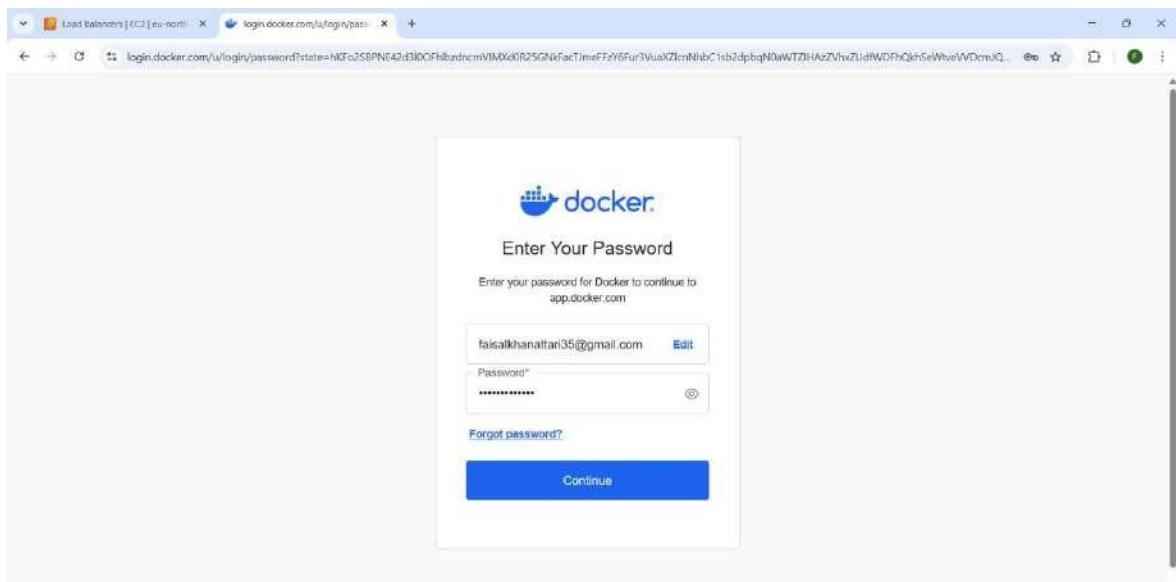
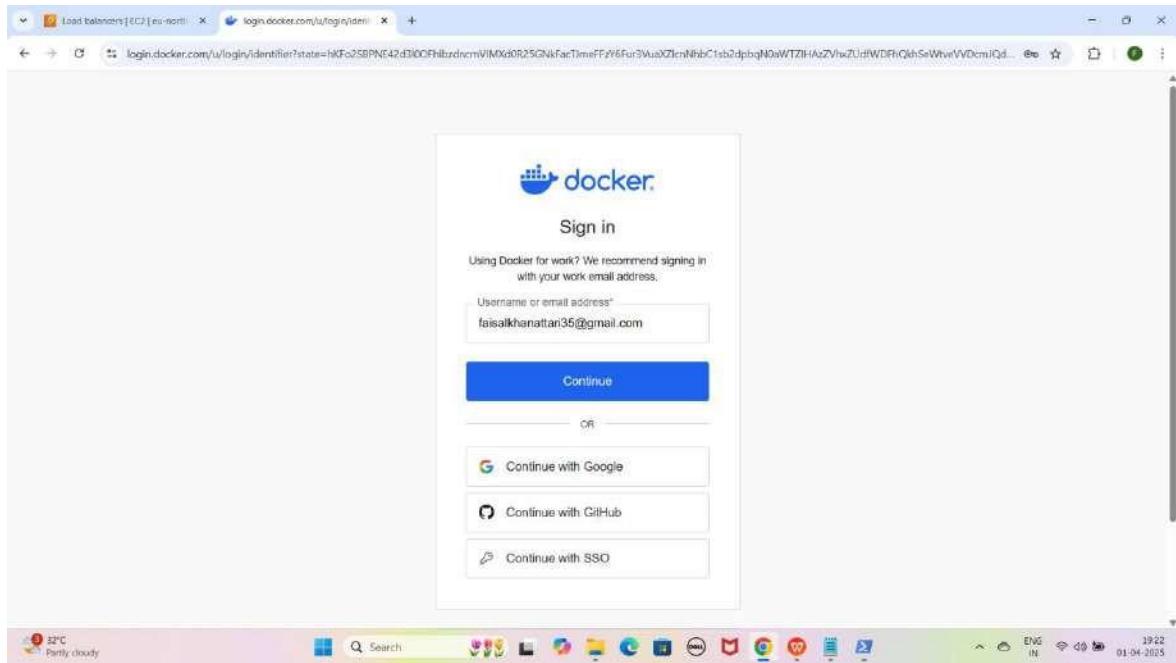
### 1. Docker Hub Account Create Karna

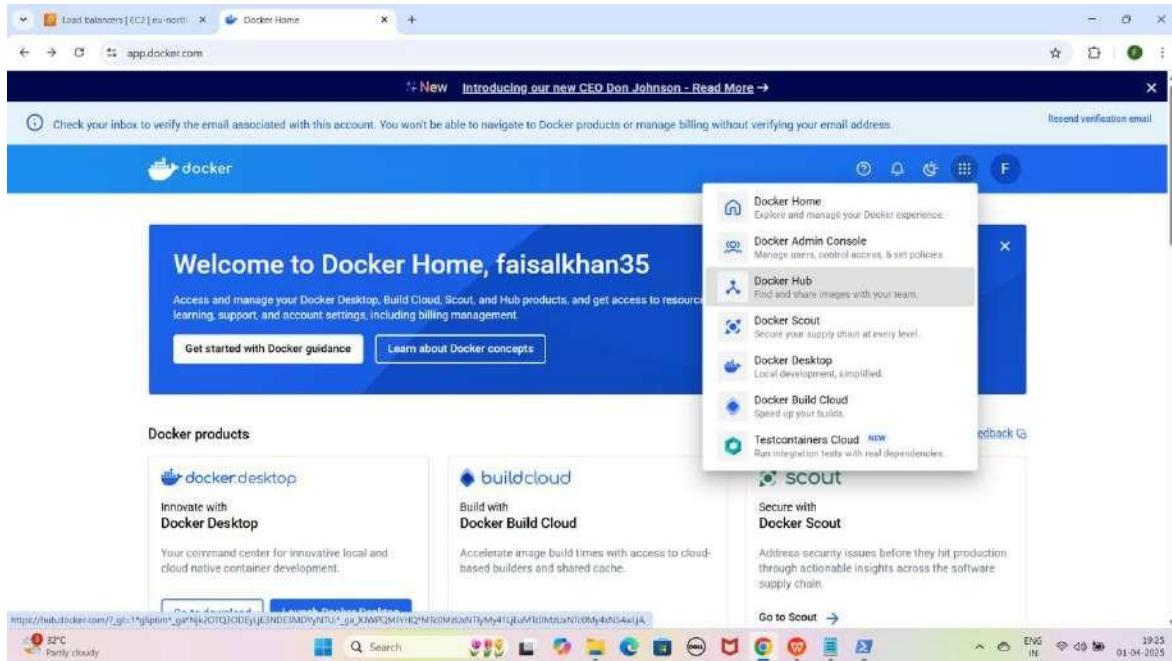
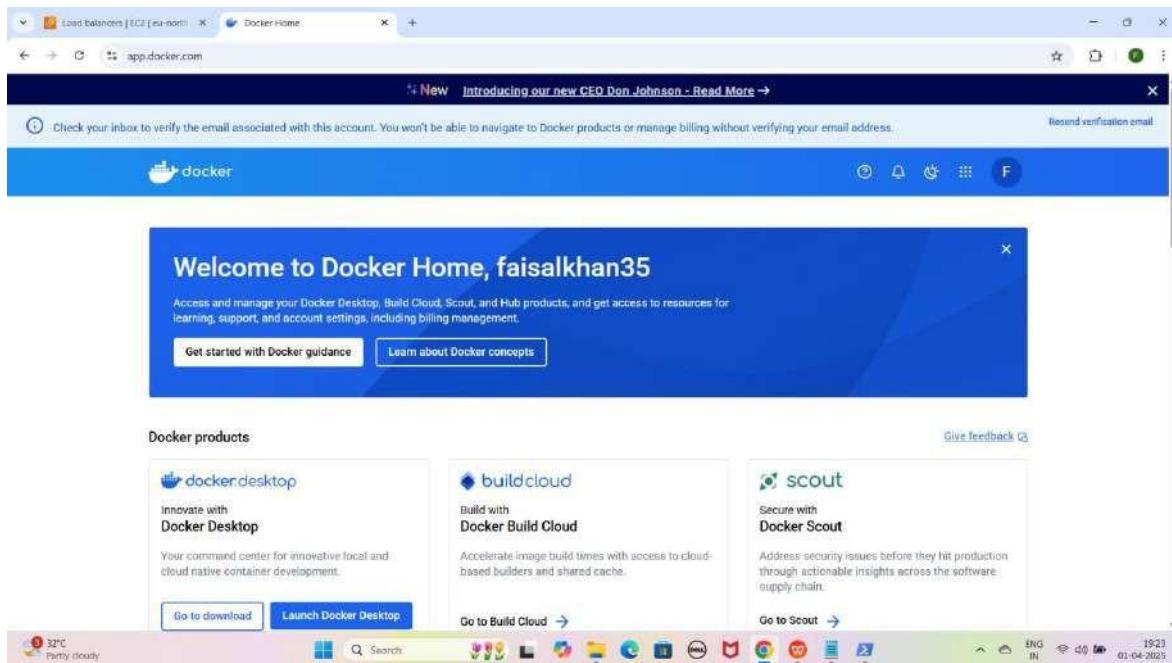
- Docker Hub ki website open karein: <https://hub.docker.com/>
- Sign Up karein aur apna Email, Username and Password set karein.
- Login karein aur Docker Hub Section me jaakar "Create a Repository" par click karein.
- Verify email address kare apna mobile me Gmail open kare and Verify Email Address pe click kare
- Repository ka naam (backend-app) likhein aur visibility option me Public select karein.
- Create button par click karein

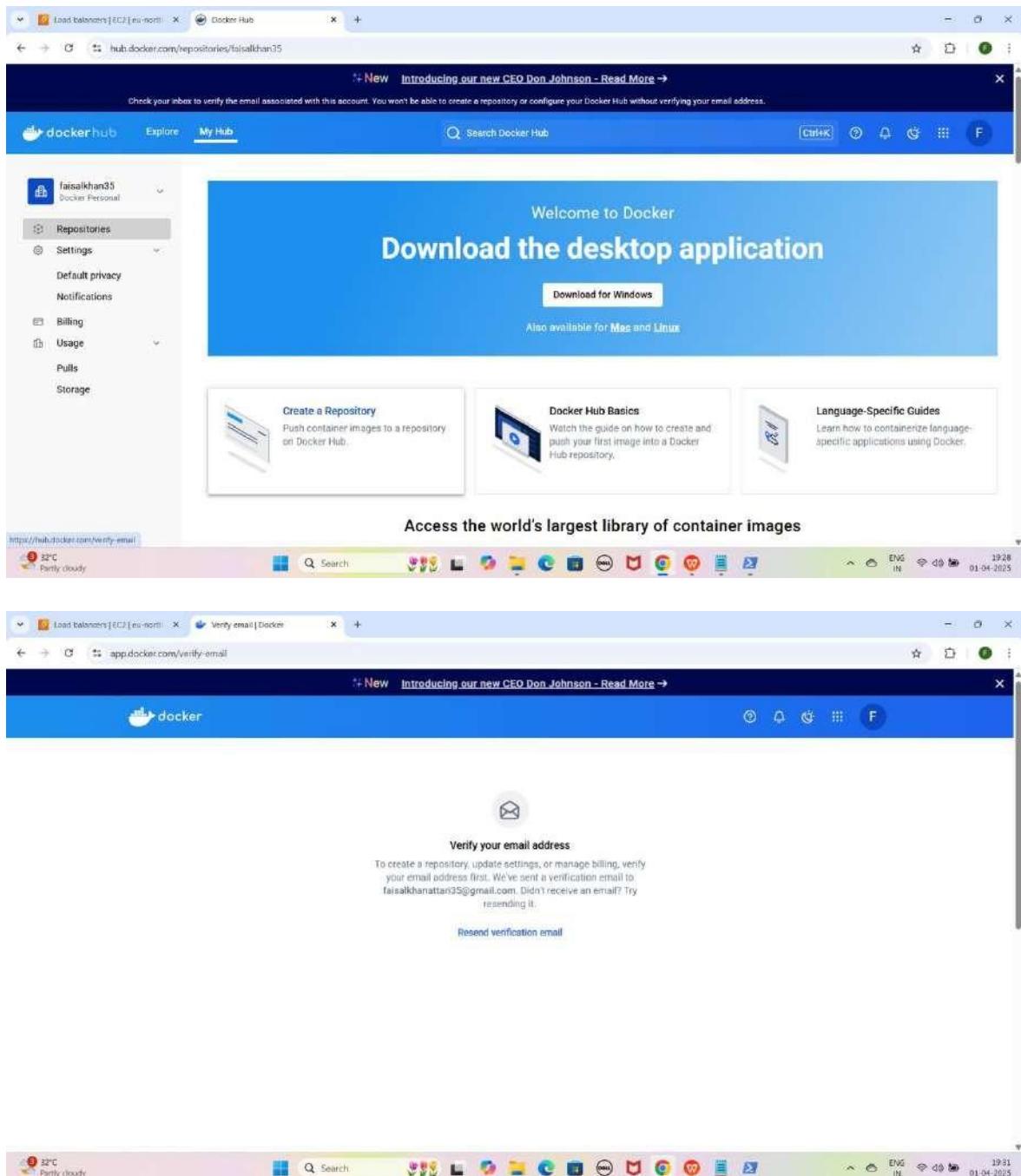
YE KUCH ISTARHA LAGEGA



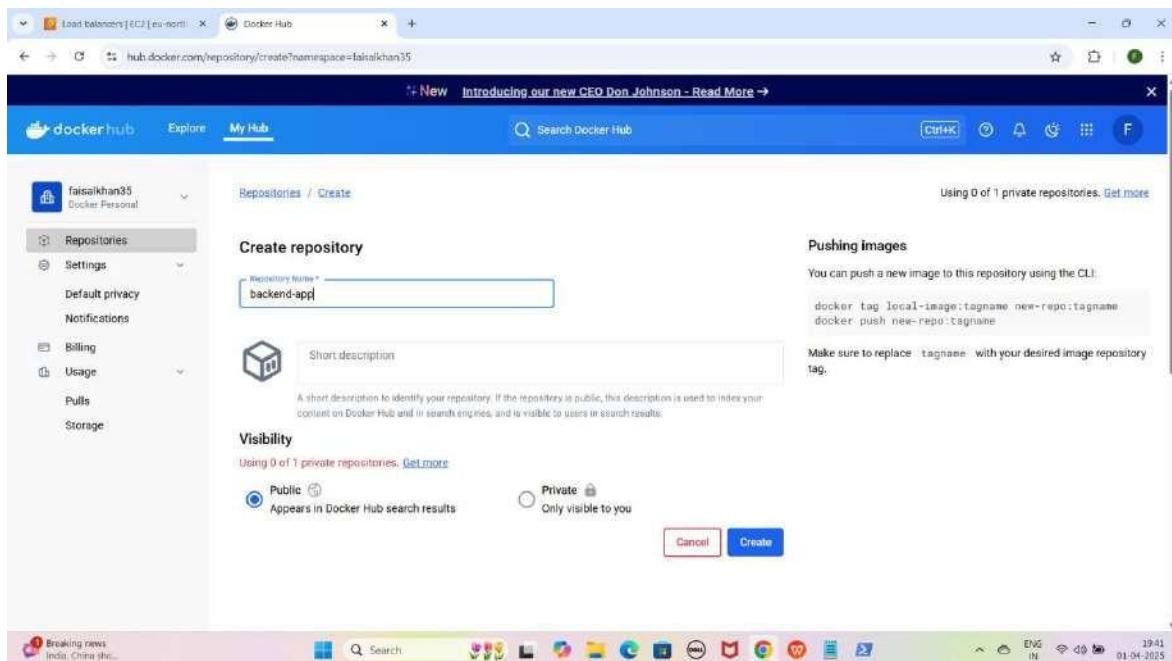
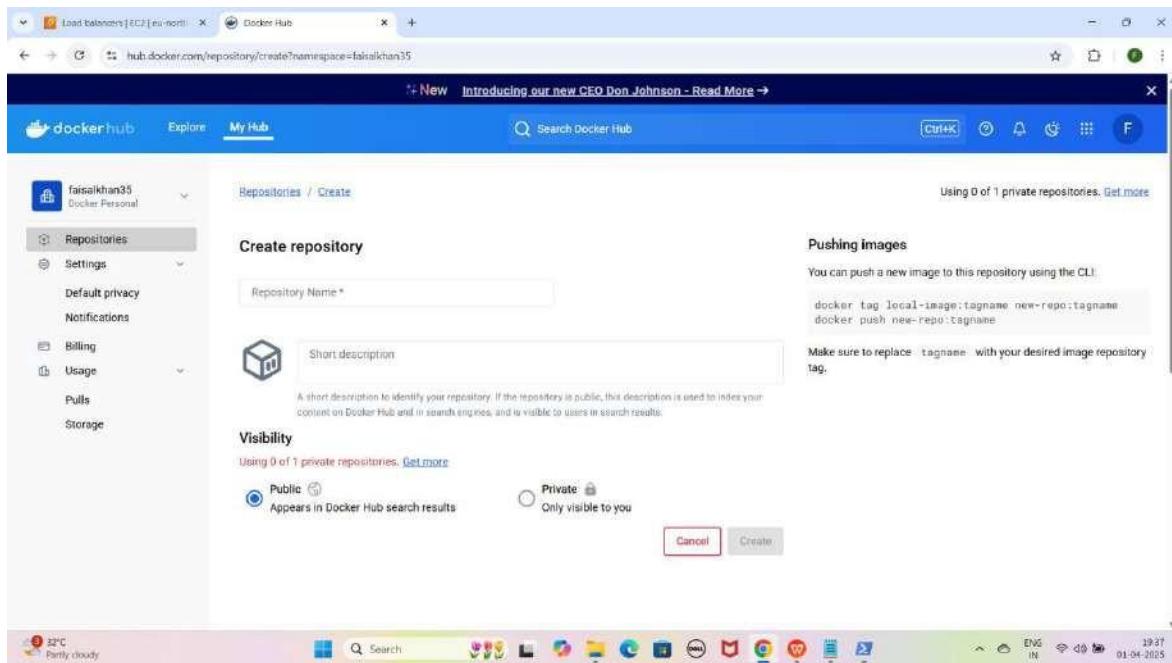


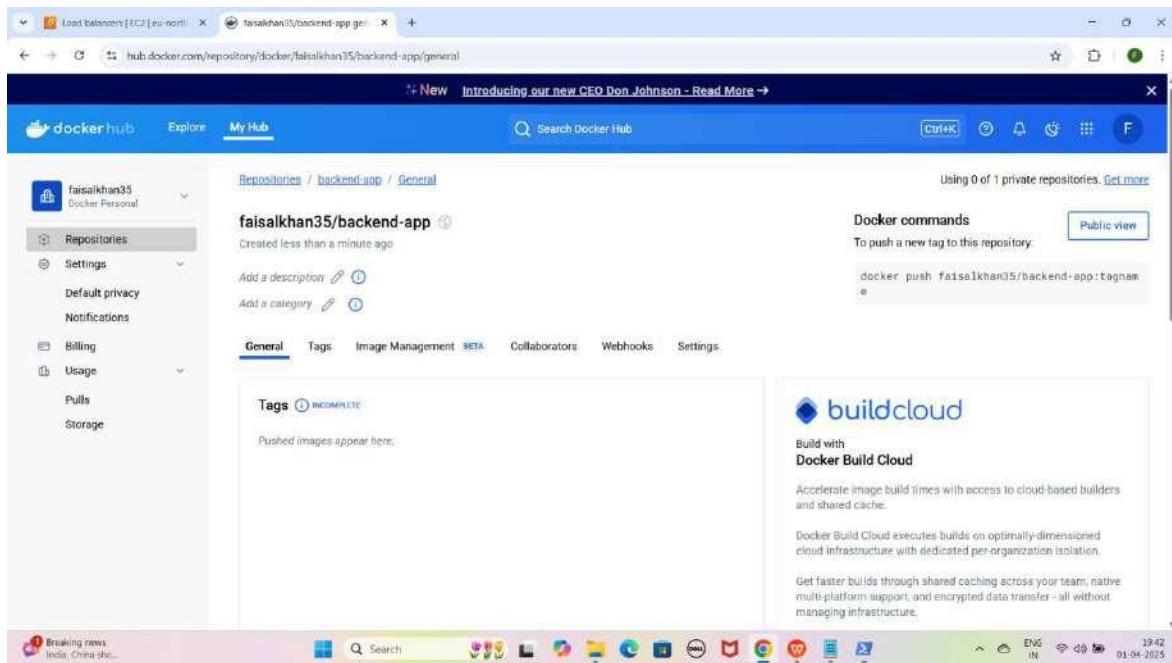






**NOTE: Jaise hi appke Mobile me Email verification successful hoga to page Refresh kare**





## 2. Backend-app.js File Ka Kaam

Yeh ek **JavaScript file** hai jo ek **Node.js application image** banane ke liye use hoti hai.

JAISE KI:-

```
const express = require('express');
const mysql = require('mysql');
const { Client } = require('pg');
const oracledb = require('oracledb');

const app = express();

// MySQL Connection
const mysqlDb =
    mysql.createConnection({ host:
        process.env.MYSQL_HOST,
        user: process.env.MYSQL_USERNAME,
        password: process.env.MYSQL_PASSWORD,
        database: process.env.MYSQL_DATABASE || "faisal-mysql-db"

mysqlDb.connect(err =>
    { if (err) {
        console.error('Error connecting to MySQL:', err.stack);
    } else {
        console.log('Connected to MySQL database');
    }
});

// PostgreSQL Connection
const postgresDb = new Client({
    host: process.env.POSTGRESQL_HOST,
    user: process.env.POSTGRESQL_USERNAME,
    password: process.env.POSTGRESQL_PASSWORD,
    database: process.env.POSTGRESQL_DATABASE || "faisal-postgresql-db",
    port: 5432
});
```

```

});
```

```

postgresDb.connect()
    .then(() => console.log('Connected to PostgreSQL database'))
    .catch(err => console.error('Error connecting to PostgreSQL:', err));
```

```

// Oracle Connection
async function connectOracleDB()
{ try {
    const connection = await
        oracledb.getConnection({ user:
            process.env.ORACLE_USERNAME,
            password: process.env.ORACLE_PASSWORD,
            connectString: process.env.ORACLE_HOST
        });
    console.log('Connected to Oracle database');
    await connection.close();
} catch (err) {
    console.error('Error connecting to Oracle:', err);
}
}
```

```

app.get('/', (req, res) => {
    res.send('Connected to MySQL, PostgreSQL, and Oracle databases!');
});
```

```

app.listen(3000, () => {
    console.log('App is running on http://localhost:3000');
});
```

**NOTE: Ye Script me aap databases change karlo aap jo RDS databases  
create karenge ukne name ke hissab se mere case me mere  
databases ke name ye hai**

1. **faisal-mysql-db**
2. **faisal-oracle-db**
3. **faisal-postgresql-db**

### 3. Dockerfile Ka Kaam

Yeh file **custom application image** create karne ke liye use hoti hai aur isme required **dependencies** aur **packages install** hote hain.

JAISE KI:-

```
# Use Oracle Linux with Instant Client
FROM ghcr.io/oracle/oraclelinux8-instantclient:21

# Upgrade system and install required packages
RUN dnf install -y mysql postgresql \
    && dnf module reset nodejs \
    && dnf module enable nodejs:18 \
    && dnf install -y nodejs \
    && dnf clean all && rm -rf /var/cache/dnf /tmp/* /var/tmp/*

# Set working directory
WORKDIR /app

# Copy package files first
COPY package.json package-lock.json ./

# Install dependencies as root first
RUN npm ci --omit=dev

# Set non-root user for security
RUN useradd -m -u 1000 appuser
RUN chown -R appuser:appuser /app

# Switch to non-root user
USER appuser

# Copy the rest of the application files
COPY --chown=appuser:appuser . .

# Ensure Backend-app.js is executable
RUN chmod +x Backend-app.js

# Expose the application port
EXPOSE 3000

# Start the application
CMD ["node", "Backend-app.js"]
```

## **Short Note:**

Yeh Dockerfile ek **Node.js application** ko containerized environment me run karne ke liye hai, jo **Oracle Linux (Instant Client)** use karta hai.

- **Base Image:** oraclelinux8-instantclient:21 use ho raha hai.
  - **System Upgrade & Installations:** mysql, postgresql, aur nodejs:18 install kar raha hai.
  - **Working Directory:** /app set kiya gaya hai.
  - **Dependencies Installation:** package.json aur package-lock.json ko copy kar ke npm ci run kar raha hai.
  - **Security:** Ek **non-root user** (appuser) create kar ke usko ownership de raha hai.
  - **File Permissions:** Backend-app.js ko executable bana raha hai.
  - **Port Exposure:** 3000 expose kar raha hai.
  - **Startup Command:** node Backend-app.js se application start ho raha hai.

## **1. Docker Image build karo aur Docker Hub me push karo**

```
sudo docker build -t faisalkhan35/backend-app .
```

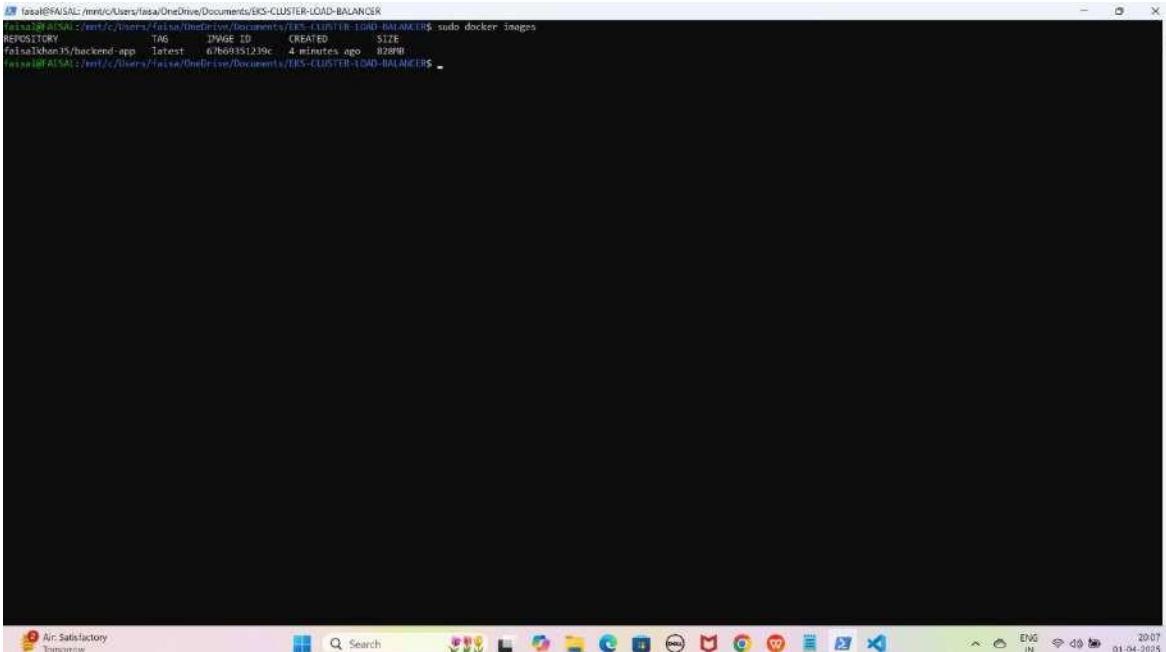
# YE KUCH ISTARHA LAGEA

**NOTE: Aap Docker image ko Docker Hub par upload karne ke liye customize image bana rahe hain, taaki yeh aapke Docker Hub username (faisalkhan35) ke saath upload ho. Agar username include nahi karoge, toh upload karte waqt error aayega.**

## **2. Ab Docker Images check karne ke liye ye command run kariye**

**sudo docker images**

**YE KUCH ISTARHA LAGEGA**



```
faisalkhan35:~$ sudo docker images
REPOSITORY          TAG      IMAGE ID            CREATED             SIZE
Faisalkhan35/backend-app   latest   6fb69351239c   4 minutes ago  828MB
faisalkhan35:~$
```

### 3. Docker Hub me login karne ke liye ye command run kariye

sudo docker login

YE KUCH ISTARHA LAGEGA

```
faisal@FAISAL:~/minic/Users/faisa/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER
faisal@FAISAL:~/minic/Users/faisa/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$ sudo docker login
USING WEB-BASED LOGIN
Info - To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: HCTR-VJMK
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate
Waiting for authentication in the browser...
```

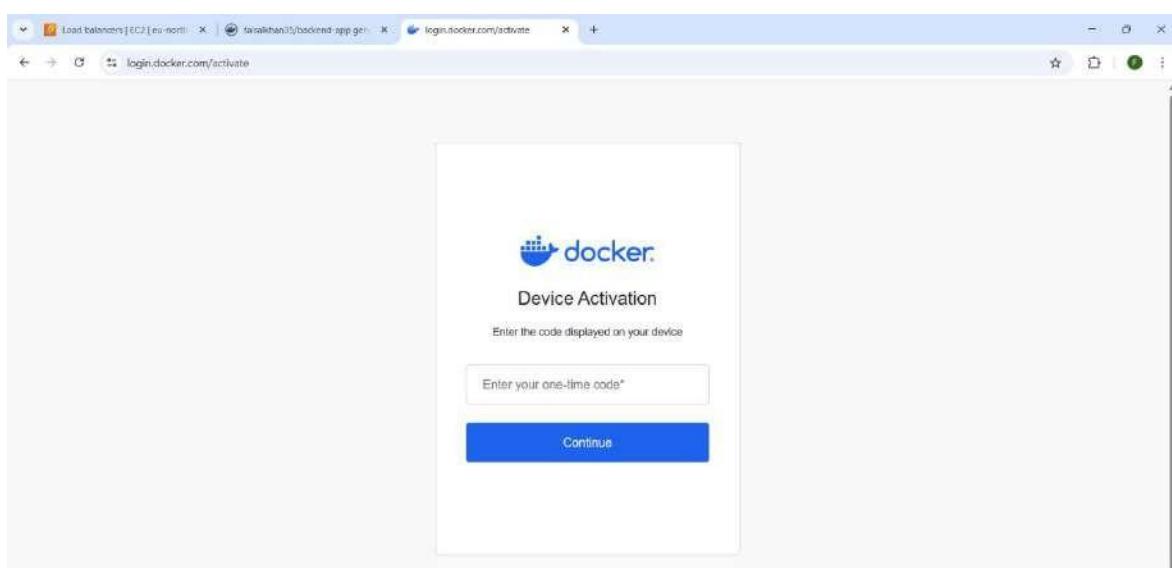
```
Select faisal@FAISAL:~/minic/Users/faisa/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER
faisal@FAISAL:~/minic/Users/faisa/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$ sudo docker login
USING WEB-BASED LOGIN
Info - To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: HCTR-VJMK
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate
Waiting for authentication in the browser...
```



Search Google or type a URL

Load balancers   Welcome to ...   Nginx aur Pr...   App Contain...   LinkedIn   Welcome to ...   Apply   Hostinger   localhost300...   Add shortcut



Enter the code displayed on your device

Enter your one-time code\*

Continue

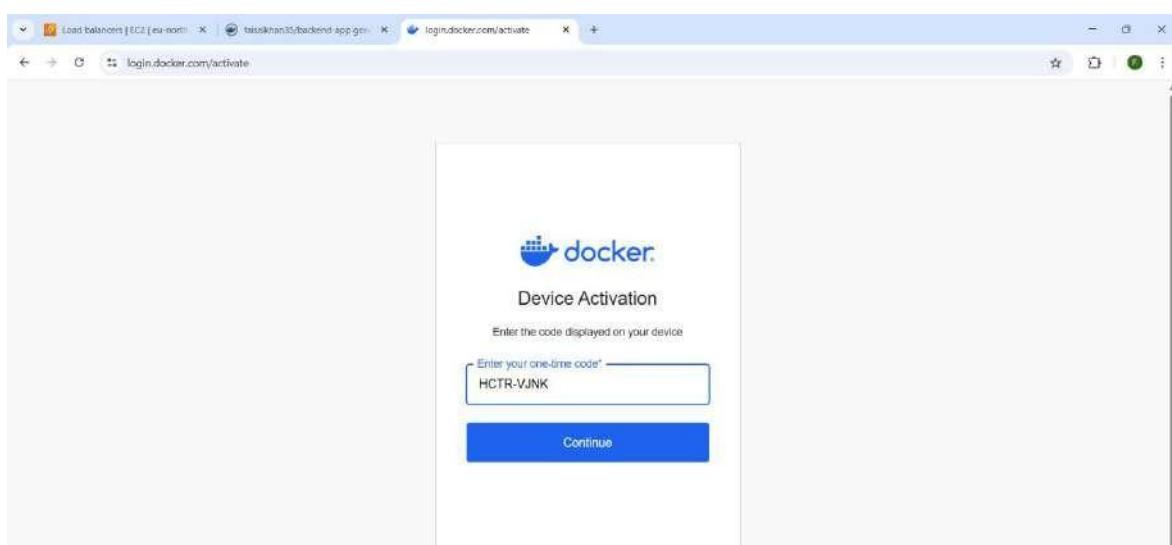


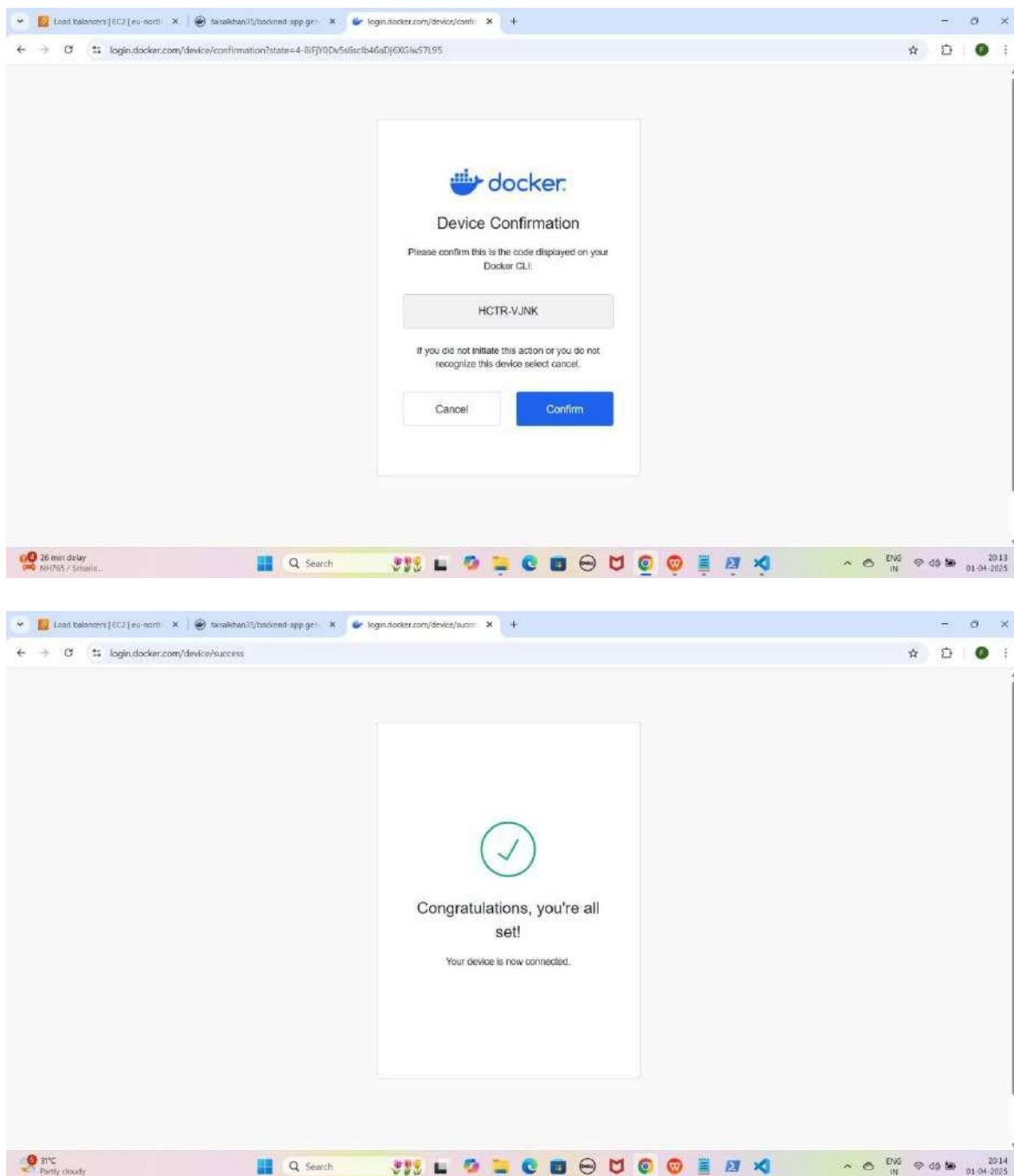
```
PS Select-Kafka@FRI-SAL: /mnt/c/Users/kaisa/OneDrive/Documents/GKS-CLUSTER-LOAD-BALANCER
PS C:\Users\kaisa\OneDrive\Documents\GKS-CLUSTER-LOAD-BALANCER> sudo docker login

USING WEB-BASED LOGIN

Info: to sign in with credentials on the command line, use "docker login -u <username>

Your one-time device confirmation code is: HCTR-VJNK
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate
Waiting for authentication in the browser...
```



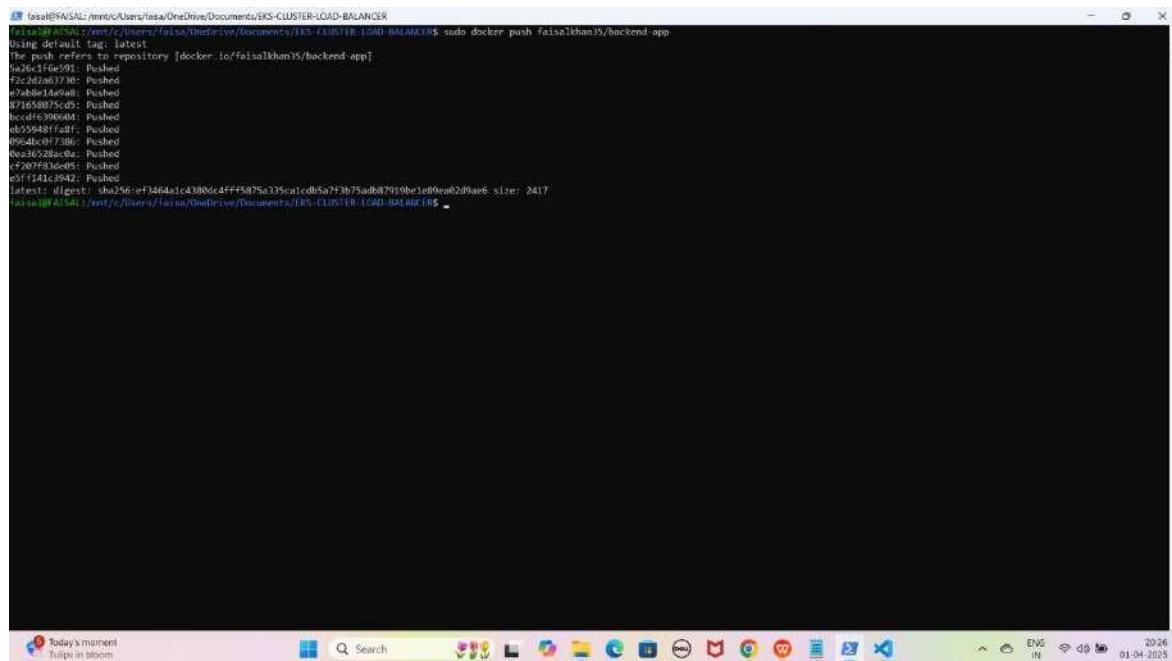


**NOTE : Ye command enter karne ke baad aapko ek URL melega ye kuch istarha ka hoga <https://login.docker.com/activate> is URL ko Browser me Open karna hoga then one-time device confirmation code jo generate howa wo enter karna hoga Code enter karne ke baad Congratulations, You're all set! Your device is now connected karke Show hoga.**

#### 4. Image ko Docker Hub par push karne ke liye ye command run kariye

```
sudo docker push faisalkhan35/backend-app
```

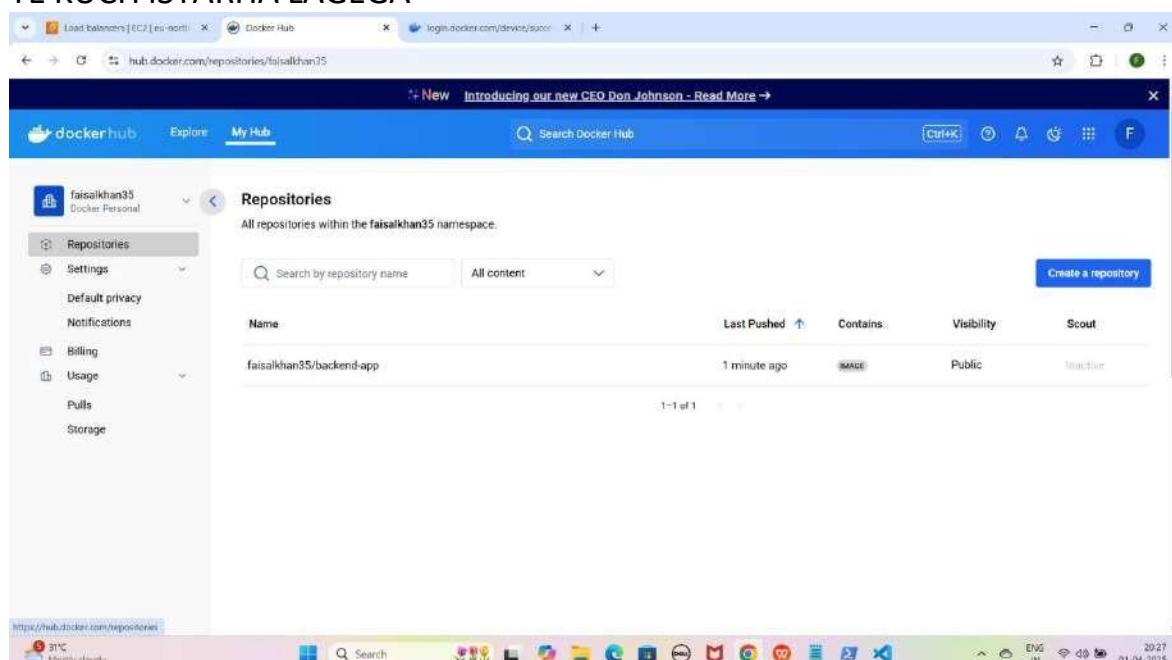
#### YE KUCH ISTARHA LAGEGA



```
faisal@Faisal-Lenovo-M540:~/Desktop$ sudo docker push faisalkhan35/backend-app
Using default tag: latest
The push refers to repository [docker.io/faisalkhan35/backend-app]
f3a6255e6730: Pushed
a7ab8e14d4bb: Pushed
k71658075cd5: Pushed
b0cd1639060d: Pushed
db55948ffea8: Pushed
09640c0f778c: Pushed
0e3d928ac0a: Pushed
4974313942: Pushed
latest: digest: sha256:ef3464a1c43086c4fffa5875a35ca1cd5a7f3b75adb87919be1ed9eaf2d9ae6 size: 2417
faisal@Faisal-Lenovo-M540:~/Desktop$
```

#### 5. Ab Docker Hub me Jaakar check karo ki image push hogai hai ya nahi

#### YE KUCH ISTARHA LAGEGA



The screenshot shows a browser window with three tabs open:

- Load balancers [EC2 | eu-north-1]
- Docker Hub
- hub.docker.com/repositories/faisalkhan35

The main content area is titled "Repositories" and shows the following details for the repository "faisalkhan35/backend-app":

- Name: faisalkhan35/backend-app
- Last Pushed: 1 minute ago
- Contains: IMAGE
- Visibility: Public
- Scout: Function

The sidebar on the left shows account information for "faisalkhan35" (Docker Personal) and links to Settings, Notifications, Billing, Usage, Pulls, and Storage.

**NOTE : Docker Hub par jane ke baad Page Refresh kare aapko aapki Customize Image dikhegi**

#### 4. rds-secrets.yaml File Ka Kaam

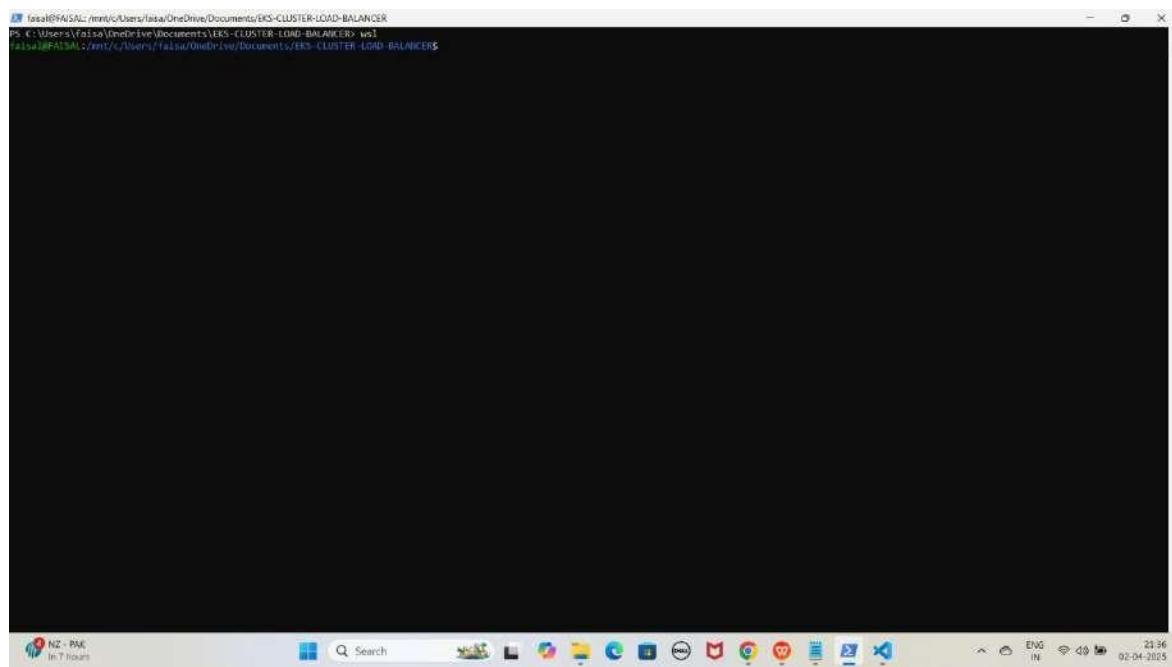
Ye file username aur password ko securely store karne ke liye use hoti hai. Kyunki **Secrets secure hoti hain**, is wajah se hum **Base64 encoding** ka use karke username aur password ko **encrypt** kar dete hain.

**RDS (MYSQL, Oracle, PostgreSQL) ka username aur password encode karna:-**

**1. Windows ke WSL (Windows Subsystem for Linux) ka use kar ke hum Base64 encoding kar sakte hain. wsl type karein aur yeh commands run karein**

wsl

YE KUCH ISTARHA LAGEGA



```
faisal@FASAL:~$ wsl
```

## 2. Username encode karein

- echo -n FaisalKhan | base64

## 3. Password encode karein

- echo -n Faisalkhan35\$| base64

# YE KUCH ISTARHA LAGEGA

```
faisal@F450L: /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$ echo -n FaisalKhan | base64  
RmVsaGhhQkE= | curl -c /Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$ echo -n FaisalKhan3$| base64  
RmVsaGhhQkEz | curl -c /Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$
```

The screenshot shows a terminal window with a dark theme. The title bar reads "Restricted Mode is intended for safe code browsing. This window enables all features." Below the title bar are standard window controls. The main area displays a YAML configuration file:

```
apiversion: V1
kind: Secret
metadata:
  name: rds-secrets
  type: Opaque
data:
  MYSQL_USERNAME: Rnfpc2Fsa2hhbg==      # Base64-encoded username
  MYSQL_PASSWORD: Rnfpc2Fsa2hhbgJH1JA==    # Base64-encoded password
  ORACLE_USERNAME: Rnfpc2Fsa2hhbgJH1JA=
  ORACLE_PASSWORD: Rnfpc2Fsa2hhbgJH1JA=
  POSTGRES_USERNAME: Rnfpc2Fsa2hhbg=
  POSTGRES_PASSWORD: Rnfpc2Fsa2hhbgJH1JA=
```

The bottom of the terminal window shows system status icons and a timestamp: "23:22 02-04-2025".

```
apiVersion: v1
kind: Secret
metadata:
  name: rds-secrets
  type: Opaque
data:
  MYSQL_USERNAME: RmFpc2FsS2hhbg== # Base64 encoded username
  MYSQL_PASSWORD: RmFpc2Fsa2hhbjM1JA== # Base64 encoded password
  ORACLE_USERNAME: RmFpc2FsS2hhbg==
  ORACLE_PASSWORD: RmFpc2Fsa2hhbjM1JA==
  POSTGRESQL_USERNAME: RmFpc2FsS2hhbg==
  POSTGRESQL_PASSWORD: RmFpc2Fsa2hhbjM1JA==
```

**NOTE:** Ye command aapko Base64 encoded values degi, jo hum rds-secrets.yaml file me store karenge. Hum sabhi databases ke liye same Username aur Password use kar rahe hain, isliye encoded values bhi same rahengi. Lekin aapko apne Username aur Password ke hisaab se rds-secrets.yaml file update karni hogi.

JAISE KI:-

```
apiVersion: v1
kind: Secret
metadata:
  name: rds-secrets
  type: Opaque
data:
  MYSQL_USERNAME: RmFpc2FsS2hhbg== # Base64 encoded username
  MYSQL_PASSWORD: RmFpc2Fsa2hhbjM1JA== # Base64 encoded password
  ORACLE_USERNAME: RmFpc2FsS2hhbg==
  ORACLE_PASSWORD: RmFpc2Fsa2hhbjM1JA==
  POSTGRESQL_USERNAME: RmFpc2FsS2hhbg==
  POSTGRESQL_PASSWORD: RmFpc2Fsa2hhbjM1JA==
```

**Agar aap encoded username aur password ko wapas original form me laana chahte hain, toh WSL me yeh commands run karein**

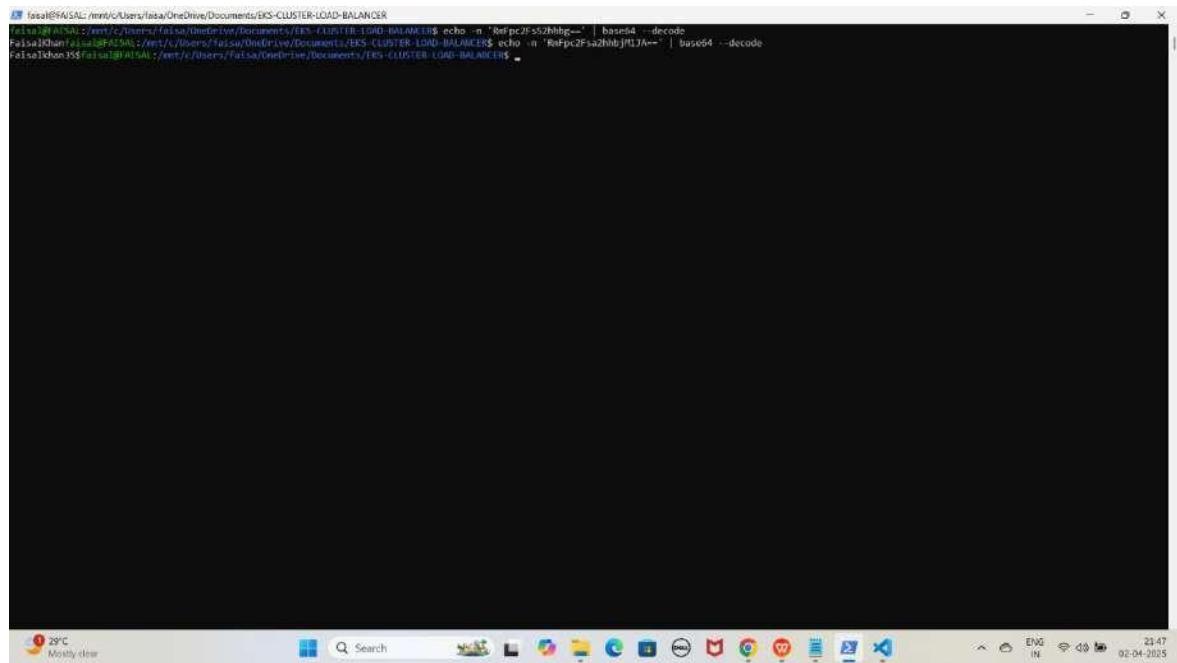
#### **4. Username decode karein**

- echo -n 'RmFpc2FsS2hhbg==' | base64 --decode

#### **5. Password decode karein**

- echo -n 'RmFpc2Fsa2hhbjM1JA==' | base64 --decode

**YE KUCH ISTARHA LAGEGA**



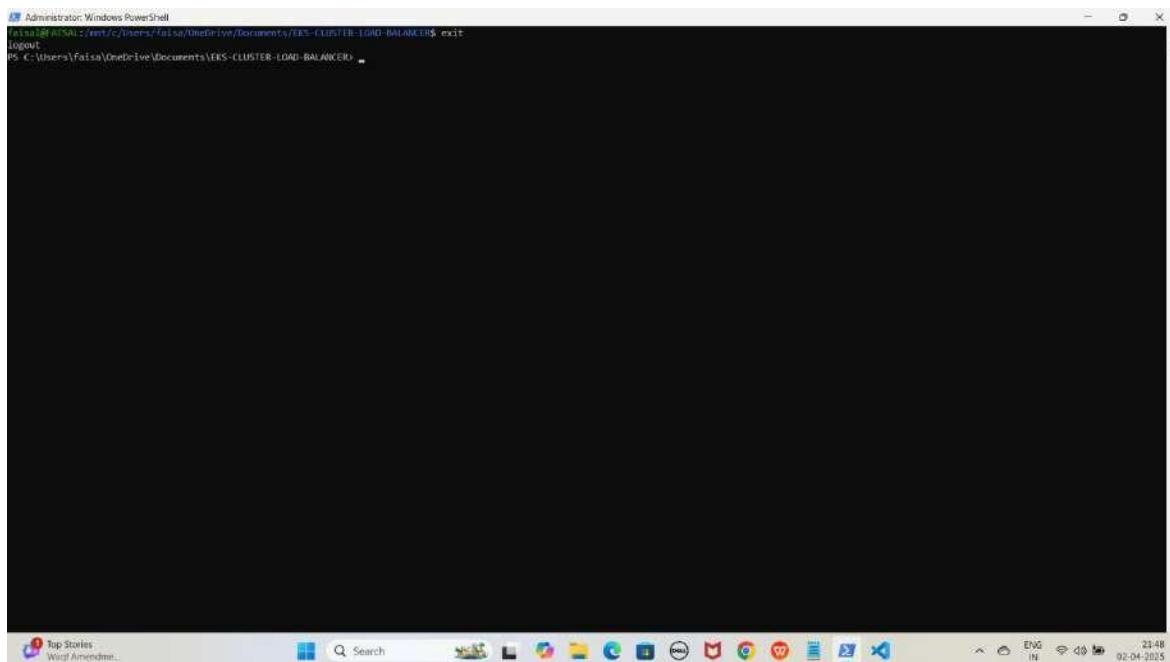
A screenshot of a Windows terminal window titled 'faisal@Faisal: /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER'. The window shows the command 'echo -n 'RmFpc2Fsa2hhbjM1JA==' | base64 --decode' being run. The output of the command is visible at the bottom of the terminal window.

```
faisal@Faisal: /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$ echo -n 'RmFpc2Fsa2hhbjM1JA==' | base64 --decode
Faisal@Faisal: /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$ echo -n 'RmFpc2Fsa2hhbjM1JA==' | base64 --decode
Faisal@Faisal: /mnt/c/Users/faisal/OneDrive/Documents/EKS-CLUSTER-LOAD-BALANCER$
```

## 6. Ab WSL se exit hogao exit hone ke liye exit type karein

exit

YE KUCH ISTARHA LAGEGA

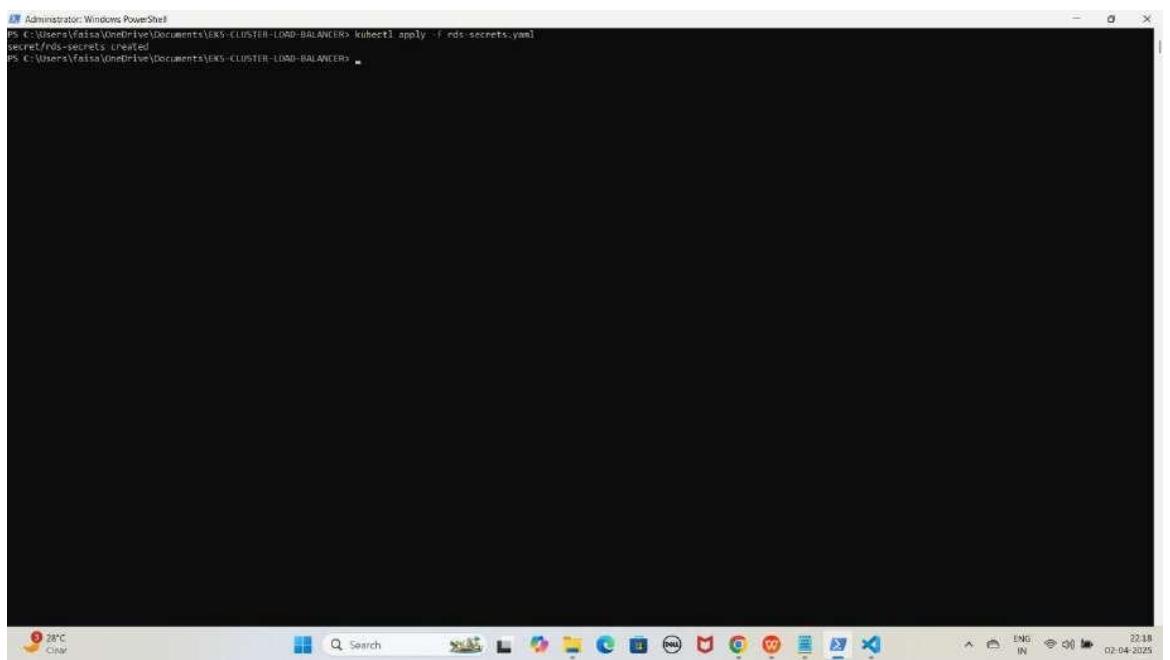


```
Faisal@Faisal-OptiPlex-5090: ~ % exit
PS C:\Users\Faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## Secret Apply Karo

kubectl apply -f rds-secrets.yaml

YE KUCH ISTARHA LAGEGA



```
Faisal@Faisal-OptiPlex-5090: ~ % kubectl apply -f rds-secrets.yaml
rds-secrets created
PS C:\Users\Faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## 5. rds-configmap.yaml File Ka Kaam

Yeh file non-sensitive data store karne ke liye use hoti hai, jaise

- Database ka naam
- Host ka naam (hostname)
- Port number

Aapko apne databases ke names and endpoints ko rds-configmap.yaml file me update karna hoga

YE KUCH ISTARHA LAGEGA

The screenshot shows the AWS Aurora and RDS Databases page. On the left, there's a sidebar with options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, and Event subscriptions. The main area is titled "Databases (3)" and lists three entries:

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations
faisal-mysql-db	Available	Instance	MySQL Co...	eu-north-1a	db.t4g.micro	
faisal-oracle-db	Available	Instance	Oracle En...	eu-north-1a	db.m6i.large	
faisal-postgresql-db	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro	

The screenshot shows the AWS Aurora and RDS Database Details page for the "faisal-mysql-db" instance. The left sidebar is identical to the previous screenshot. The main area has tabs for Summary, Connectivity & security, Monitoring, Logs & events, Configuration, Zero-ETL integrations, Maintenance & backups, and Data. The "Connectivity & security" tab is selected. It shows the endpoint (faisal-mysql-db.qlsbywwgsw.eu-north-1.rds.amazonaws.com), port (3306), and networking details (Availability Zone: eu-north-1a, VPC: vpc-0e345f19effbe822, Subnet group: faisal-subnet). The security section shows VPC security groups (ALL-TRAFFIC-ALLOW [sg-0ea8265816a26be1c]), which is active. The public accessibility is set to No, and the certificate authority is listed as RSA-2048-q1.

The screenshot shows a code editor interface with several tabs open. The main tab displays a Dockerfile:

```
version: '1
services:
  rds-configmap:
    metadata:
      name: rds-configmap
    data:
      MYSQL_DATABASE: "faisal-mysql-db"
      MYSQL_HOST: "faisal-mysql-db.ellisbywgd.eu-north-1.rds.amazonaws.com" # Replace with your RDS MySQL Endpoint
      MYSQL_PORT: "3306"
      ORACLE_DATABASE: "faisal-oracle-db"
      ORACLE_HOST: "faisal-oracle-db.ellisbywgd.eu-north-1.rds.amazonaws.com" # Replace with your RDS Oracle Endpoint
      ORACLE_PORT: "1521"
      POSTGRESQL_DATABASE: "/faisal-postgresql-db"
      POSTGRESQL_HOST: "faisal-postgresql-db.ellisbywgd.eu-north-1.rds.amazonaws.com" # Replace with your RDS PostgreSQL Endpoint
      POSTGRESQL_PORT: "5432"
```

A sidebar on the left shows a file tree with files like `Backend/app.yaml` and `rds-configmap.yaml`. A status bar at the bottom indicates "restricted Mode".

The screenshot shows the AWS RDS console under the "Aurora and RDS" section. The left sidebar lists options like Dashboard, Databases, Query Editor, and Performance insights. The main area displays a table of databases:

DB identifier	Status	Role	Engine	Region	Size	Recommendations
faisal-mysql-db	Available	Instance	MySQL Co...	eu-north-1a	db.t4g.micro	
faisal-oracle-db	Available	Instance	Oracle En...	eu-north-1a	db.m6i.xlarge	
faisal-postgresql-db	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro	

The browser address bar shows the URL: `eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases`.

The screenshot shows the AWS RDS console for the 'faisal-oracle-db' database. The 'Summary' section indicates the database is available, running Oracle Enterprise Edition in the eu-north-1 region. The 'Connectivity & security' tab is selected, showing the endpoint copied to 'faisal-oracle-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com' on port 1521. The 'Networking' section shows the availability zone as eu-north-1a and the VPC as vpc-0e345f19effbea822. The 'Security' section lists the VPC security group as ALL-TRAFFIC-ALLOW (sg-0eaa265816a2bbed) and it is active. The 'Publicly accessible' setting is off.

The screenshot shows a Dockerfile in a code editor. The file defines configurations for MySQL, Oracle, and PostgreSQL databases. It sets the MySQL database to 'faisal-mysql-db', the MySQL host to 'faisal-mysql-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com' (with a note to replace with the RDS MySQL endpoint), and the MySQL port to 3306. It also sets the Oracle database to 'faisal-oracle-db', the Oracle host to 'faisal-oracle-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com' (with a note to replace with the RDS Oracle endpoint), and the Oracle port to 1521. Finally, it sets the PostgreSQL database to 'faisal-postgresql-db', the PostgreSQL host to 'faisal-postgresql-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com' (with a note to replace with the RDS PostgreSQL endpoint), and the PostgreSQL port to 5432.

```
1  aptVersion: V1
2  kml: configup
3  metadata:
4    name: rds-configup
5  data:
6    MYSQL_DATABASE: "faisal-mysql-db"
7    MYSQL_HOST: "faisal-mysql-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com" # Replace with your RDS MySQL Endpoint
8    MYSQL_PORT: "3306"
9
10   ORACLE_DATABASE: "faisal-oracle-db"
11   ORACLE_HOST: "faisal-oracle-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com" # Replace with your RDS Oracle Endpoint
12   ORACLE_PORT: "1521"
13
14   POSTGRESOL_DATABASE: "faisal-postgresql-db"
15   POSTGRESOL_HOST: "faisal-postgresql-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com" # Replace with your RDS PostgreSQL Endpoint
16   POSTGRESOL_PORT: "5432"
```

Screenshot of the AWS RDS console showing the 'Databases' page. The left sidebar shows 'Aurora and RDS' with 'Databases' selected. The main area displays three databases: 'faisal-mysql-db' (MySQL), 'faisal-oracle-db' (Oracle), and 'faisal-postgresql-db' (PostgreSQL). Each database row includes columns for DB identifier, Status, Role, Engine, Region, and Size.

DB identifier	Status	Role	Engine	Region	Size
faisal-mysql-db	Available	Instance	MySQL Co...	eu-north-1a	db.t4g.micro
faisal-oracle-db	Available	Instance	Oracle En...	eu-north-1a	db.m6i.large
faisal-postgresql-db	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro

Screenshot of the AWS RDS console showing the 'Database Details' page for 'faisal-postgresql-db'. The left sidebar shows 'Aurora and RDS' with 'Databases' selected. The main area displays the 'Summary' section and the 'Connectivity & security' tab. In the 'Connectivity & security' tab, there is a note: 'Endpoint copied' followed by a link: 'faisal-postgresql-db.clsBywwg5wd.eu-north-1.rds.amazonaws.com'. Other sections include 'Networking' (Availability Zone: eu-north-1a, VPC: vpc-0e345f19effbra622, Subnet group: faisal-subnet, Subnets), 'Security' (VPC security groups: ALL-TRAFFIC ALLOW (sg-0ee8265b16a26bed), Active), and 'Publicly accessible' (No).

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rds-configmap
data:
  MYSQL_DATABASE: "faisal-mysql-db"
  MYSQL_HOST: faisal-mysql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com      # Replace with your RDS MySQL Endpoint
  MYSQL_PORT: "3306"

  ORACLE_DATABASE: "faisal-oracle-db"
  ORACLE_HOST: faisal-oracle-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com      # Replace with your RDS Oracle Endpoint
  ORACLE_PORT: "1521"

  POSTGRESQL_DATABASE: "faisal-postgresql-db"
  POSTGRESQL_HOST: faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com      # Replace with your RDS PostgreSQL Endpoint
  POSTGRESQL_PORT: "5432"
```

## JAISE KI:-

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: rds-configmap
data:
  MYSQL_DATABASE: "faisal-mysql-db"
  MYSQL_HOST: faisal-mysql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com      # Replace with your RDS MySQL Endpoint
  MYSQL_PORT: "3306"

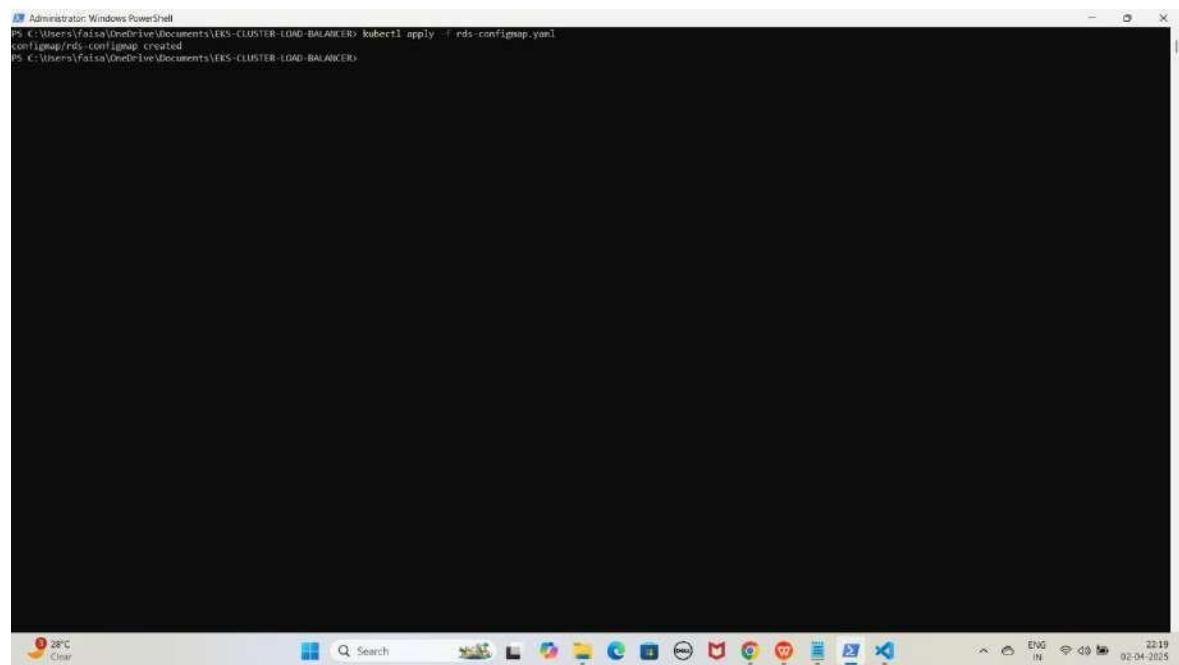
  ORACLE_DATABASE: "faisal-oracle-db"
  ORACLE_HOST: faisal-oracle-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com      # Replace with your RDS Oracle Endpoint
  ORACLE_PORT: "1521"

  POSTGRESQL_DATABASE: "faisal-postgresql-db"
  POSTGRESQL_HOST: faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com      # Replace with your RDS PostgreSQL Endpoint
  POSTGRESQL_PORT: "5432"
```

## ConfigMap Apply Karo

```
kubectl apply -f rds-configmap.yaml
```

## YE KUCH ISTARHA LAGEGA



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command entered is "kubectl apply -f rds-configmap.yaml". The output shows "rds-configmap created". The window is positioned over a desktop background featuring a dark theme with a weather widget in the bottom-left corner displaying "28°C Clear". The taskbar at the bottom of the screen is visible, showing various pinned icons and the system tray.

```
Administrator: Windows PowerShell
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f rds-configmap.yaml
rds-configmap created
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## 6. service-account.yaml File Ka Kaam

Yeh **file Secrets aur ConfigMaps** ko access karne ke liye **permissions** dene ke liye use hoti hai. Isko Kubernetes **Service Account** banane ke liye likha jata hai.

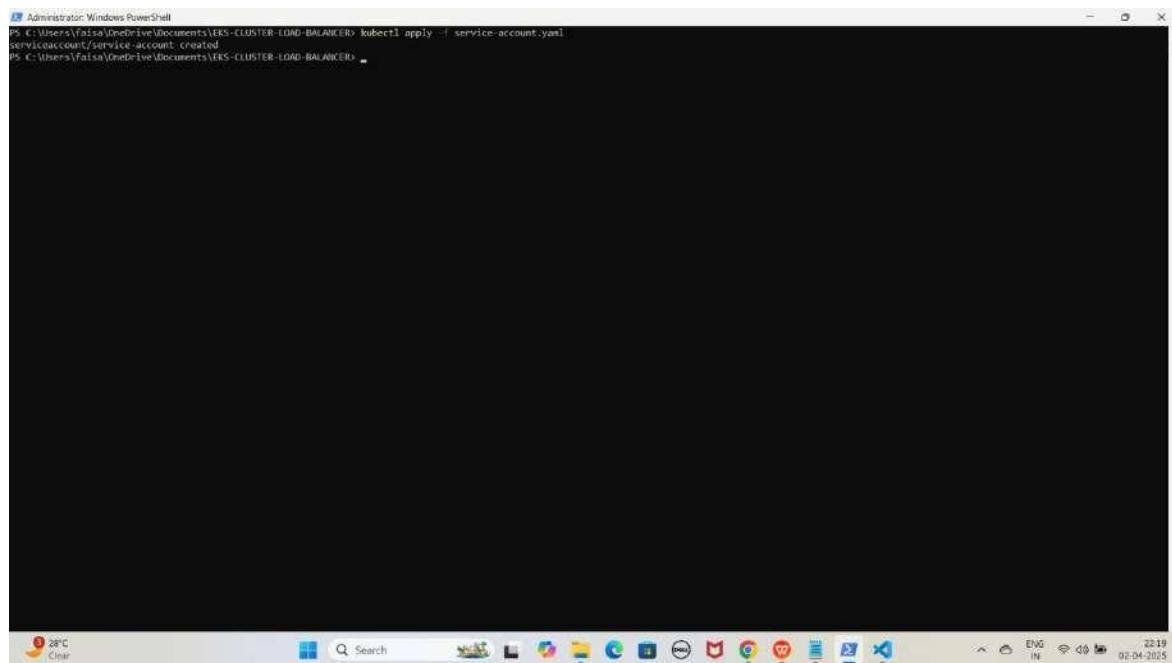
JAISE KI:-

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: service-account
```

## Service Account Apply Karo

kubectl apply -f service-account.yaml

YE KUCH ISTARHA LAGEGA



```
[PS] Administration: Windows PowerShell
PS C:\Users\Yashas\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f service-account.yaml
serviceaccount/service-account created
PS C:\Users\Yashas\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## 7. role.yaml File Ka Kaam

Yeh file **Secrets** aur **ConfigMaps** ke liye **read-only permissions** define karne ke liye use karrahe hai. Isko Kubernetes **Role** banane ke liye likha jata hai.

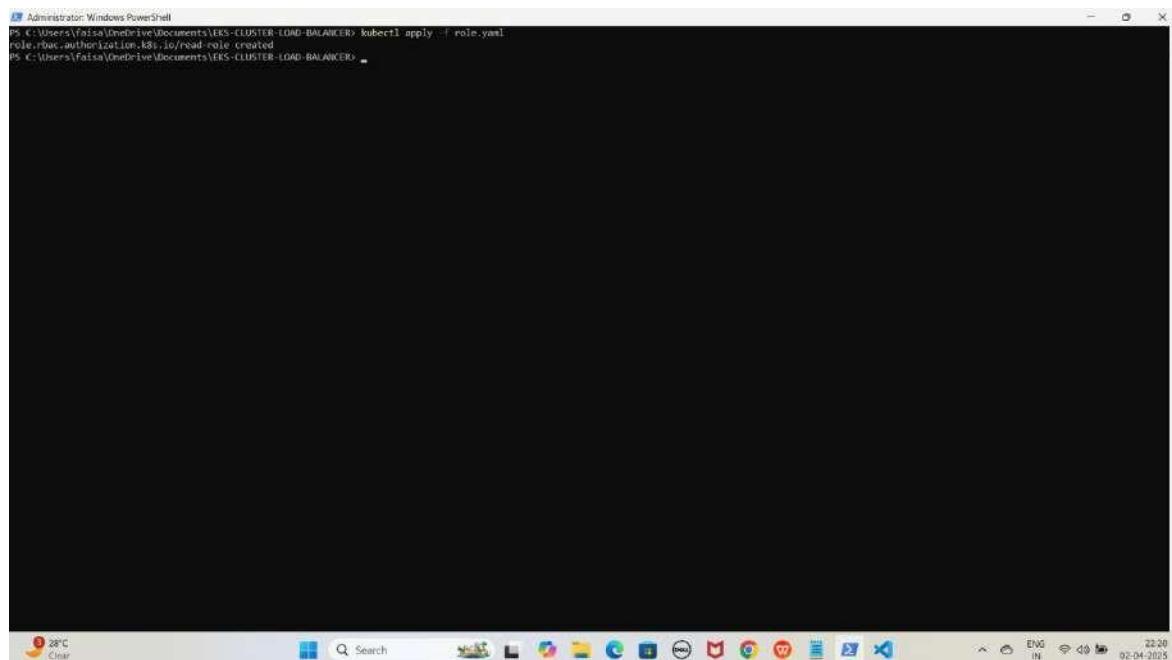
JAISE KI:-

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: read-role
rules:
  - apiGroups: [""]
    resources: ["secrets", "configmaps"]
    verbs: ["get", "list"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["get", "list"]
  - apiGroups: [""]
    resources: ["endpoints"]
    verbs: ["get", "list"]
```

## Role Apply Karo

```
kubectl apply -f role.yaml
```

YE KUCH ISTARHA LAGEGA



```
[Administrator: Windows PowerShell
PS C:\Users\Fatima\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f role.yaml
role.rbac.authorization.k8s.io/read-role created
PS C:\Users\Fatima\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## 8. rolebinding.yaml File Ka Kaam

Yeh file Service Account ko Secrets aur ConfigMaps ke liye **read-only permissions** assign karne ke liye use karrahe hai. Isko **Kubernetes RoleBinding** banane ke liye likha jata hai.

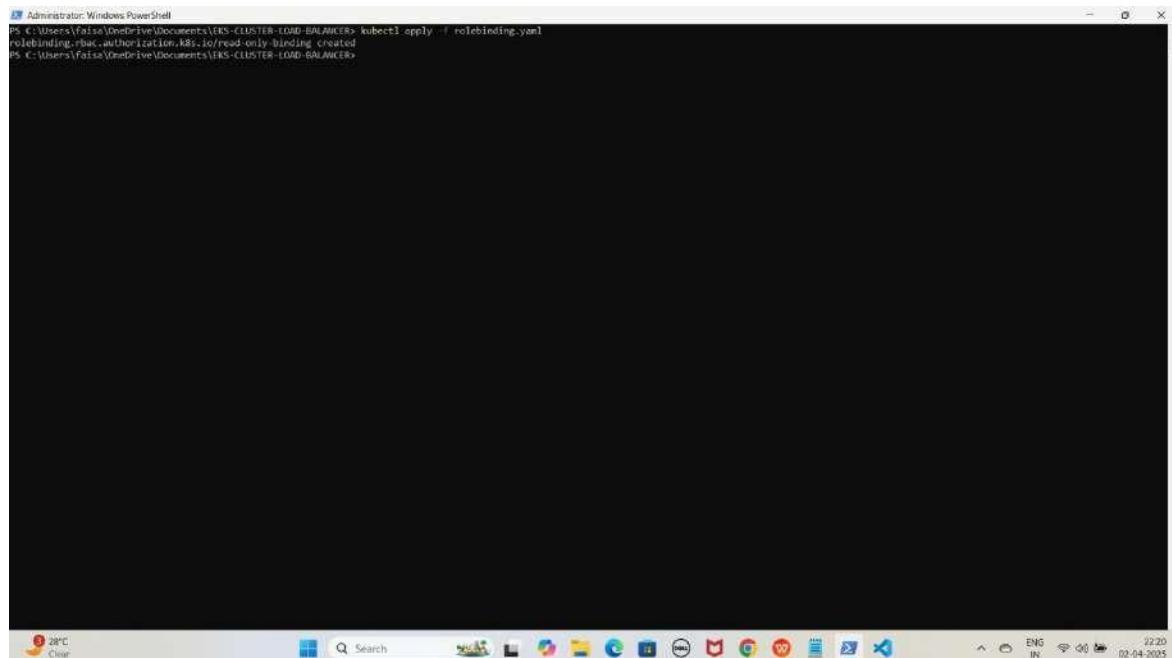
JAISE KI:-

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: read-only-binding
subjects:
  - kind: ServiceAccount
    name: service-account
    namespace: default
roleRef:
  kind: Role
  name: read-role
  apiGroup: rbac.authorization.k8s.io
```

## RoleBinding Apply Karo

kubectl apply -f rolebinding.yaml

## YE KUCH ISTARHA LAGEGA



```
Administrator: Windows PowerShell
PS C:\Users\Faiza\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER> kubectl apply -f rolebinding.yaml
rolebinding.rbac.authorization.k8s.io/read-only-binding created
PS C:\Users\Faiza\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER>
```

## 9. Backend-app-pod.yaml File Ka Kaam

Yeh file Docker Hub par hosted application image jo humne create kiya tha uska use karke ek pod create karne ke liye use karrahe hai.

JAISE KI:-

```
apiVersion: v1
kind: Pod
metadata:
  name: backend-app-pod
spec:
  containers:
    - name: app-container
      image: faisalkhan45/backend-app
      env:
        - name: MYSQL_HOST
          valueFrom:
            configMapKeyRef:
              name: rds-configmap
              key: MYSQL_HOST
        - name: MYSQL_USERNAME
          valueFrom:
            secretKeyRef:
              name: rds-secrets
              key: MYSQL_USERNAME
        - name: MYSQL_PASSWORD
          valueFrom:
            secretKeyRef:
              name: rds-secrets
              key: MYSQL_PASSWORD

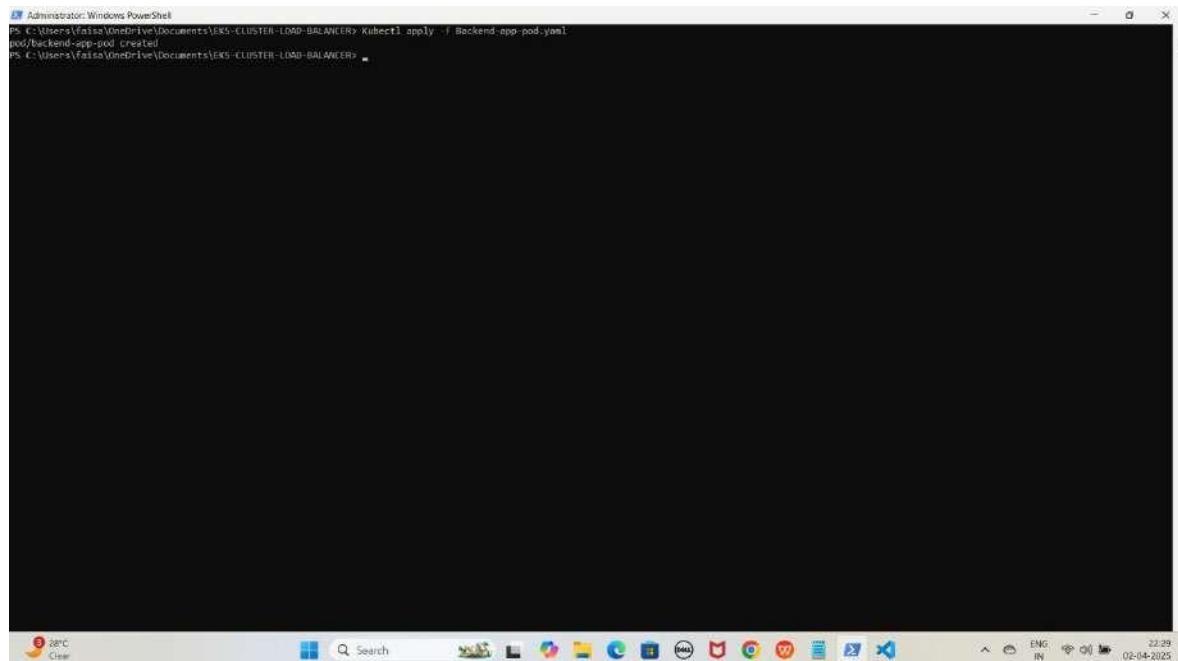
        - name: ORACLE_HOST
          valueFrom:
            configMapKeyRef:
              name: rds-configmap
              key: ORACLE_HOST
        - name: ORACLE_USERNAME
          valueFrom:
            secretKeyRef:
              name: rds-secrets
              key: ORACLE_USERNAME
        - name: ORACLE_PASSWORD
          valueFrom:
            secretKeyRef:
              name: rds-secrets
              key: ORACLE_PASSWORD

        - name: POSTGRESQL_HOST
          valueFrom:
            configMapKeyRef:
              name: rds-configmap
              key: POSTGRESQL_HOST
        - name: POSTGRESQL_USERNAME
          valueFrom:
            secretKeyRef:
              name: rds-secrets
              key: POSTGRESQL_USERNAME
        - name: POSTGRESQL_PASSWORD
          valueFrom:
            secretKeyRef:
              name: rds-secrets
              key: POSTGRESQL_PASSWORD
```

## Backend App Pod Apply Karo

Kubectl apply -f Backend-app-pod.yaml

YE KUCH ISTARHA LAGEGA

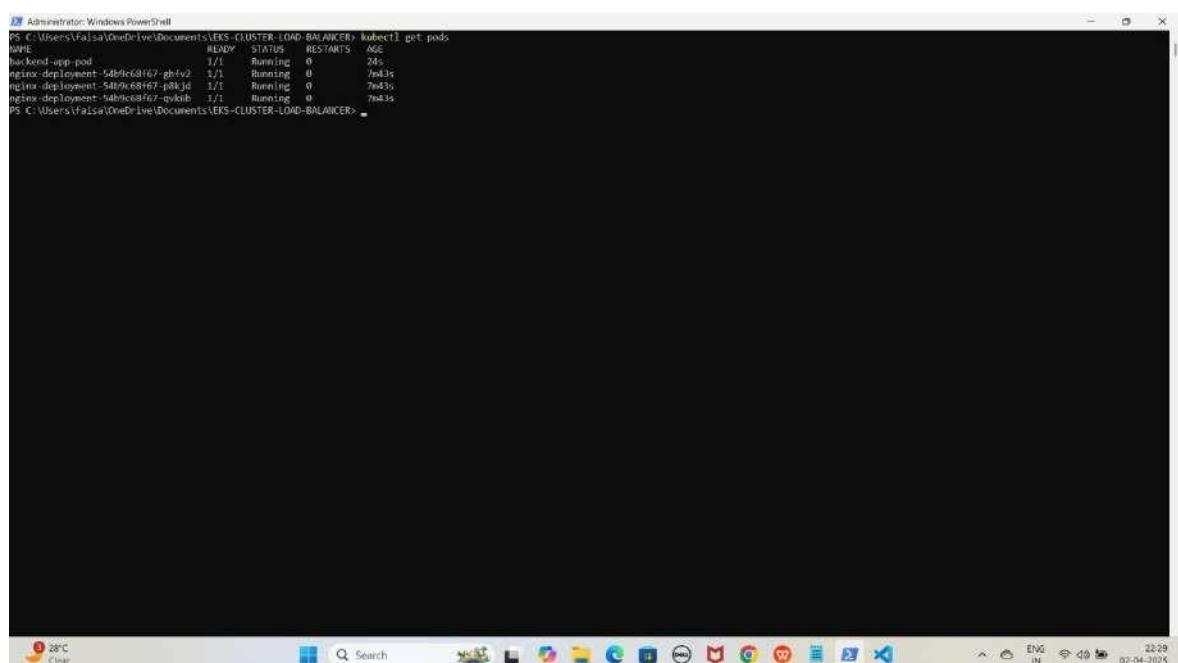


```
[Administrator: Windows PowerShell]
PS C:\Users\Faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f Backend-app-pod.yaml
pod/app-pod created
PS C:\Users\Faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## Pods check karne ke liye ye command run karo

kubectl get pods

YE KUCH ISTARHA LAGEGA



```
[Administrator: Windows PowerShell]
PS C:\Users\Faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
app-pod       1/1     Running   0          24s
nginx-deployment-5dbfc68167-phfv2  1/1     Running   0          7m43s
nginx-deployment-5dbfc68167-phjkd  1/1     Running   0          7m43s
nginx-deployment-5dbfc68167-qylob  1/1     Running   0          7m43s
PS C:\Users\Faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

**NOTE: Agar sabhi Pods ka STATE Running show karraha hai to sab kuch sahi hai**

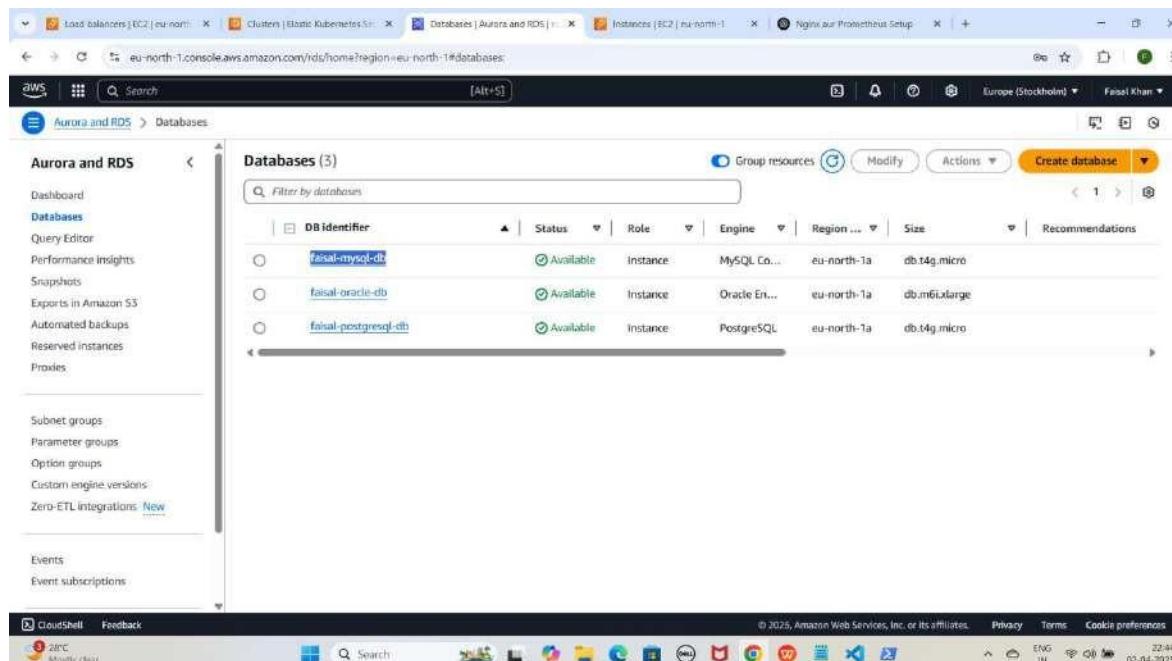
## Step 3: Amazon RDS (MySQL) Database Access Karo-Method 1

### 1. Pod ke andar jane ke liye ye command run karo

```
kubectl exec -it backend-app-pod -- /bin/bash
```

YE KUCH ISTARHA LAGEGA

```
@backend-app-pod/app
PS C:\Users\faisla\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl exec -it backend-app-pod -- /bin/bash
/appuser@backend-app-pod app$ ~
```



The screenshot shows the AWS RDS (Aurora and RDS) console with the 'Databases' page open. The left sidebar shows navigation options like Dashboard, Databases, Query Editor, and Performance Insights. The main area displays a table of databases:

DB identifier	Status	Role	Engine	Region	Size	Recommendations
faisal-mysql-db	Available	Instance	MySQL Community Server	eu-north-1a	db.t4g.micro	
faisal-oracle-db	Available	Instance	Oracle Database	eu-north-1a	db.m6xlarge	
faisal-postgresql-db	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro	

The screenshot shows the AWS RDS console for the 'faisal-mysql-db' database. The 'Connectivity & security' tab is active. Key details shown include:

- Endpoint:** faisal-mysql-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com
- Port:** 3306
- VPC:** vpc-0e345f19effbea822
- Subnet group:** faisal-subnet
- Subnets:** (not explicitly listed)

**NOTE : Aapko Endpoint aapke database ke According command me bhi update karna hoga**

## 2. MySQL database se connect karne ke liye ye command kuch istarha hogi

mysql -h faisal-mysql-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com -u FaisalKhan -p

YE KUCH ISTARHA LAGEGA

```
mysql> Select @backend-app-pod/app;
+-----+
| Select @backend-app-pod/app |
+-----+
| mysql -h faisal-mysql-db.cls8ywwg5wd.eu-north-1.rds.amazonaws.com -u FaisalKhan -p |
+-----+
1 row in set (0.00 sec)

Enter password:
+-----+
| Enter password: |
+-----+
|                                     |
+-----+
1 row in set (0.00 sec)

Welcome to the MySQL monitor.  Commands end with ; or \q.
Your MySQL connection id is 36
Server version: 8.0.20 Source distribution

Copyright (c) 2000, 2025, Oracle and/or Its affiliates.

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

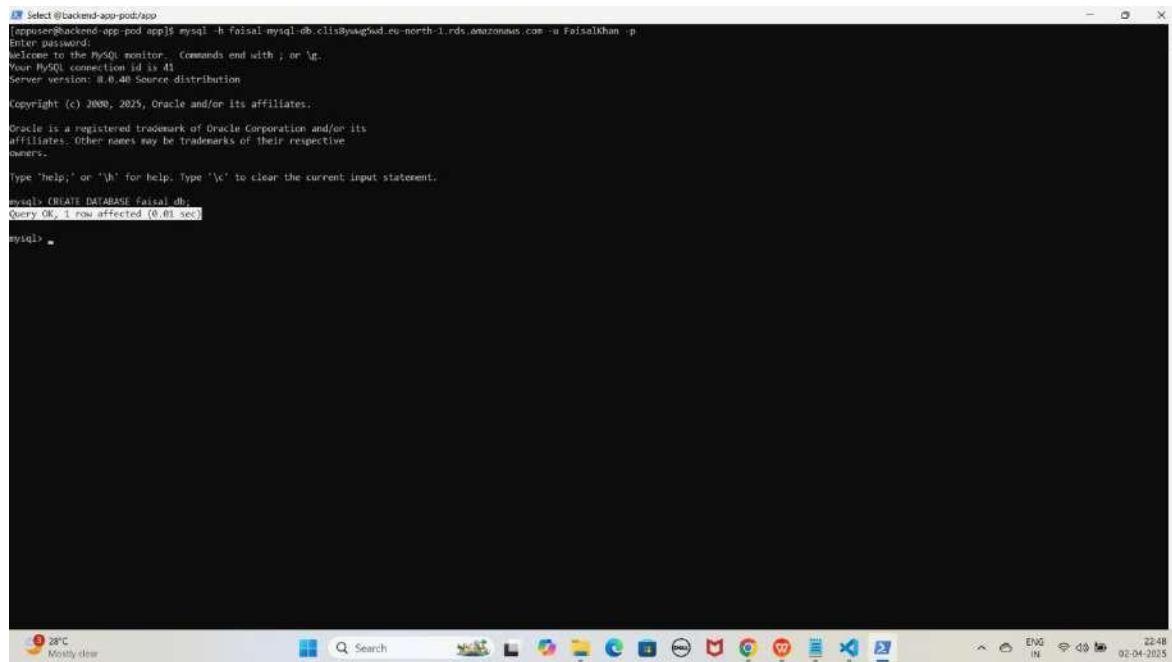
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

**Note: Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.**

### **3. Database create karne keliye ye command run karo**

```
CREATE DATABASE faisal_db;
```

**YE KUCH ISTARHA LAGEGA**



The screenshot shows a terminal window titled "Select @backend-app-pod/app". It displays the MySQL command-line interface. The user has run the command "CREATE DATABASE faisal\_db;" and received a response indicating "Query OK, 1 row affected (0.01 sec)". The terminal window is set against a dark background with a light-colored text area. The system tray at the bottom shows weather information ("28°C, Mostly clear"), a search bar, and various system icons.

```
[appuser@backend-app-pod app]$ mysql -h faisal-mysql-db.clsBywq5ad.eu-north-1.rds.amazonaws.com -u Faisalkhan -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 8.0.40-Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> CREATE DATABASE faisal_db;
Query OK, 1 row affected (0.01 sec)

mysql> .
```

## 4. Databases check karne ke liye ye command run kariye

SHOW DATABASES;

YE KUCH ISTARHA LAGEGA

```
[Select @backend-app-pod/app]
[appuser@backend-app-pod app]$ mysql -h faisal.mysql-db.clickbywsg5wd.eu-north-1.rds.amazonaws.com -u Faisalkhan -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> CREATE DATABASE faisal_db;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| faisal_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql>
```

## 5. Ab MYSQL database se exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA

```
[Select @backend-app-pod/app]
[appuser@backend-app-pod app]$ mysql -h faisal.mysql-db.clickbywsg5wd.eu-north-1.rds.amazonaws.com -u Faisalkhan -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> CREATE DATABASE faisal_db;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| faisal_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> exit
Bye
[appuser@backend-app-pod app]$
```

## Step 4: Amazon RDS (Oracle) Database Access Karo-Method 1

The screenshot shows the AWS RDS console with the following details:

**Databases (3)**

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations
faisal-mysql-db	Available	Instance	MySQL Community	eu-north-1a	db.t4g.micro	
<b>faisal-oracle-db</b>	Available	Instance	Oracle Enterprise	eu-north-1a	db.m6i.xlarge	
faisal-postgresql-db	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro	

**Aurora and RDS > Databases > Database Details | Aurora and RDS | Instances | EC2 | eu-north-1 | Nginx our Prometheus Setup | Faisal Khan | CloudShell | Feedback | 18 min delay | NH062 / Makers... | Q Search | [Alt+S] | Europe (Stockholm) | Faisal Khan | 23:02 | 02-04-2025**

**Database Details for 'faisal-oracle-db'**

**Summary**

DB identifier: <b>faisal-oracle-db</b>	Status: Available	Role: Instance	Engine: Oracle Enterprise Edition	Recommendations
CPU: 2.36%	Class: db.m6i.xlarge	Current activity: 0.02 sessions	Region & AZ: eu-north-1a	

**Connectivity & security**

Endpoint copied: **faisal-oracle-db.cll8ywwg5wd.eu-north-1.odb.amazonaws.com**

<b>Port:</b> 1521	<b>Networking</b>	<b>Security</b>
	Availability Zone: eu-north-1a VPC: vpc-0e345f19effbea022 Subnet group: faisal-subnet Subnets: subnet-048cd5fb4ac1b8365e	VPC security groups: ALL-TRAFFIC-ALLOW (sg: 0eaa265816a26bec) Active: Yes Publicly accessible: No Certificate authority: rds-ca-ssl2048-g1

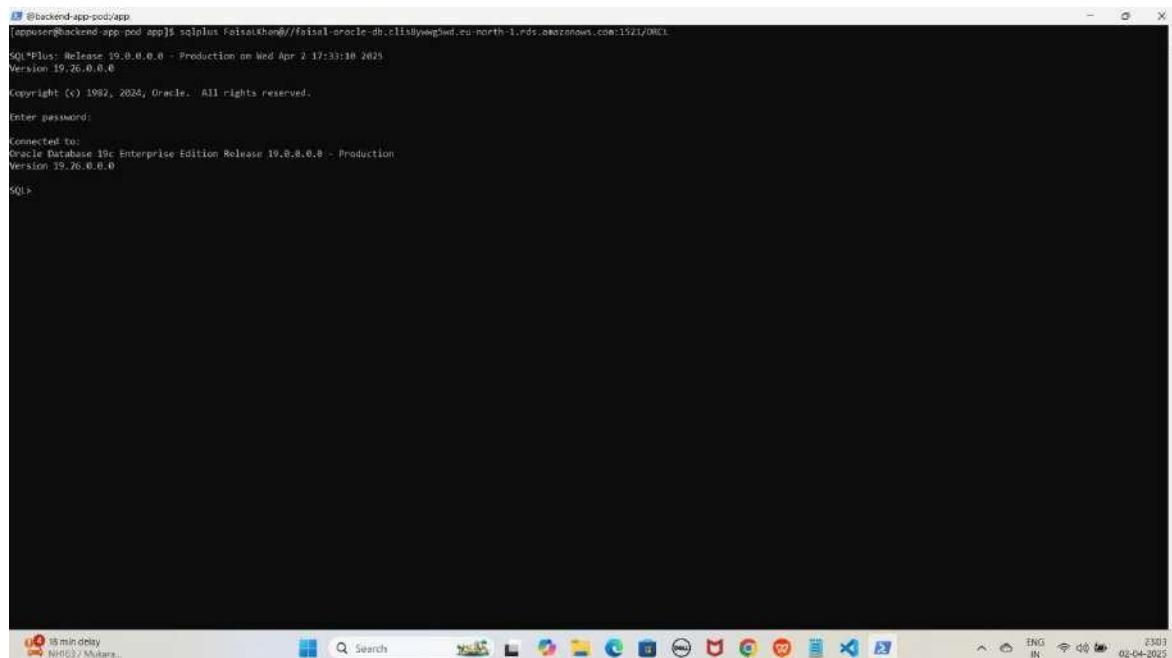
**CloudShell | Feedback | 18 min delay | NH062 / Makers... | Q Search | [Alt+S] | Europe (Stockholm) | Faisal Khan | 23:02 | 02-04-2025**

**NOTE : Aapko Endpoint aapke database ke According command me bhi update karna hoga**

## **1. Oracle database se connect karne ke liye ye command kuch istarha hogi**

```
sqlplus FaisaLKhan@//faisal-oracle-db.cis8ywwg5wd.eu-north-1.rds.amazonaws.com:1521/ORCL
```

**YE KUCH ISTARHA LAGEGA**



```
[appuser@backend-app-pod app]$ sqlplus FaisaLKhan@//faisal-oracle-db.cis8ywwg5wd.eu-north-1.rds.amazonaws.com:1521/ORCL
SQL*Plus: Release 19.8.0.0.0 - Production on Wed Apr 2 17:33:10 2025
Version 19.26.0.0.0
Copyright (c) 1982, 2024, Oracle. All rights reserved.

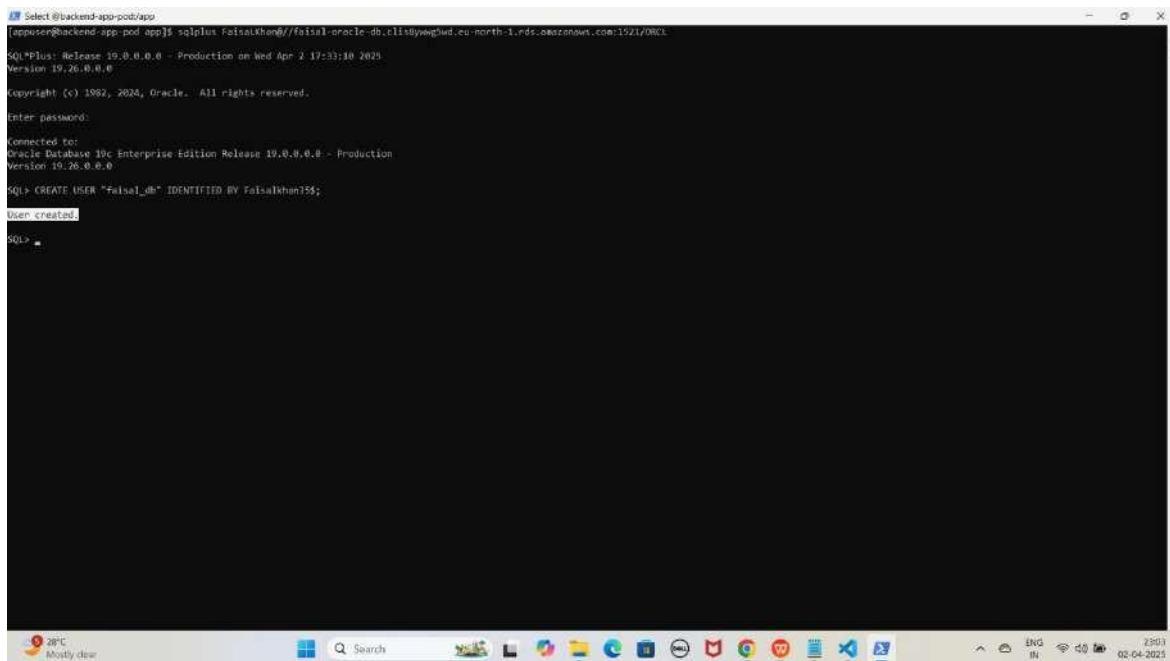
Enter password:
Connected to:
Oracle Database 19c Enterprise Edition Release 19.8.0.0.0 - Production
Version 19.26.0.0.0
SQL>
```

**Note:** Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.

## 2. Database create karne keliye ye command run karo

```
CREATE USER "faisal_db" IDENTIFIED BY Faisalkhan35$;
```

YE KUCH ISTARHA LAGEGA



```
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0

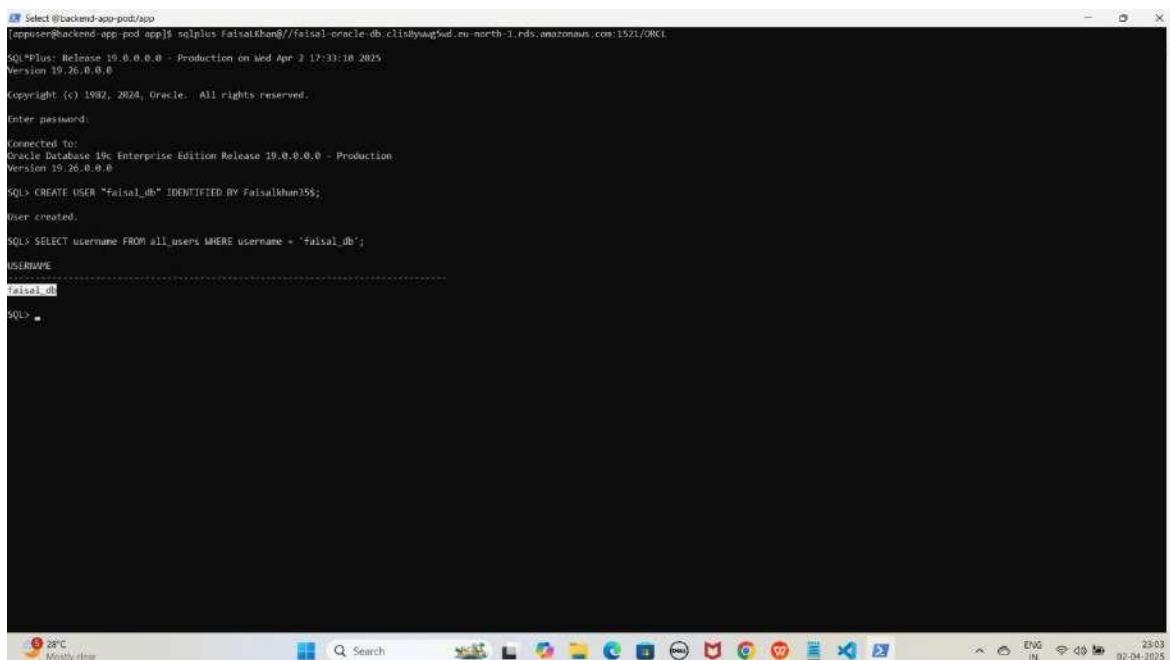
SQL> CREATE USER "faisal_db" IDENTIFIED BY Faisalkhan35$;
User created.

SQL>
```

## 3. Databases check karne ke liye ye command run kariye

```
SELECT username FROM all_users WHERE username = 'faisal_db';
```

YE KUCH ISTARHA LAGEGA



```
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0

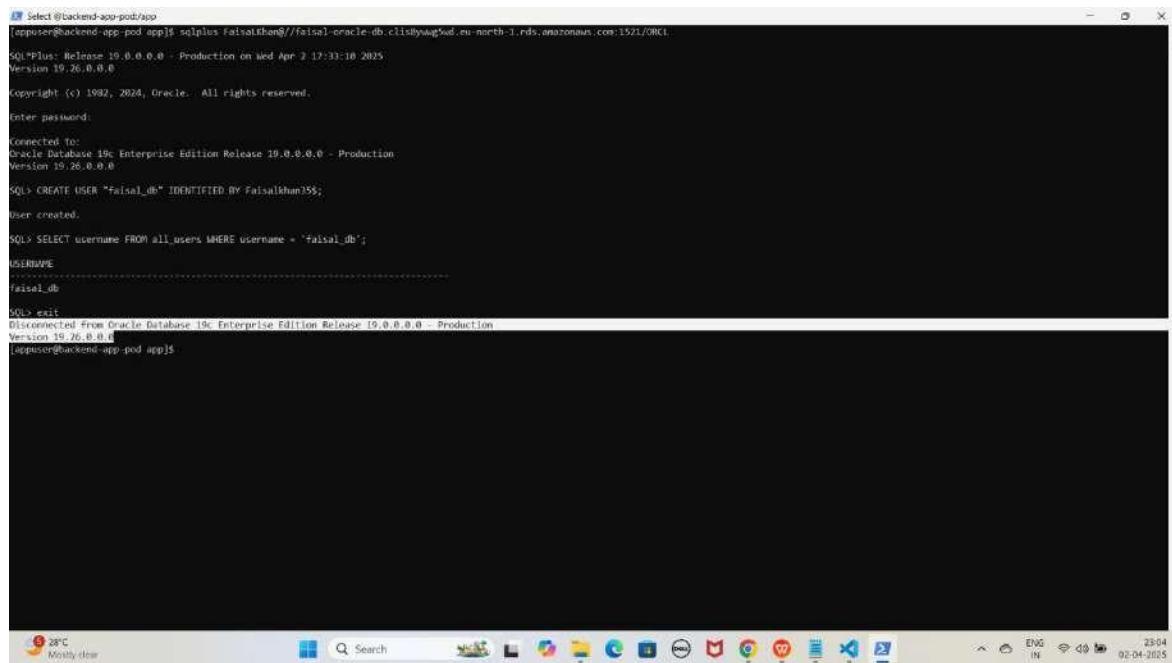
SQL> SELECT username FROM all_users WHERE username = 'faisal_db';
USERNAME
-----
faisal_db

SQL>
```

#### 4. Ab Oracle database se exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



sqlplus FaisalKhan@//faisal.oracle\_db.cldbyag5u.eu-north-1.rds.amazonaws.com:1521/ORCL  
SQL\*Plus: Release 19.0.0.0.0 - Production on wed Apr 2 17:33:10 2025  
Version 19.26.0.0.0  
Copyright (c) 1982, 2024, Oracle. All rights reserved.  
Enter password:  
Connected To:  
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production  
Version 19.26.0.0.0  
SQL> CREATE USER "Faisal\_db" IDENTIFIED BY FaisalKhan35\$;  
User created.  
SQL> SELECT username FROM all\_users WHERE username = 'Faisal\_db';  
USERNAME  
-----  
Faisal\_db  
SQL> exit  
Disconnected From Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production  
Version 19.26.0.0.0  
(appuser@backend-app-pod app15)

## Step 5: Amazon RDS (PostgreSQL) Database Access Karo-Method 1

The screenshot shows two windows of the AWS RDS console.

**Top Window:** Shows the 'Databases' list under the 'Aurora and RDS' section. It lists three databases: 'faisal-mysql-db', 'faisal-oracle-db', and 'faisal-postgresql-db'. The 'faisal-postgresql-db' row is selected.

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations
faisal-mysql-db	Available	Instance	MySQL Co...	eu-north-1a	db.t4g.micro	
faisal-oracle-db	Available	Instance	Oracle En...	eu-north-1a	db.m6large	
<b>faisal-postgresql-db</b>	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro	

**Bottom Window:** Shows the 'Database Details' page for the 'faisal-postgresql-db' database. It displays various details like DB identifier, status, role, engine, and connectivity information.

**Summary:**

DB identifier	Status	Role	Engine	Recommendations
faisal-postgresql-db	Available	Instance	PostgreSQL	

**Connectivity & security:**

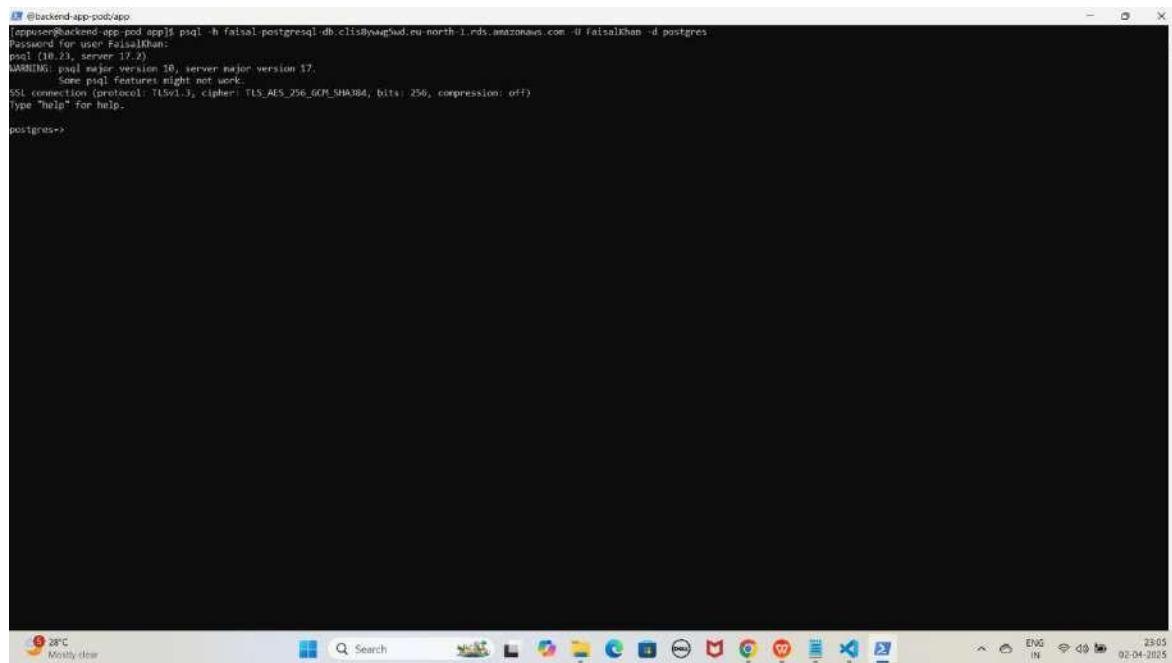
- Endpoint copied: [faisal-postgresql-db.ell8wwg5wd.eu-north-1.rds.amazonaws.com](#)
- Port: 5432
- Networking:
  - Availability Zone: eu-north-1a
  - VPC: vpc-0e345f19effbea122
  - Subnet group: faisal-subnet
  - Subnet: Subnet-00000000
- Security:
  - VPC security groups: ALL-TRAFFIC-ALLOW (sg-0eaa9265b16a260ec)
  - Active
  - Publicly accessible: No
  - Certificate authority: rds-ca-ssl2048-q1

**NOTE : Aapko Endpoint aapke database ke According command me bhi update karna hogा**

**1. PostgreSQL database se connect karne ke liye ye command kuch istarha hogi**

```
psql -h faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -U FaisalKhan -d postgres
```

**YE KUCH ISTARHA LAGEGA**



The screenshot shows a terminal window titled '@backend-app-pod/app' running on a Windows operating system. The command entered is 'psql -h faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -U FaisalKhan -d postgres'. The output shows the PostgreSQL prompt 'postgres>' indicating a successful connection. The terminal window is set against a dark background with light-colored text. The taskbar at the bottom of the screen displays various application icons, including a weather widget showing '28°C Mostly clear'.

```
[appuser@backend-app-pod app]$ psql -h faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -U FaisalKhan -d postgres
Password for user FaisalKhan:
psql (10.23, server 17.2)
WARNING: psql major version 10, server major version 17.
         SSL features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

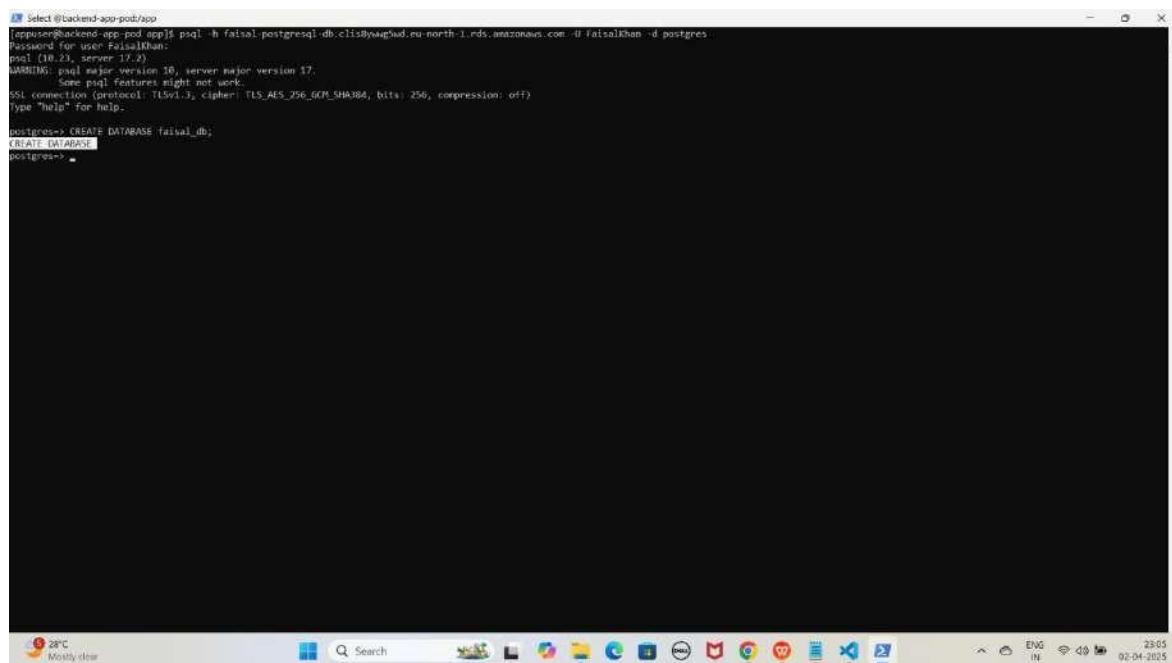
postgres>
```

**Note:** Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.

## 2. Database create karne keliye ye command run karein

```
CREATE DATABASE faisal_db;
```

YE KUCH ISTARHA LAGEGA



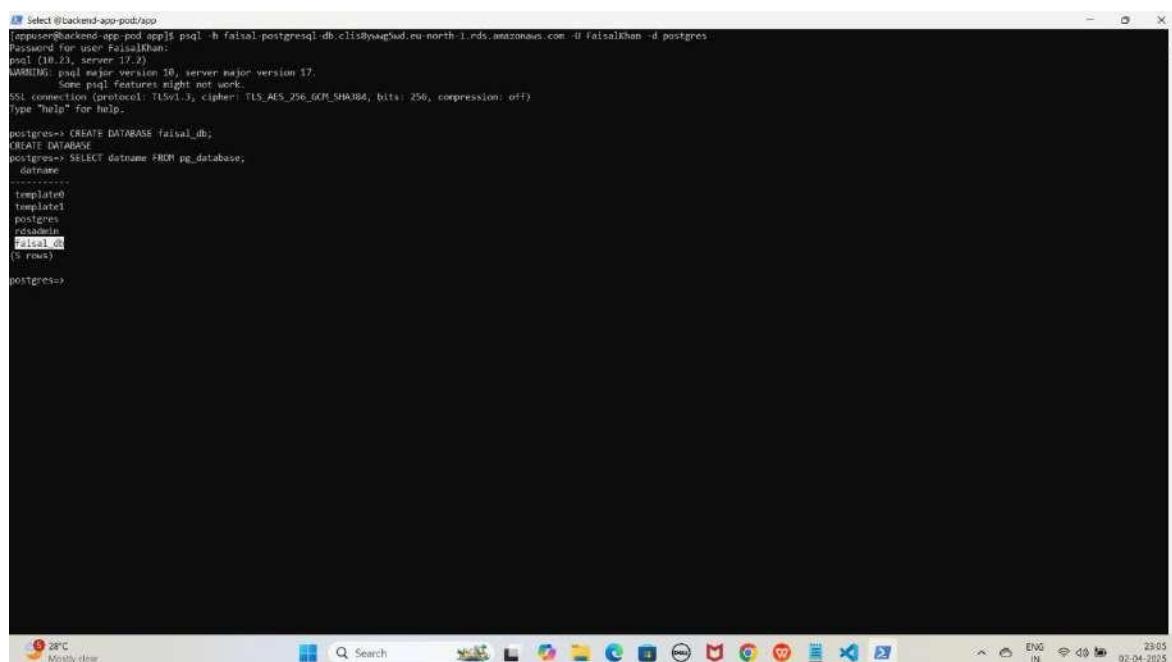
```
[epuser@backend-app-pod app]$ psql -h faisal.postgresql.db.clsbywqjwd.eu.north-1.rds.amazonaws.com -U faisalkhan -d postgres
Password for user Faisalkhan:
psql (10.23, server 17.2)
WARNING: pgsql major version 10, server major version 17.
        Some pgsql features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=> CREATE DATABASE faisal_db;
CREATE DATABASE
postgres=> ^C
```

## 3. Databases check karne ke liye ye command run kariye

```
SELECT datname FROM pg_database;
```

YE KUCH ISTARHA LAGEGA



```
[epuser@backend-app-pod app]$ psql -h faisal.postgresql.db.clsbywqjwd.eu.north-1.rds.amazonaws.com -U faisalkhan -d postgres
Password for user Faisalkhan:
psql (10.23, server 17.2)
WARNING: pgsql major version 10, server major version 17.
        Some pgsql features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

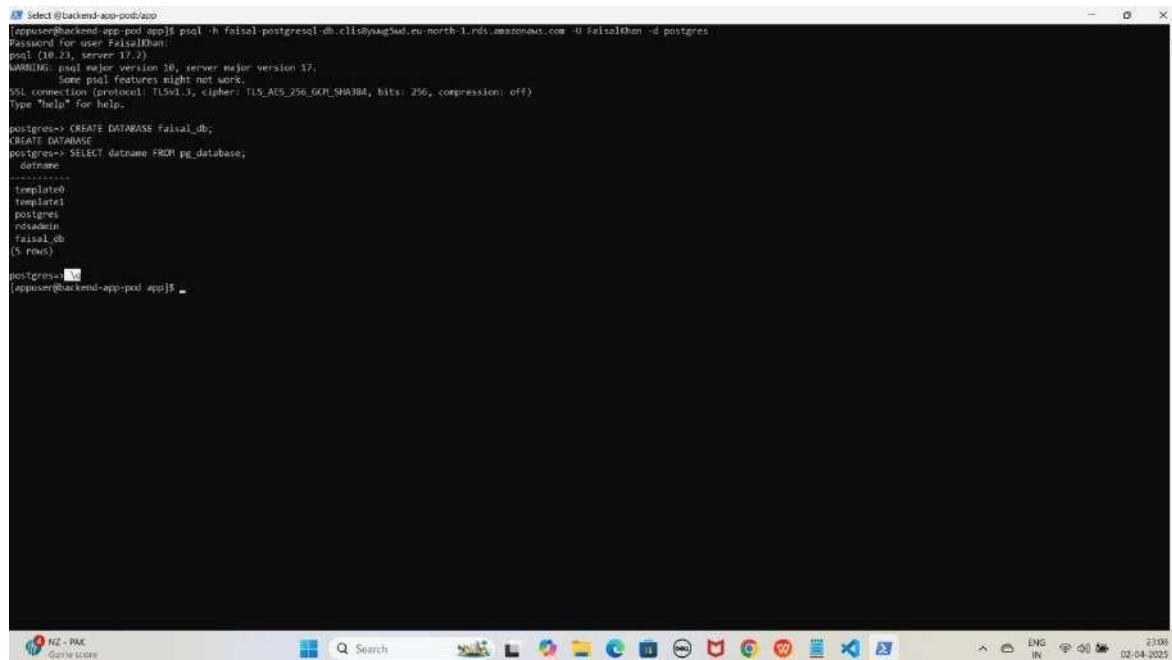
postgres=> CREATE DATABASE faisal_db;
CREATE DATABASE
postgres=> SELECT datname FROM pg_database;
 datname
-----
 template0
 template1
 postgres
 regression
 faisal_db
(5 rows)

postgres=>
```

#### **4. Ab PostgreSQL database se exit hoagaye exit hone ke liye exit typ kariye**

\q

YE KUCH ISTARHA LAGEGA



```
[appuser@backend-app-pod app]$ psql -h faisal-postgresql-db.clsdywagjed.eu-north-1.rds.amazonaws.com -U faisalKhem -d postgres
Password for user faisalKhem:
psql (10.23, server 17.2)
WARNING: pgsql major version 10, server major version 17.
          Some pgsql features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

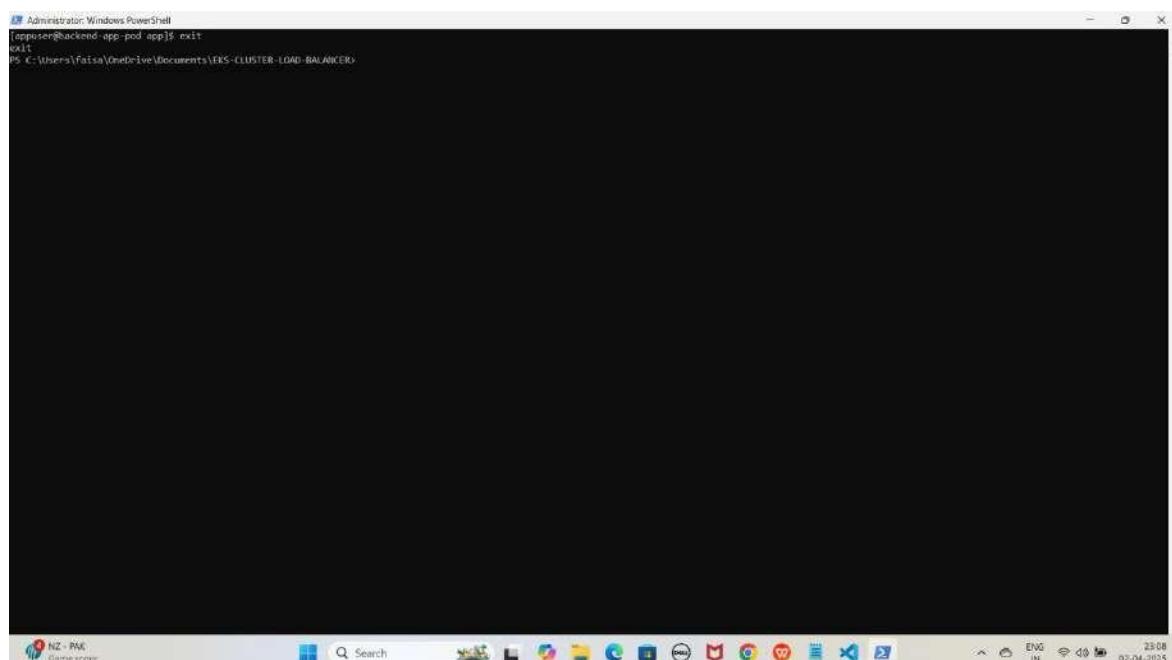
postgres=> CREATE DATABASE faisal_db;
CREATE DATABASE
postgres=> \l
          List of databases
   Name    | Owner
-----+-----
template0 | postgres
template1 | postgres
postgres  | postgres
faisal   | faisal_khem
faisal_db| faisal_khem
(5 rows)

postgres=\q
[appuser@backend-app-pod app]$
```

#### **5. Ab pod se exit hoagaye exit hone ke liye exit type kariye:**

exit

YE KUCH ISTARHA LAGEGA



```
[Administrator: Windows PowerShell
[appuser@backend-app-pod app]$ exit
exit
PS C:\Users\faisa\OneDrive\Documents\ERS-CLUSTER-LOAD-BALANCER]
```

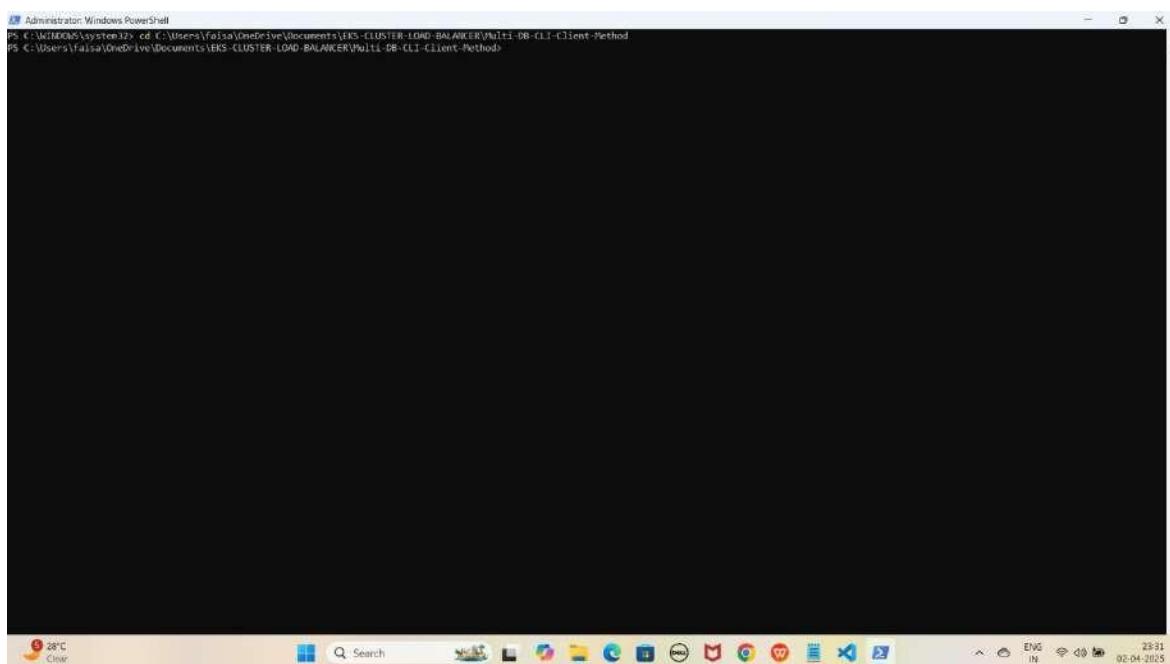
## **Method 2: Client Pods ke through Database Access Karna (With Endpoint)**

### **Step 1: Client Pods ki YAML Files ki Location par Navigate Karo**

#### **1. Location par jao**

```
cd C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\Multi-DB-CLI-Client-Method
```

YE KUCH IS TARHA LAGEGA



### **Step 2: Client Pods Deploy karo**

**Client Pods** ka kaam database se **securely connect** hona aur uska access lena hota hai. Yeh **pods databases (MySQL, PostgreSQL, aur Oracle) Database se connect hoke data retrieve** karne ya **update** karne ke liye use hote hain.

#### **1. Client Method me files jo hum use karenge**

- 1. mysql-client.yaml**
- 2. oracle-client.yaml**
- 3. postgresql-client.yaml**

## 1. mysql-client.yaml File Ka Kaam

Yeh **file MySQL client image** ka **pod create** karne ke liye use hoti hai, jisse hum **Amazon RDS MySQL** ko **access** kar sakte hain.

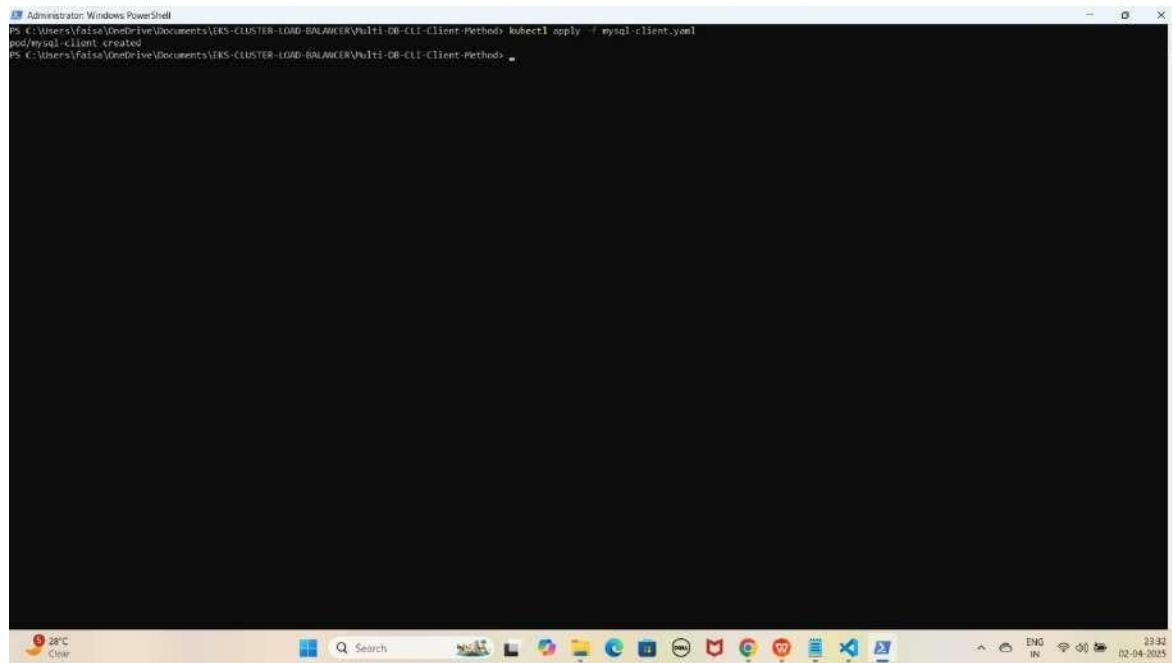
JAISE KI:-

```
apiVersion: v1
kind: Pod
metadata:
  name: mysql-client
spec:
  containers:
  - name: mysql
    image: mysql:latest # Prebuilt MySQL client image from Docker Hub
    command: [ "sleep", "infinity" ] # Keeps the container running
```

## MYSQL Client Apply Karo

kubectl apply -f mysql-client.yaml

YE KUCH ISTARHA LAGEGA



```
Administrator: Windows PowerShell
PS C:\Users\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\multi-DB-CLI-Client-Method> kubectl apply -f mysql-client.yaml
pod/mysql-client created
PS C:\Users\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\multi-DB-CLI-Client-Method>
```

## 2. oracle-client.yaml File Ka Kaam

Yeh **file Oracle client image** ka **pod create** karne ke liye use hoti hai, jisse hum **Amazon RDS Oracle** ko **access** kar sakte hain.

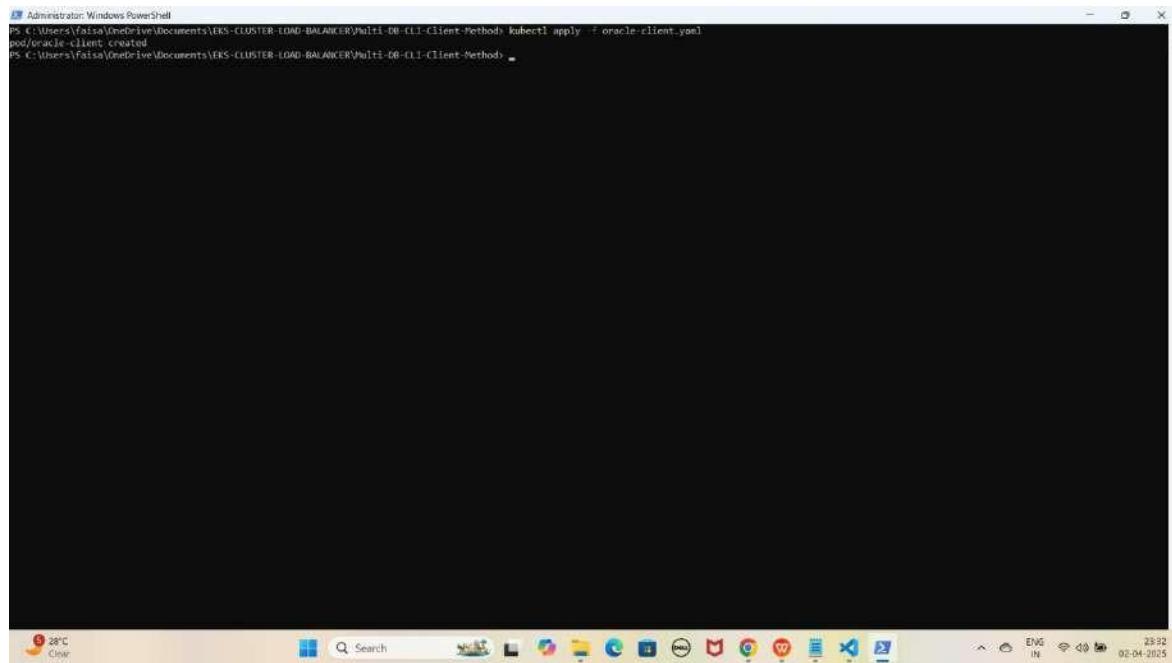
JAISE KI:-

```
apiVersion: v1
kind: Pod
metadata:
  name: oracle-client
spec:
  containers:
    - name: oracle-client
      image: ghcr.io/oracle/oraclelinux8-instantclient:21 # Oracle Instant Client image
      command: [ "sleep", "infinity" ] # Keeps the container running for exec access
```

## Oracle Client Apply Karo

kubectl apply -f oracle-client.yaml

YE KUCH ISTARHA LAGEGA



### 3. postgresql-client.yaml File Ka Kaam

Yeh **file PostgreSQL client image** ka **pod create** karne ke liye use hoti hai, jisse hum **Amazon RDS PostgreSQL** ko **access** kar sakte hain.

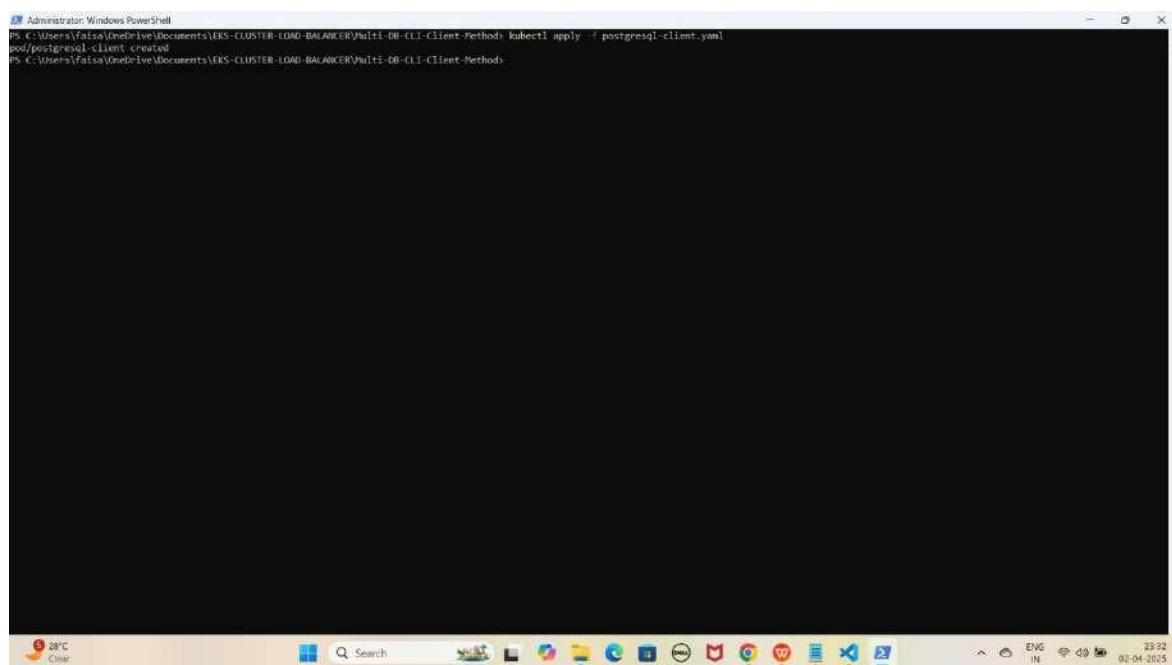
JAISE KI:-

```
apiVersion: v1
kind: Pod
metadata:
  name: postgresql-client
spec:
  containers:
    - name: postgresql-client
      image: postgres:latest # Prebuilt PostgreSQL client image from Docker Hub
      command: [ "sleep", "infinity" ] # Keeps the container running so you can exec into it later
```

### PostgreSQL Client Apply Karo

kubectl apply -f postgresql-client.yaml

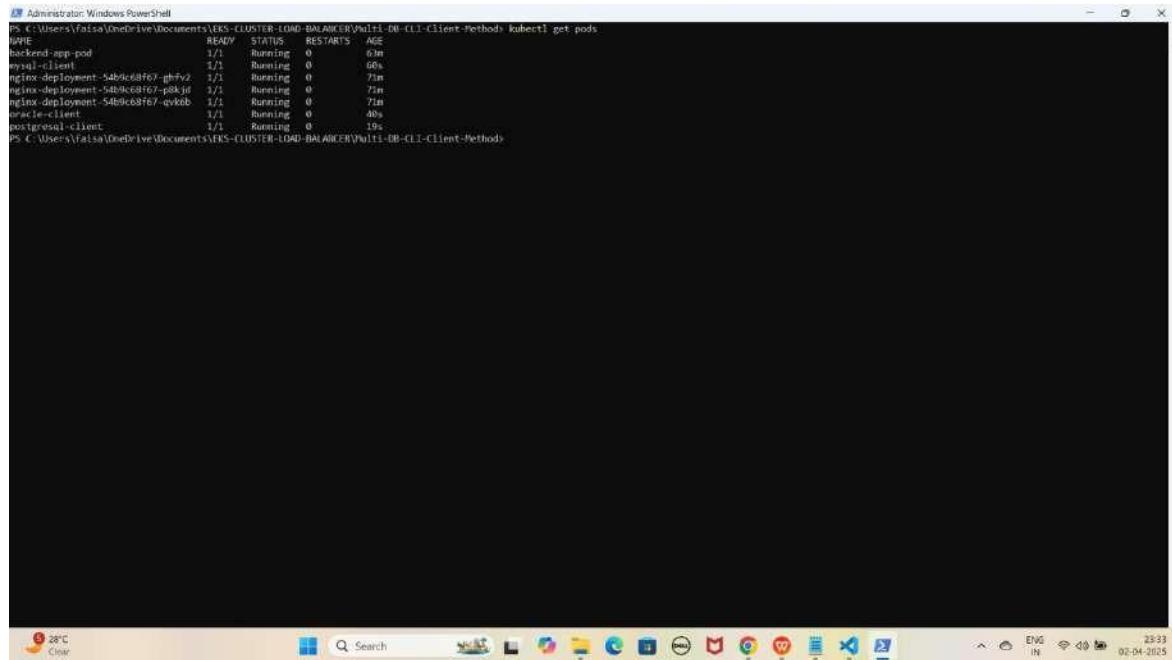
YE KUCH ISTARHA LAGEGA



```
Administrator: Windows PowerShell
PS C:\Users\faiza\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\Multi-DB-CLI-Client> kubectl apply -f postgresql-client.yaml
pod/postgresql-client created
PS C:\Users\faiza\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\Multi-DB-CLI-Client>
```

## **1. Client pods check karne ke liye ye command run karo**

**kubectl get pods**



```
Administrator: Windows PowerShell
PS C:\Users\faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\multi-db-CLI-Client-Method> kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
backend-app-pod   1/1     Running   0          6m
mysql-client     1/1     Running   0          60s
nginx-deployment-54b9c68f67-gh7v2   1/1     Running   0          71m
nginx-deployment-54b9c68f67-pdkjd   1/1     Running   0          73m
oracle-client     1/1     Running   0          40s
postgresql-client 1/1     Running   0          19s
PS C:\Users\faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\multi-db-CLI-Client-Method>
```

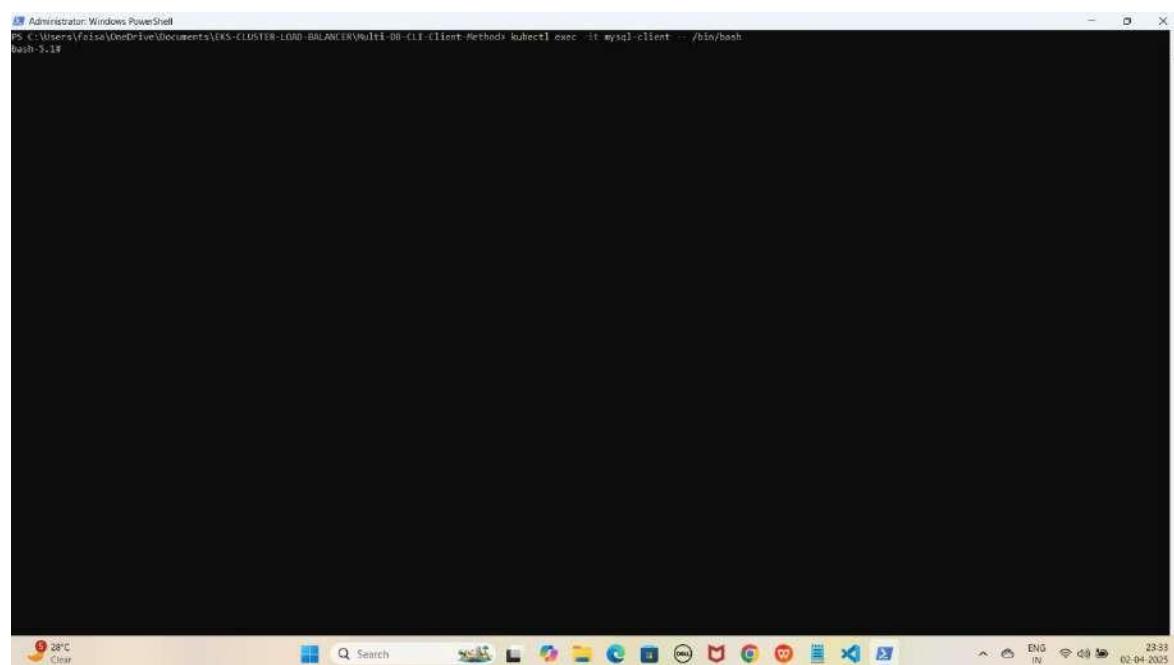
**NOTE: Agar sabhi pods, including client pods, STATUS Running show karraha hai, to sab kuch sahi hai.**

## **Step 3: Amazon RDS (MySQL) Database Access Karo-Method 2**

### **1. MySQL client pod me jane ke liye ye command run kaiye**

```
kubectl exec -it mysql-client -- /bin/bash
```

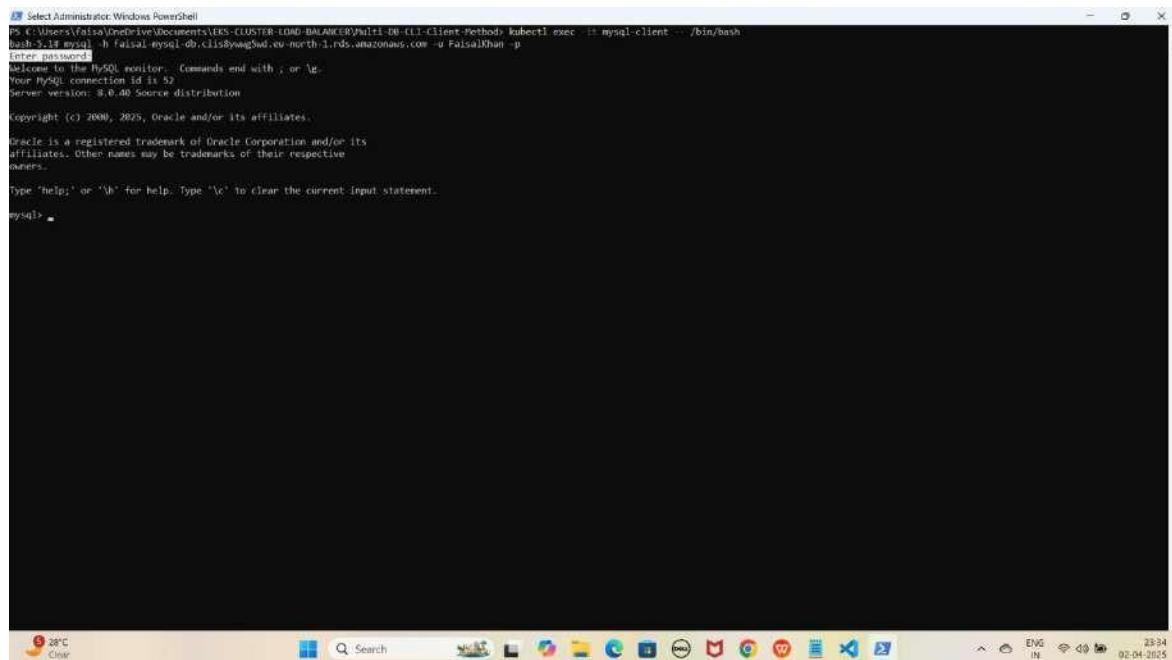
YE KUCH ISTARHA LAGEGA



## 2. MYSQL database se connect karne ke liye ye command kuch istarha hogi

```
mysql -h faisal-mysql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -u  
FaisalKhan -p
```

YE KUCH ISTARHA LAGEGA



The screenshot shows a Windows PowerShell window titled "Select Administrator: Windows PowerShell". The command entered is:

```
PS C:\Users\faisa\OneDrive\Documents\ECS-CLUSTER-LOAD-BALANCING\multi-node-CLI-Client\method; kubectl exec -i mysql-client -- /bin/bash
```

The MySQL prompt is visible, indicating a successful connection:

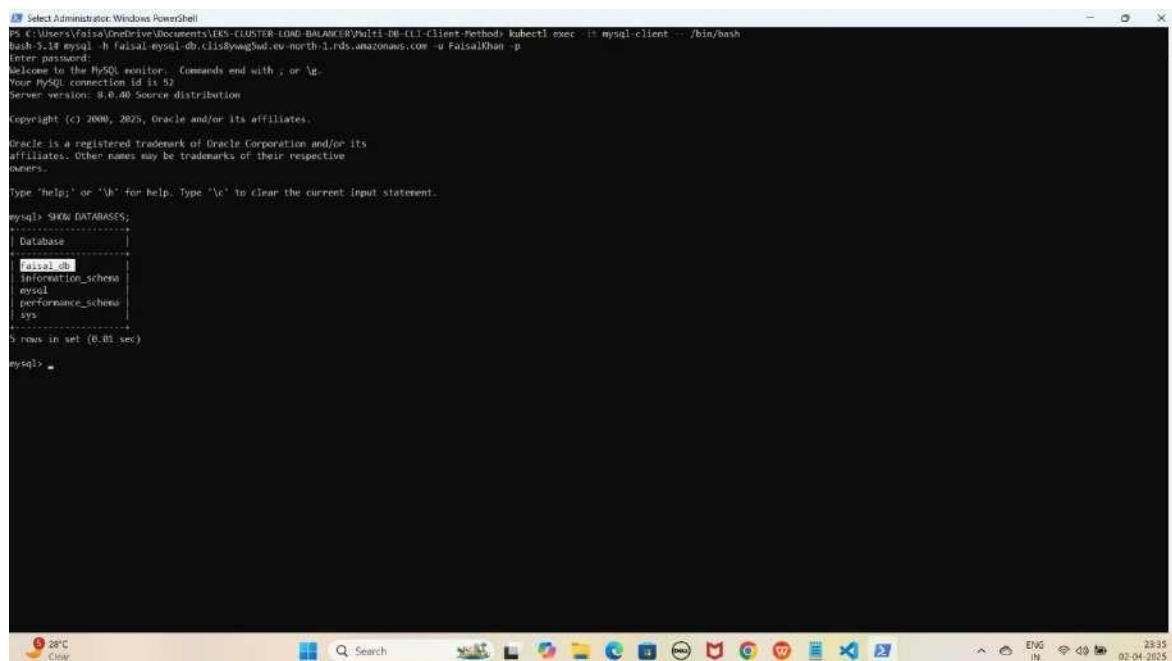
```
mysql>
```

**Note: Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke rds-secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.**

### 3. Databases check karne ke liye ye command run kariye

SHOW DATABASES;

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\Faisal\OneDrive\Documents\LEKS-CLUSTER LOAD BALANCING\multi-db CLI Client\Method1>kubectl exec -it mysql-client -- /bin/bash
bash-5.1$ mysql -h faisal.mysql-db.clsbbywq5w.eu-north-1.rds.amazonaws.com -u FaisalKhan -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 52
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| Faisal_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

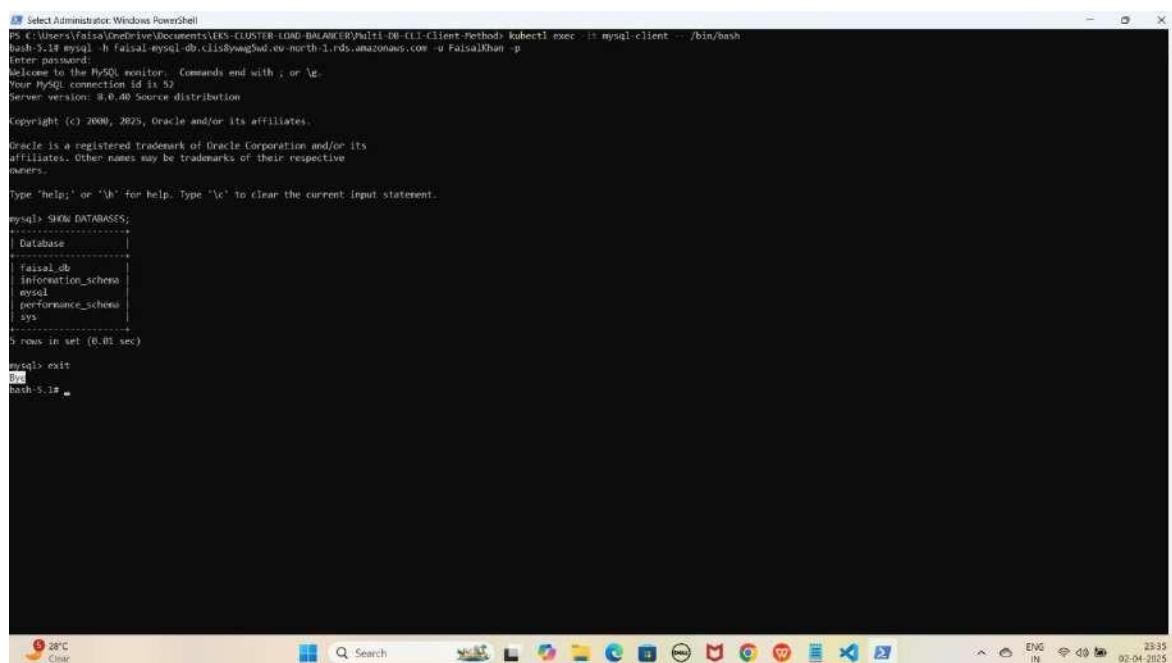
mysql>
```

**NOTE: Yahan hum dobara database create nahi karenge, kyunki pehle hi Method 1 mein create kar chuke hain. Hum sirf ye check karenge ki database available hai ya nahi.**

#### 4. MYSQL database se exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisal\OneDrive\Documents\eks-cluster-load-balancer\multi-db\cli-client-method> kubectl exec -it mysql-client -- /bin/bash
bash-5.1# mysql -h faisal.mysql-db.cls8wq5wd.eu-north-1.rds.amazonaws.com -u FaisalKhan -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

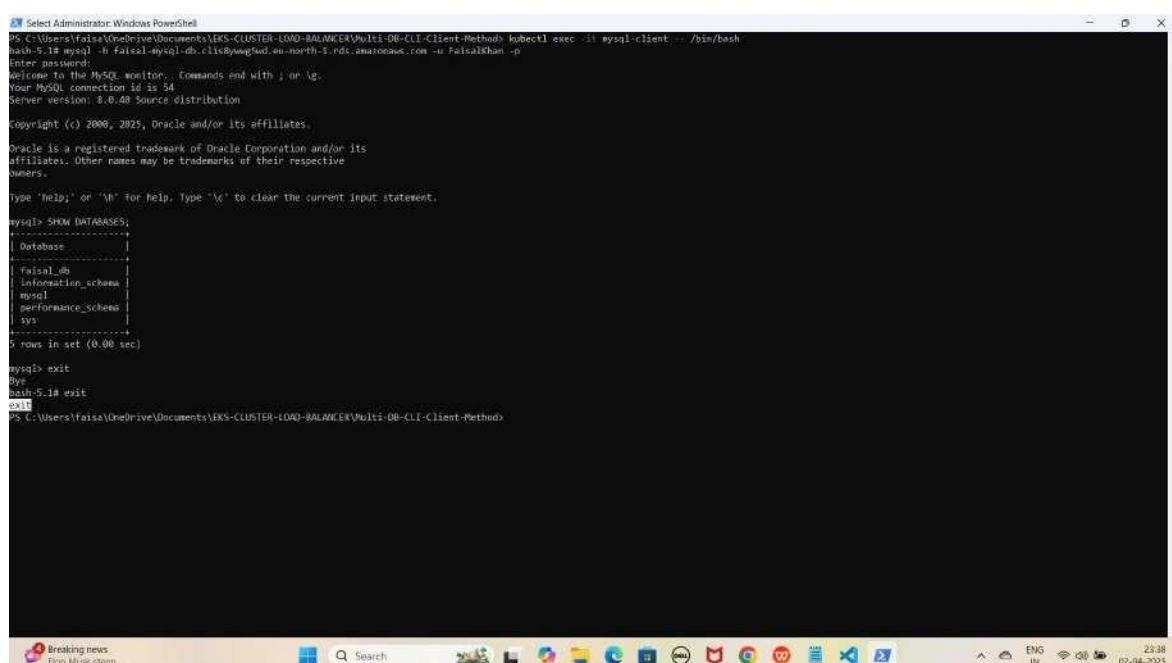
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| faisal_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> exit
Bye
bash-5.1#
```

#### 5. MYSQL client pod se bhi exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisal\OneDrive\Documents\eks-cluster-load-balancer\multi-db\cli-client-method> kubectl exec -it mysql-client -- /bin/bash
bash-5.1# mysql -h faisal.mysql-db.cls8wq5wd.eu-north-1.rds.amazonaws.com -u FaisalKhan -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| faisal_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

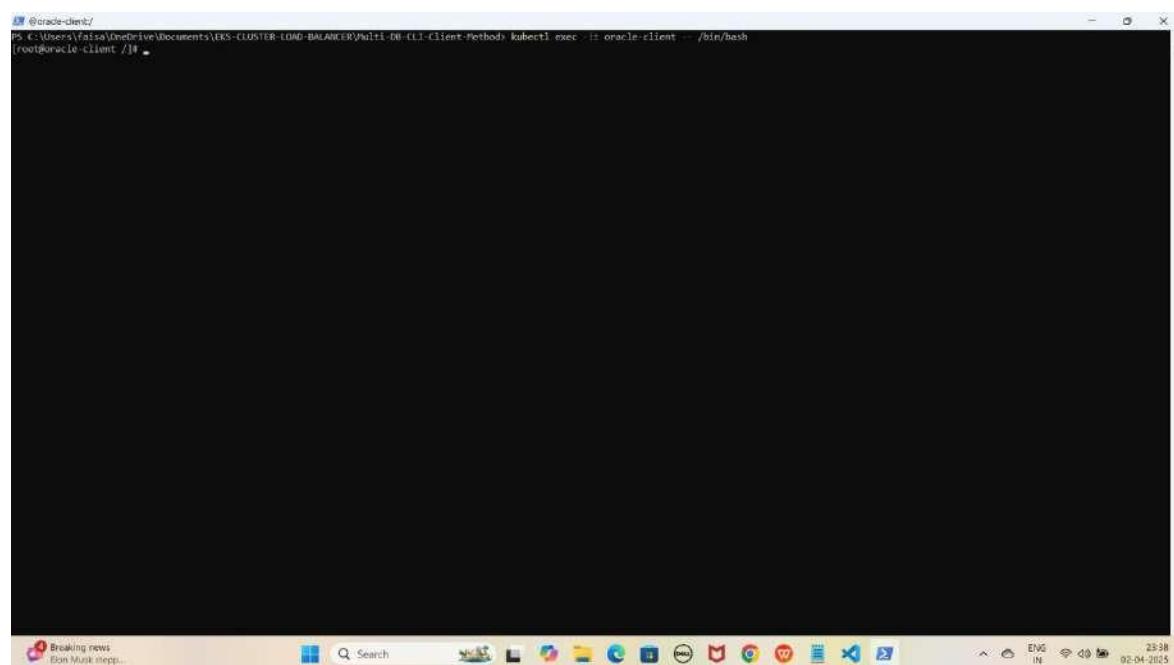
mysql> exit
Bye
bash-5.1# exit
exit
PS C:\Users\faisal\OneDrive\Documents\eks-cluster-load-balancer\multi-db\cli-client-method>
```

## Step 4: Amazon RDS (Oracle) Database Access Karo-Method 2

### 1. Oracle client pod me jane ke liye ye command run kaiye

```
kubectl exec -it oracle-client -- /bin/bash
```

YE KUCH ISTARHA LAGEGA

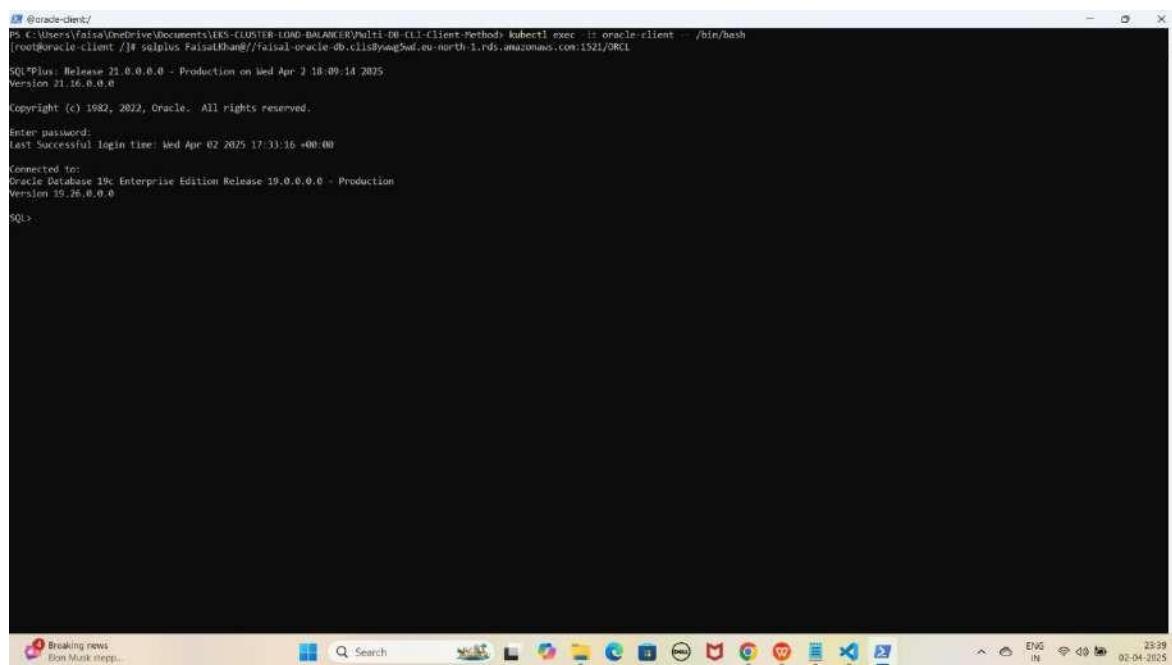


The screenshot shows a Windows desktop environment with a terminal window open. The terminal window title is '@oracle-client' and the path is 'C:\Users\Yafisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\multi-db-client-Method2'. The command entered is 'kubectl exec -it oracle-client -- /bin/bash'. The terminal window is mostly blank, indicating no output has been displayed yet. The taskbar at the bottom shows various icons for applications like File Explorer, Edge, and FileZilla. The system tray indicates the date as 02-04-2025 and the time as 23:38.

## 2. Oracle database se connect karne ke liye ye command kuch istarha hogi

```
sqlplus FaisaLKhan@//faisal-oracle-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com:1521/ORCL
```

YE KUCH ISTARHA LAGEGA



```
@oracle-client
ps C:\Users\faisa\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER\multi-node-CLI-Client\Methods> kubectl exec -it oracle-client -- /bin/bash
root@oracle-client:/# sqlplus FaisaLKhan@//faisal-oracle-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com:1521/ORCL
SQL*Plus: Release 22.0.0.0.0 - Production on Wed Apr 2 18:09:14 2025
Version 22.16.0.0.0
Copyright (c) 1982, 2022, Oracle. All Rights Reserved.

Enter password:
Last Successful login time: Wed Apr 02 2025 17:33:16 +00:00

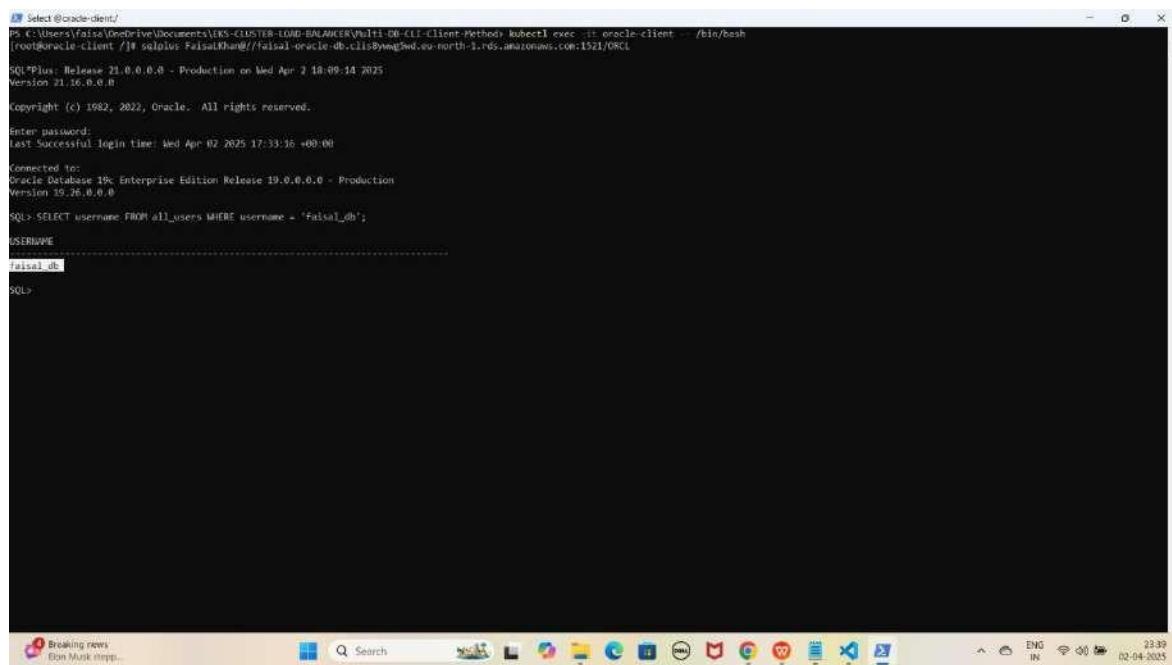
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0
SQL>
```

**Note:** Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke rds-secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.

### **3. Databases check karne ke liye ye command run kariye**

```
SELECT username FROM all_users WHERE username = 'faisal_db';
```

**YE KUCH ISTARHA LAGEGA**



The screenshot shows a terminal window titled "Select @oracle-client". The command executed is:

```
ps -C oracle-client
ps C:\Users\faisa\OneDrive\Documents\LKS-CLUSTER-LINK-BALANCER\multi_db\cli-client\Method1\kubectl exec -it oracle-client /bin/bash
root@oracle-client:/# sqlplus FaisalKhan@//faisal.oracle-db.clsBymg5ed.eu-north-1.rds.amazonaws.com:1521/ORCL
SQL*Plus: Release 23.0.0.0.0 - Production on Wed Apr 2 18:09:14 2025
Version 23.0.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.
Enter password:
Last Successful login time: Wed Apr 02 2025 17:33:36 +00:00
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.0.0.0.0
SQL> SELECT username FROM all_users WHERE username = 'faisal_db';
USERNAME
-----
faisal_db
SQL>
```

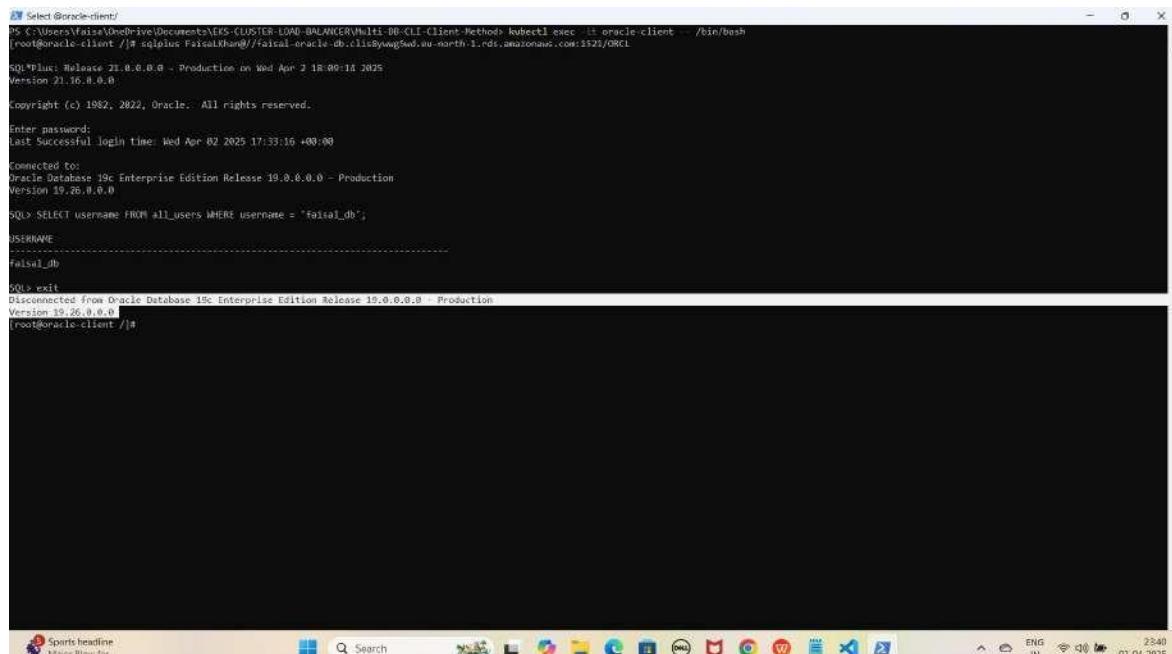
The terminal window is running on a Windows operating system, as indicated by the taskbar icons at the bottom.

**NOTE: Yahan hum dobara database create nahi karenge, kyunki pehle hi Method 1 mein create kar chuke hain. Hum sirf ye check karenge ki database available hai ya nahi.**

#### 4. Oracle database se exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\Faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\multi-db-CLI-Client-Method> kubectl exec -it oracle-client -- /bin/bash
[root@oracle-client ~]# sqlplus FaisalKhan@//faisal-oracle-db.cldBywq5ud.eu-north-1.rds.amazonaws.com:1521/ORCL
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 2 18:00:14 2025
Version 21.0.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed Apr 02 2025 17:33:16 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0

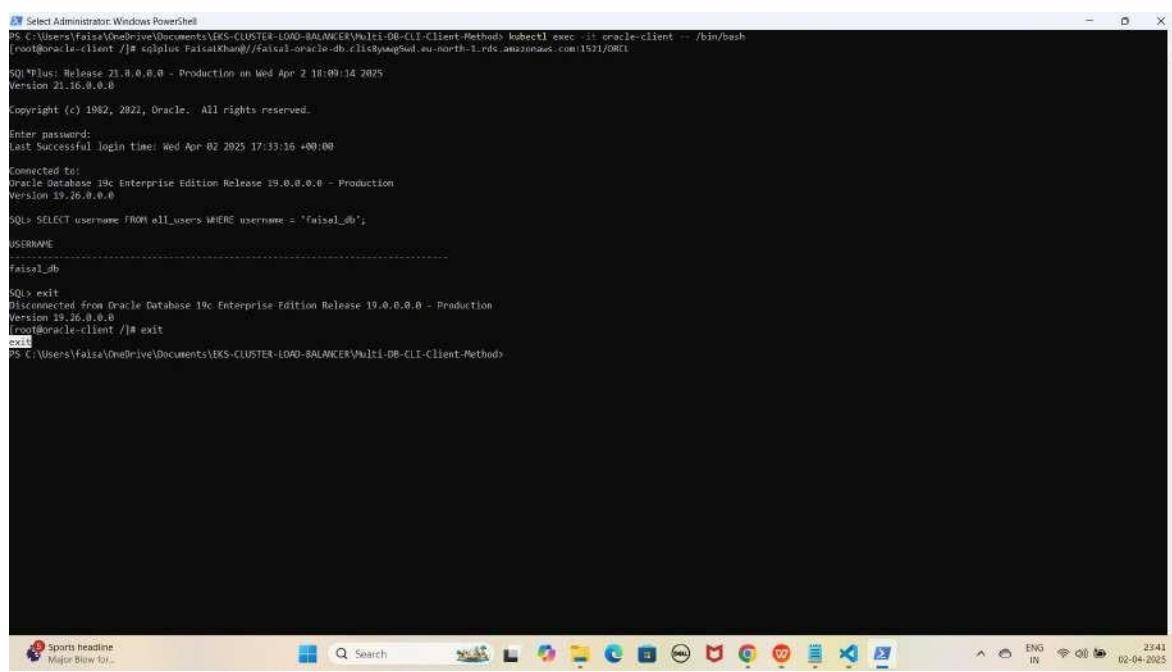
SQL> SELECT username FROM all_users WHERE username = 'faisal_db';
USERNAME
-----
faisal_db

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0
[root@oracle-client ~]#
```

#### 5. Oracle client pod se bhi exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\Faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\multi-db-CLI-Client-Method> kubectl exec -it oracle-client -- /bin/bash
[root@oracle-client ~]# sqlplus FaisalKhan@//faisal-oracle-db.cldBywq5ud.eu-north-1.rds.amazonaws.com:1521/ORCL
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 2 18:09:14 2025
Version 21.0.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed Apr 02 2025 17:33:16 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0

SQL> SELECT username FROM all_users WHERE username = 'faisal_db';
USERNAME
-----
faisal_db

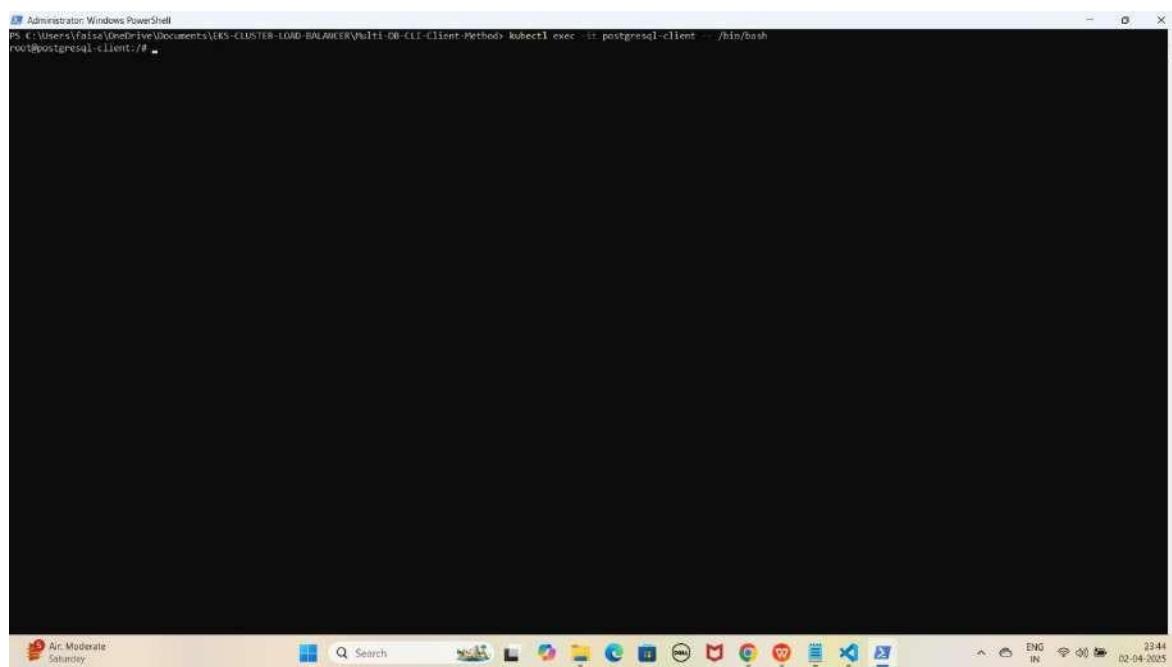
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0
[root@oracle-client ~]# exit
exit
PS C:\Users\Faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER\multi-db-CLI-Client-Method>
```

## Step 5: Amazon RDS (PostgreSQL) Database Access Karo-Method 2

### 1. PostgreSQL client pod me jane ke liye ye command run kaiye

```
kubectl exec -it postgresql-client -- /bin/bash
```

YE KUCH ISTARHA LAGEGA

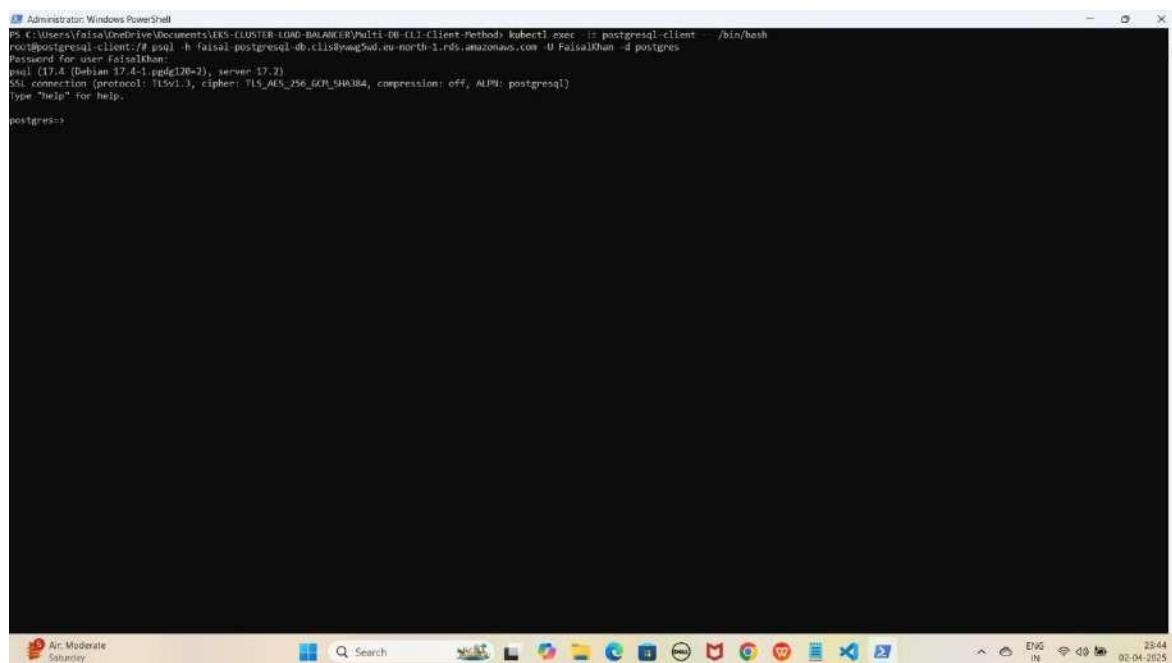


A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command entered is "kubectl exec -it postgresql-client -- /bin/bash". The window is mostly black, indicating no output has been displayed yet. The taskbar at the bottom shows various icons for applications like File Explorer, Edge, and FileZilla. The system tray indicates the date as 02-04-2025.

## **2. PostgreSQL database se connect karne ke liye ye command kuch istarha hogi**

```
psql -h faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -U FaisalKhan -d postgres
```

**YE KUCH ISTARHA LAGEGA**



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command entered was:

```
psql -h faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -U FaisalKhan -d postgres
```

The output shows the PostgreSQL prompt:

```
psql (17.4 (Debian 17.4-1.pgdget20-2), server 17.2)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgres)
Type "help" for help.

postgres>
```

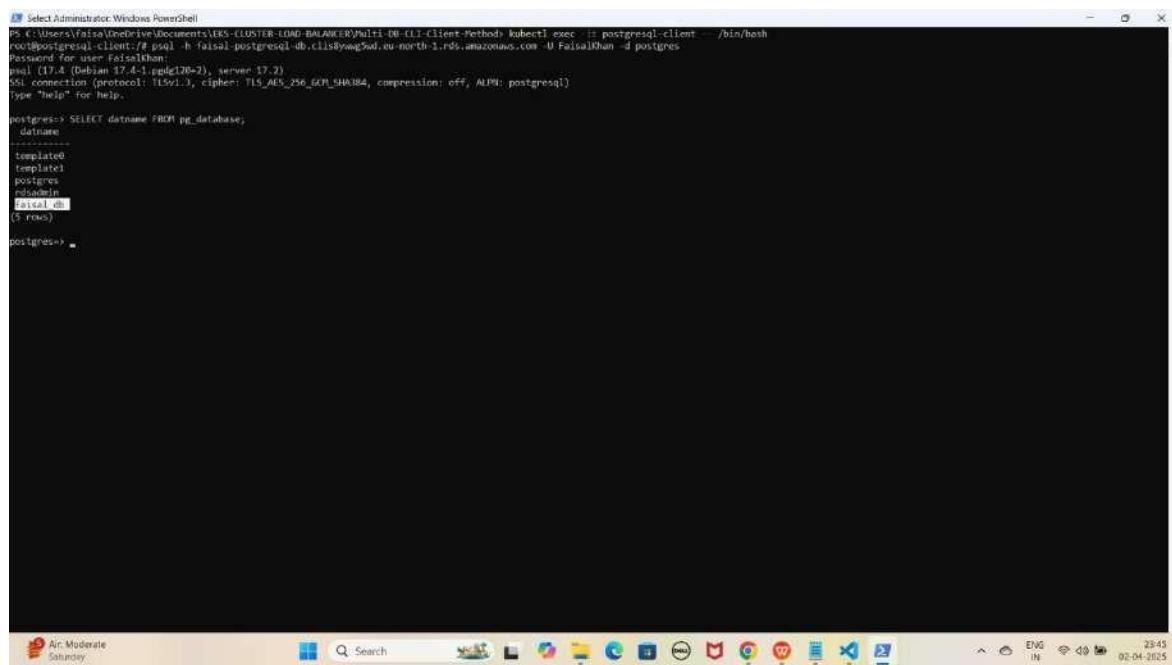
The taskbar at the bottom shows various icons and the date/time: 02-04-2025.

**Note: Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke rds-secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.**

### **3. Databases check karne ke liye ye command run kariye**

```
SELECT datname FROM pg_database;
```

**YE KUCH ISTARHA LAGEGA**



A screenshot of a Windows PowerShell window titled "Select Administrator: Windows PowerShell". The command entered is "SELECT datname FROM pg\_database;". The output shows five database entries: template0, template1, postgres, pgbench, and Faisal\_db. The command "postgres> ." is shown at the bottom.

```
PS C:\Users\Faisal\OneDrive\Documents\LEARN-BALANCER\Multi-DB CLI Client\Methods\kubert1 exec ./postgresql-client /bin/bash
root@postgres:~# psql -h faisal postgresql-db.clsrwmg5wd.eu.north-1.ress.amazonaws.com -U FaisalKhan -d postgres
Password for user FaisalKhan:
psql (17.4 (Debian 17.4-1.pgdg20-2), server 17.3)
SSL connection (protocol: tls1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgresql)
Type "help" for help.

postgres=> SELECT datname FROM pg_database;
 datname
 -----
 template0
 template1
 postgres
 pgbench
 Faisal_db
(5 rows)

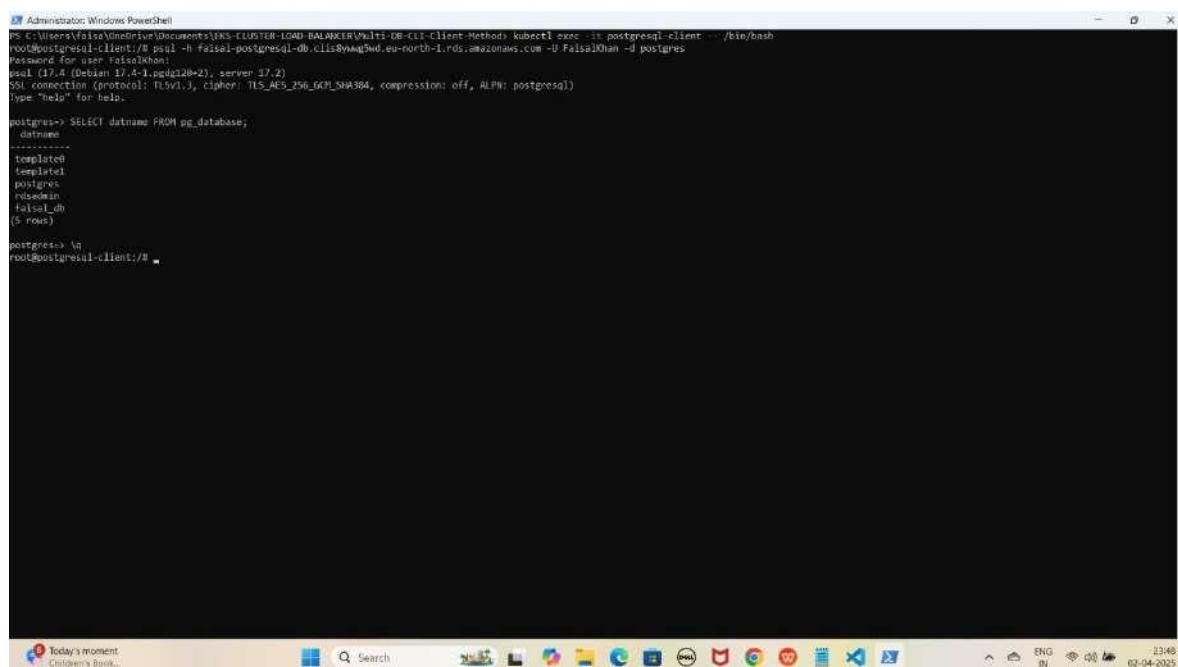
postgres=> .
```

**NOTE: Yahan hum dobara database create nahi karenge, kyunki pehle hi Method 1 mein create kar chuke hain. Hum sirf ye check karenge ki database available hai ya nahi.**

#### 4. PostgreSQL database se exit hoagaye exit hone ke liye exit type kariye

\q

YE KUCH ISTARHA LAGEGA



```
Administrator: Windows PowerShell
PS C:\Users\faisal\OneDrive\Documents\AWS-CLUSTER-LOAD-BALANCER\multi-00-CLI-client\Method1> kubectl exec -it postgresql-client -- /bin/bash
root@postgresql-client:/# psql -h faisal-postgresql-db.cis8wmg5wd.eu-north-1.rds.amazonaws.com -U FaisalKhan -d postgres
Password for user FaisalKhan:
psql (17.4 (Debian 17.4.1-0ubuntu2), server 17.2)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgres)
Type "help" for help.

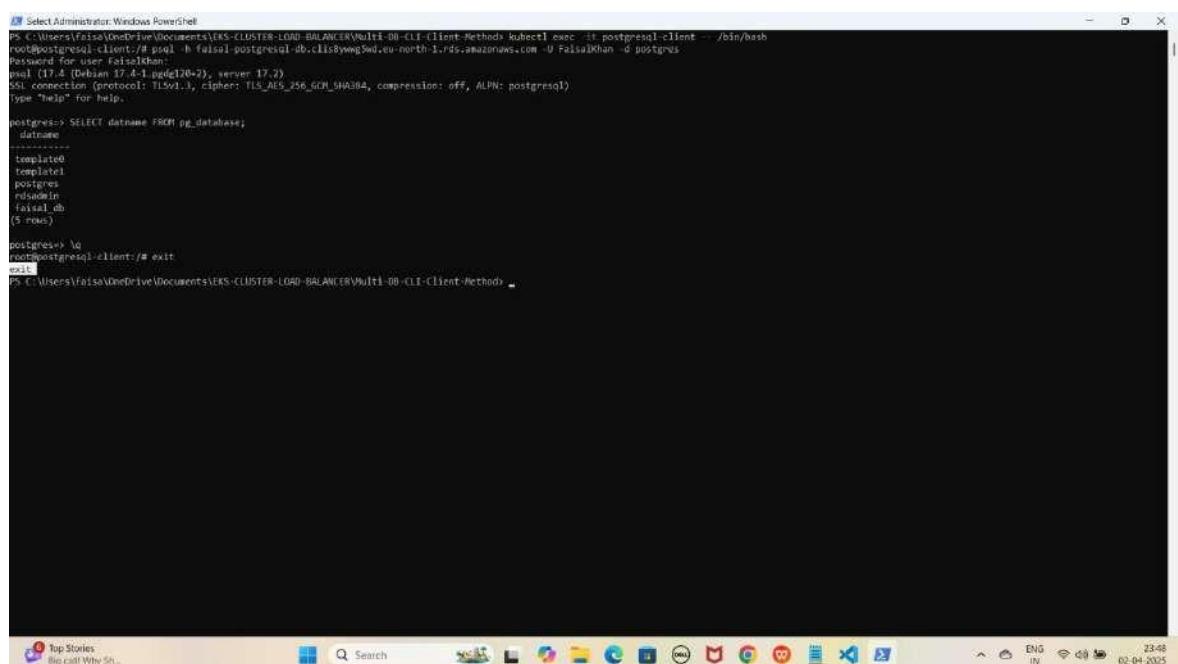
postgres=> SELECT datname FROM pg_database;
 datname
-----
 template0
 template1
 postgres
 faisal
 (4 rows)

postgres=> \q
root@postgresql-client:/#
```

#### 5. PostgreSQL client pod se bhi exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
Select Administrator: Windows PowerShell
PS C:\Users\faisal\OneDrive\Documents\AWS-CLUSTER-LOAD-BALANCER\multi-00-CLI-client\Method1> kubectl exec -it postgresql-client -- /bin/bash
root@postgresql-client:/# psql -h faisal-postgresql-db.cis8wmg5wd.eu-north-1.rds.amazonaws.com -U FaisalKhan -d postgres
Password for user FaisalKhan:
psql (17.4 (Debian 17.4.1-0ubuntu2), server 17.2)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgres)
Type "help" for help.

postgres=> SELECT datname FROM pg_database;
 datname
-----
 template0
 template1
 postgres
 faisal
 (4 rows)

postgres=> \q
root@postgresql-client:/# exit
exit
PS C:\Users\faisal\OneDrive\Documents\AWS-CLUSTER-LOAD-BALANCER\multi-00-CLI-client\Method1>
```

# Part 9: Accessing Amazon RDS (MySQL, Oracle, PostgreSQL) from an EKS Cluster (Without Hardcoded Endpoint)

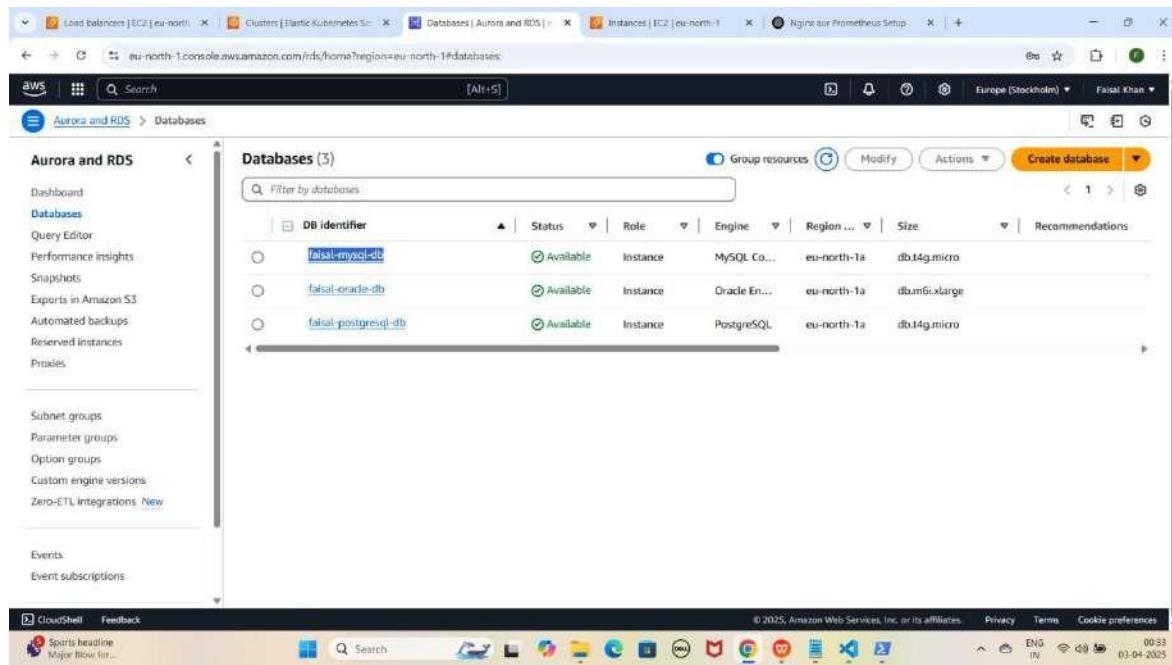
## Method 1: Application ke through Database Access Karna (Without Hardcoded Endpoint)

### Step 1: external-name.yaml file ka kaam

Jab bhi humein **RDS** ko access karna hota hai, toh humein har baar uska **endpoint** dena padhta hai, jo kafi **time-consuming** aur **hardcoded** ho jata hai. Is issue ko solve karne ke liye hum **ExternalName service** ka use kar rahe hain. Yeh method easy hai aur har baar **RDS ka endpoint** type karne ki zaroorat nahi hoti. Bas **external name service** apply karni hai aur **short name** se **RDS connect** kar sakte hain.

First aapko aapke **RDS Databases** ke **endpoints** ko **external-name.yaml** me **update** karna hogा

YE KUCH ISTARHA LAGEGA



The screenshot shows the AWS RDS console with the URL [eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases](https://eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases). The left sidebar is titled 'Aurora and RDS' and includes links for Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, and Zero-ETL integrations. The main content area is titled 'Databases (3)' and lists three database instances:

DB identifier	Status	Role	Engine	Region	Size
faisal-mysql-db	Available	Instance	MySQL Co...	eu-north-1a	db.t4g.micro
faisal-oracle-db	Available	Instance	Oracle En...	eu-north-1a	db.m6i.xlarge
faisal-postgresql-db	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro

**faisal-mysql-db**

**Summary**

DB identifier faisal-mysql-db	Status Available	Role Instance	Engine MySQL Community	Recommendations
CPU 2.03%	Class db.t4g.micro	Current activity 1 Connections	Region & AZ eu-north-1a	

**Connectivity & security**

**Endpoint copied**

Port 3306	Networking	Security
Availability Zone eu-north-1a	VPC vpc-0e345f19effbe822	VPC security groups ALL-TRAFFIC-ALLOW (sg-0ea98265816a2bbec)
Subnet group faisal-subnet	Subnets	Publicly accessible No
		Certificate authority rds-ca-m-2048-q1

```
Dockerfile
rds-config.yaml
rds-secrets.yaml
external-name.yaml
```

```
1  apiVersion: V1
2  kind: Service
3  metadata:
4    name: faisal-mysql
5  spec:
6    type: ExternalName
7    externalName: faisal-mysql-db.ellsbywg5wd.eu-north-1.rds.amazonaws.com
8
9  apiVersion: V1
10 kind: Service
11 metadata:
12   name: faisal-oracle
13 spec:
14   type: ExternalName
15   externalName: faisal-oracle-db.ellsbywg5wd.eu-north-1.rds.amazonaws.com
16
17  apiVersion: V1
18  kind: Service
19  metadata:
20   name: faisal-postgresql
21  spec:
22   type: ExternalName
23   externalName: faisal-postgresql-db.ellsbywg5wd.eu-north-1.rds.amazonaws.com
24
```

Screenshot of the AWS RDS console showing the 'Databases' page. The left sidebar shows 'Aurora and RDS' with 'Databases' selected. The main area displays three databases: 'faisal-mysql-db' (MySQL), 'faisal-oracle-db' (Oracle), and 'faisal-postgresql-db' (PostgreSQL). Each database row includes columns for DB identifier, Status, Role, Engine, Region, and Size.

DB identifier	Status	Role	Engine	Region	Size
faisal-mysql-db	Available	Instance	MySQL Co...	eu-north-1a	db.t4g.micro
faisal-oracle-db	Available	Instance	Oracle En...	eu-north-1a	db.m6xlarge
faisal-postgresql-db	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro

Screenshot of the AWS RDS console showing the 'Database Details' page for 'faisal-oracle-db'. The left sidebar shows 'Aurora and RDS' with 'Databases' selected. The main area displays summary information for the database, including DB identifier, Status, Role, Engine, and Region. Below the summary, there are tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, Tags, and Recommendations. The 'Connectivity & security' tab is active, showing the endpoint details: port 1521 and VPC connection to 'faisal-oracle-db.clsbywwq5wd.eu-north-1.rds.amazonaws.com'.

DB identifier	Status	Role	Engine	Recommendations
faisal-oracle-db	Available	Instance	Oracle Enterprise Edition	

The screenshot shows a terminal window with several tabs open, displaying Kubernetes YAML configuration files. The files define services and external names for MySQL and Oracle databases.

```
apiVersion: v1
kind: Service
metadata:
  name: faisal-mysql
spec:
  type: ExternalName
  externalName: faisal-mysql-db.clsbywq5ed.eu-north-1.rds.amazonaws.com
  ports:
    - port: 3306
      targetPort: 3306
      protocol: TCP
      name: mysql
  selector:
    app: mysql

apiVersion: v1
kind: Service
metadata:
  name: faisal-oracle
spec:
  type: ExternalName
  externalName: faisal-oracle-db.clsbywq5ed.eu-north-1.rds.amazonaws.com
  ports:
    - port: 1521
      targetPort: 1521
      protocol: TCP
      name: oracle
  selector:
    app: oracle

apiVersion: v1
kind: Service
metadata:
  name: faisal-postgresql
spec:
  type: ExternalName
  externalName: faisal-postgresql-db.clsbywq5ed.eu-north-1.rds.amazonaws.com
  ports:
    - port: 5432
      targetPort: 5432
      protocol: TCP
      name: postgresql
  selector:
    app: postgresql
```

The screenshot shows the AWS Aurora and RDS Databases page. On the left, there's a sidebar with links like Dashboard, Databases, Query Editor, Performance Insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL Integrations, Events, and Event subscriptions. The main area has a search bar and a 'Databases (3)' title. A table lists three databases:

DB identifier	Status	Role	Engine	Region	Size
faisal-mysql-db	Available	Instance	MySQL Co...	eu-north-1a	db.t4g.micro
faisal-oracle-db	Available	Instance	Oracle En...	eu-north-1a	db.m6i.xlarge
faisal-postgresql-db	Available	Instance	PostgreSQL	eu-north-1a	db.t4g.micro

The screenshot shows the AWS RDS console with the 'faisal-postgresql-db' database selected. The main summary pane displays the database identifier as 'faisal-postgresql-db', status as 'Available', engine as 'PostgreSQL', and region as 'eu-north-1a'. The 'Connectivity & security' tab is active, showing the endpoint copied to 'faisal-postgresql-db.clsBywwg5wd.eu-north-1.rds.amazonaws.com' on port 5432. The networking section shows the VPC as 'vpc-0e345f19effbea822' and the subnet group as 'faisal-subnet'. The security section lists the VPC security group as 'ALL-TRAFFIC-ALLOW (sg-0eef8265816a28ed)' and the certificate authority as 'rds-ca-rsa2048-q1'. The left sidebar shows various RDS management options like Dashboard, Databases, and Query Editor.

The screenshot shows a code editor displaying a YAML configuration file for an RDS cluster load balancer. The file defines three services: 'faisal-mysql' (version V1), 'faisal-oracle' (version V1), and 'faisal-postgresql' (version V1). Each service has its own metadata and spec sections, including an external name that maps to the RDS endpoint. The code editor interface includes tabs for Dockerfile, Dockerfile, rds-config.yaml, rds-secrets.yaml, and external-name.yaml. The status bar at the bottom indicates the file has 23 lines and 52 selected characters, and the encoding is UTF-8.

```
1: apiVersion: v1
2: kind: Service
3: metadata:
4:   name: faisal-mysql
5:   spec:
6:     type: ExternalName
7:     externalName: faisal-mysql-db.clsBywwg5wd.eu-north-1.rds.amazonaws.com
8:
9: ---
10: apiVersion: v1
11: kind: Service
12: metadata:
13:   name: faisal-oracle
14:   spec:
15:     type: ExternalName
16:     externalName: faisal-oracle-db.clsBywwg5wd.eu-north-1.rds.amazonaws.com
17:
18: ---
19: apiVersion: v1
20: kind: Service
21: metadata:
22:   name: faisal-postgresql
23:   spec:
24:     type: ExternalName
25:     externalName: faisal-postgresql-db.clsBywwg5wd.eu-north-1.rds.amazonaws.com
```

## JAISE KI:-

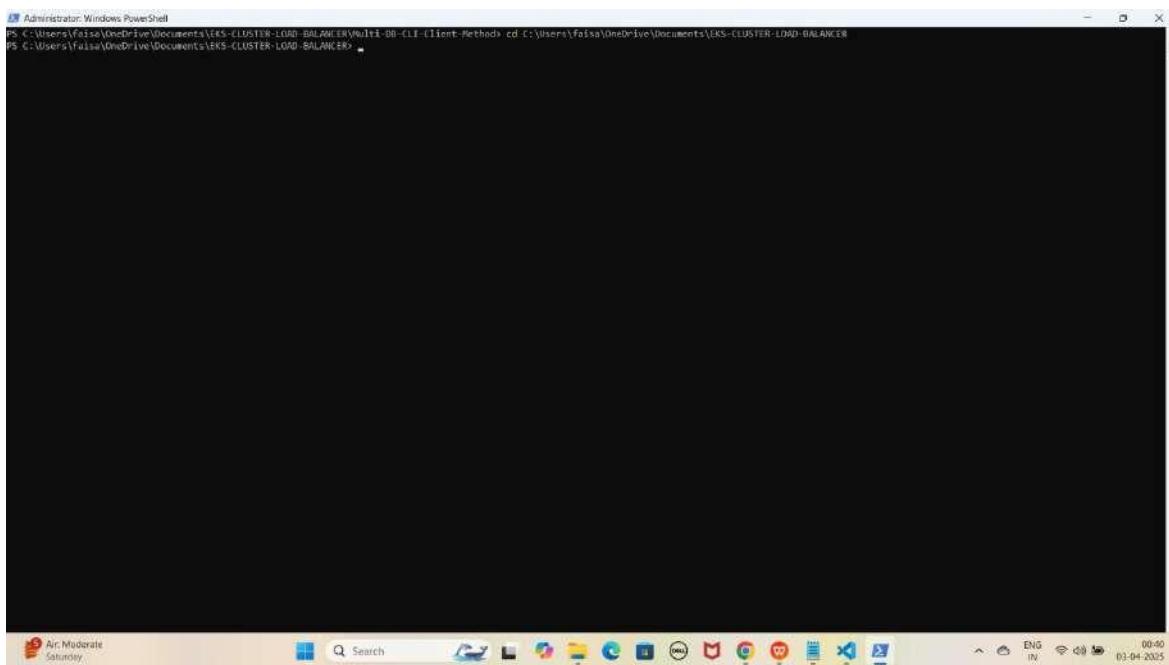
```
apiVersion: v1
kind: Service
metadata:
  name: faisal-mysql
spec:
  type: ExternalName
  externalName: faisal-mysql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com
---
apiVersion: v1
kind: Service
metadata:
  name: faisal-oracle
spec:
  type: ExternalName
  externalName: faisal-oracle-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com
---
apiVersion: v1
kind: Service
metadata:
  name: faisal-postgresql
spec:
  type: ExternalName
  externalName: faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com
```

**NOTE: Agar aap apna naam use kar rahe hain, to phir commands bhi usi ke according apply karni hongi.**

## 1. Location par jao

```
cd C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER
```

YE KUCH ISTARHA LAGEGA

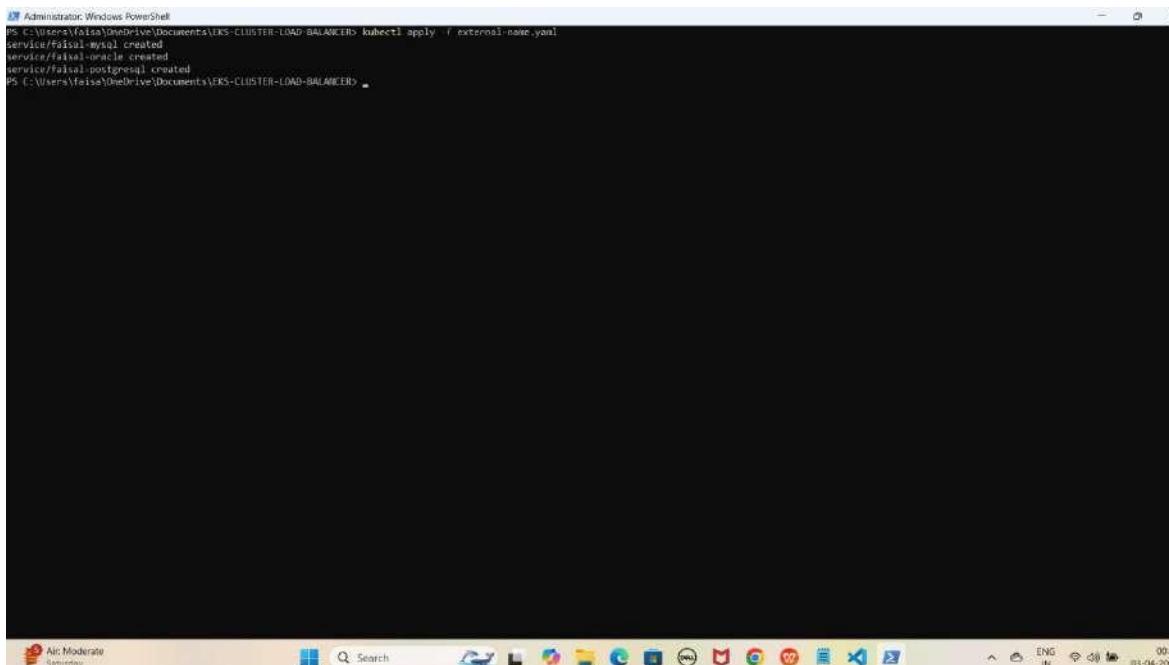


```
Administrator: Windows PowerShell
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> cd C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## External Name Service Apply Karo

```
kubectl apply -f external-name.yaml
```

YE KUCH ISTARHA LAGEGA



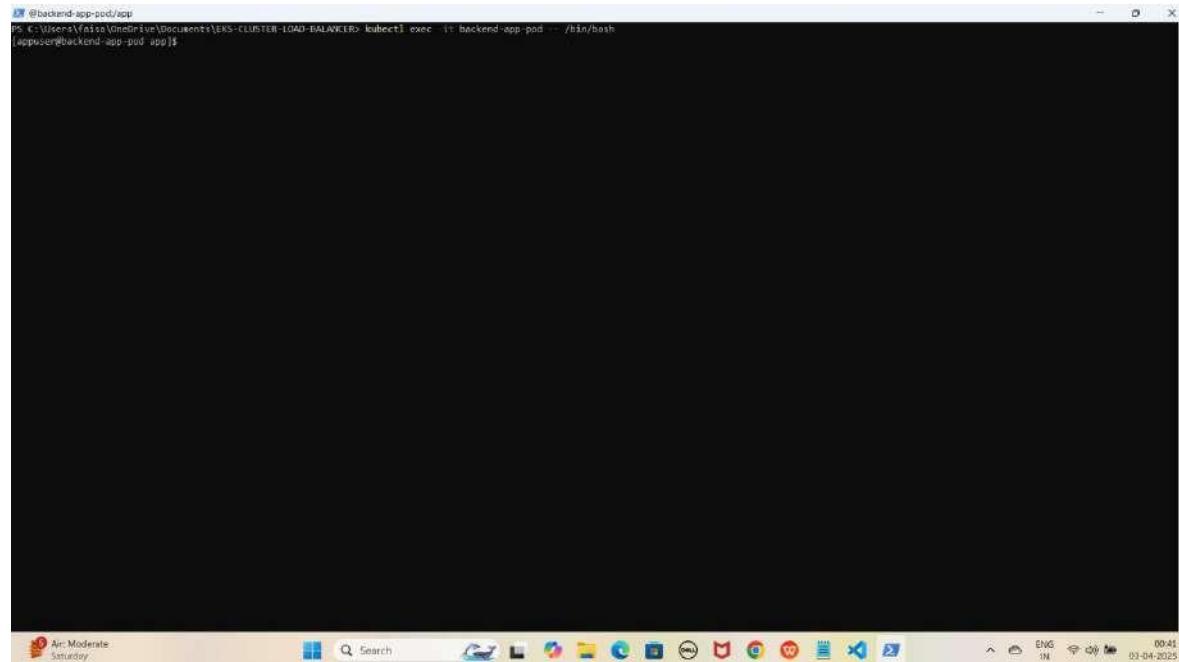
```
Administrator: Windows PowerShell
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f external-name.yaml
service/faisal-mysql created
service/faisal-oracle created
service/faisal-postgresql created
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## Step 2: Amazon RDS (MySQL) Database Access Karo-Method 1

### 1. Pod ke andar jane ke liye ye command run karo

```
kubectl exec -it backend-app-pod -- /bin/bash
```

YE KUCH ISTARHA LAGEGA

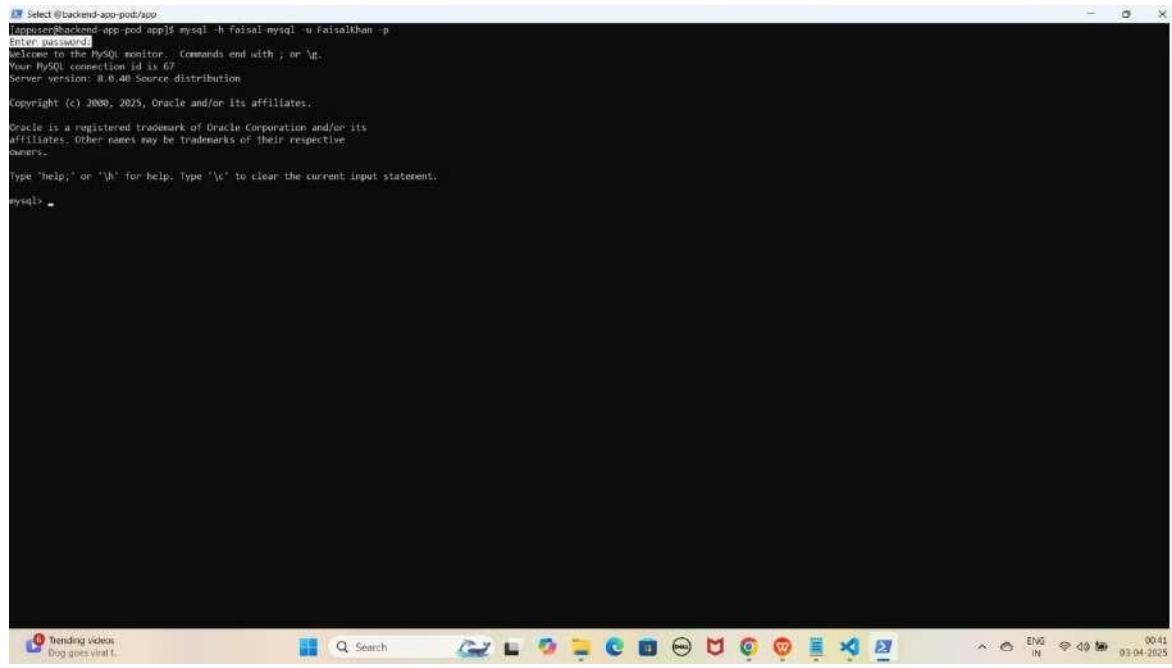


The screenshot shows a Windows command-line interface (CMD) window. The title bar reads '@backend-app-pod/app'. The command entered is 'kubectl exec -it backend-app-pod -- /bin/bash'. The prompt shows the user is now in the pod's bash shell, indicated by '(appuser@backend-app-pod app)\$'. The rest of the window is a solid black color.

## 2. MySQL database se connect karne ke liye ye command kuch istarha hogi

```
mysql -h faisal-mysql -u FaisalKhan -p
```

### YE KUCH ISTARHA LAGEGA



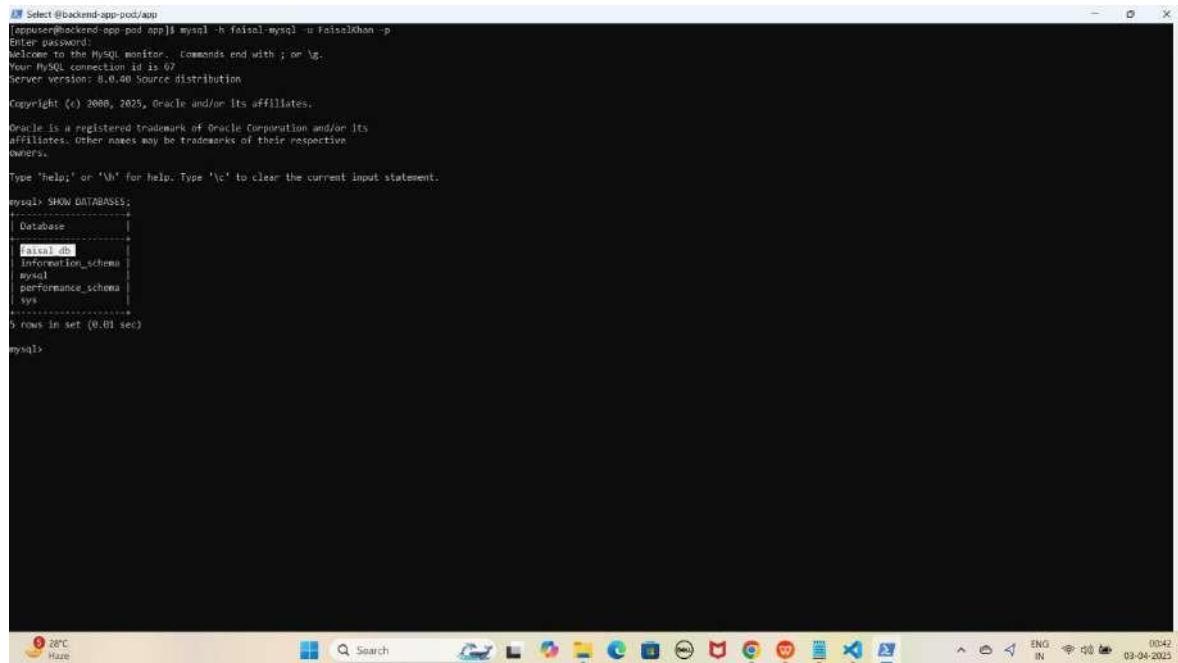
The screenshot shows a terminal window titled 'Select @backend-app-pod/app'. The command entered is 'mysql -h faisal-mysql -u FaisalKhan -p'. The MySQL monitor prompt is visible, showing connection details like 'Welcome to the MySQL monitor. Commands end with ; or \g.', 'Your MySQL connection id is 7', and 'Server version: 8.0.40 Source distribution'. It also displays copyright information from Oracle and trademarks. The MySQL prompt 'mysql>' is at the bottom.

**Note:** Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke rds-secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.

### 3. Databases check karne ke liye ye command run kariye

SHOW DATABASES;

YE KUCH ISTARHA LAGEGA



```
[Select@Backend-app-pod/app]$ mysql -h faisal mysql -u faisalchon -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 67
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

Type 'help' or 'man' for help. Type 'quit' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| faisal_db          |
| information_schema |
| mysql               |
| performance_schema |
+--------------------+
0 rows in set (0.01 sec)

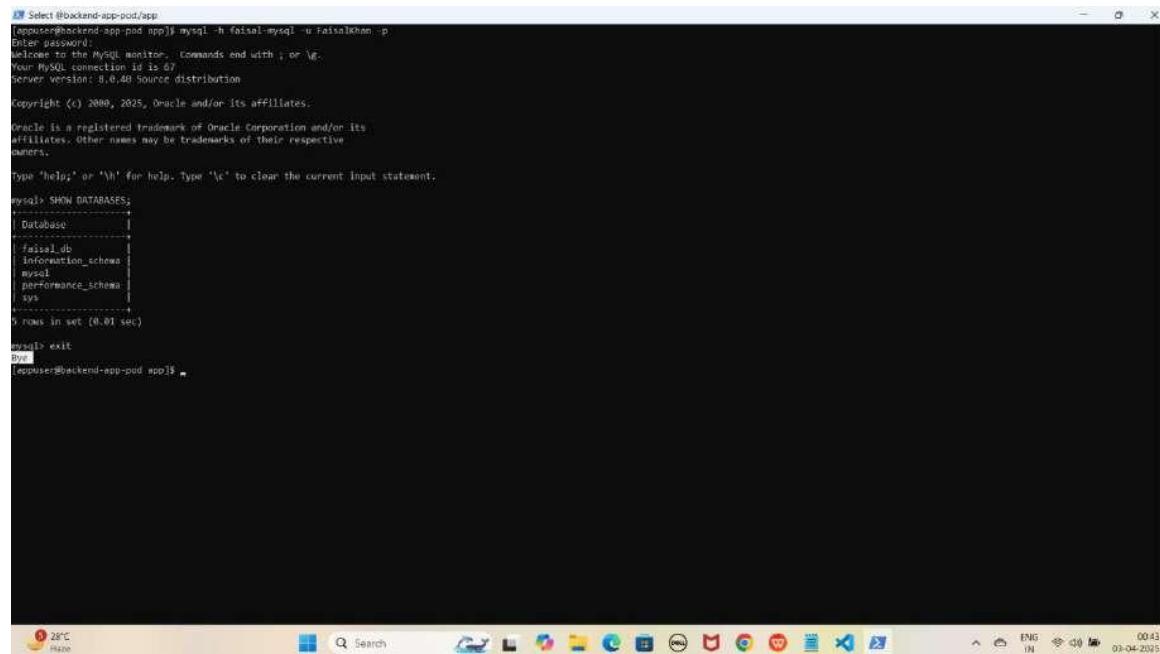
mysql>
```

**NOTE:** Yahan par hum dobara database create nahi karenge, kyunki pehle Method 1 mein create kar chuke hain. Sirf endpoint replace karne ke liye humne ExternalName service ka use kiya hai.

#### 4. Ab MYSQL database se exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



The screenshot shows a terminal window titled "Select @Backend-app-pod/app" running on a Windows operating system. The window displays the MySQL monitor interface. The user has connected to the MySQL server using the command "mysql -h Faisalkhan -u Faisalkhan -p". The MySQL prompt is visible, and the user has run the "SHOW DATABASES;" command, which lists the databases: "faisal\_db", "information\_schema", "mysql", "performance\_schema", and "sys". The user then types "exit" to exit the MySQL monitor, and the terminal returns to the command prompt. The taskbar at the bottom of the screen shows various application icons, and the system tray indicates the date and time as 03-04-2025.

```
[appuser@backend-app-pod app]$ mysql -h Faisalkhan -u Faisalkhan -p
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 67
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| faisal_db          |
| information_schema |
| mysql               |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.01 sec)

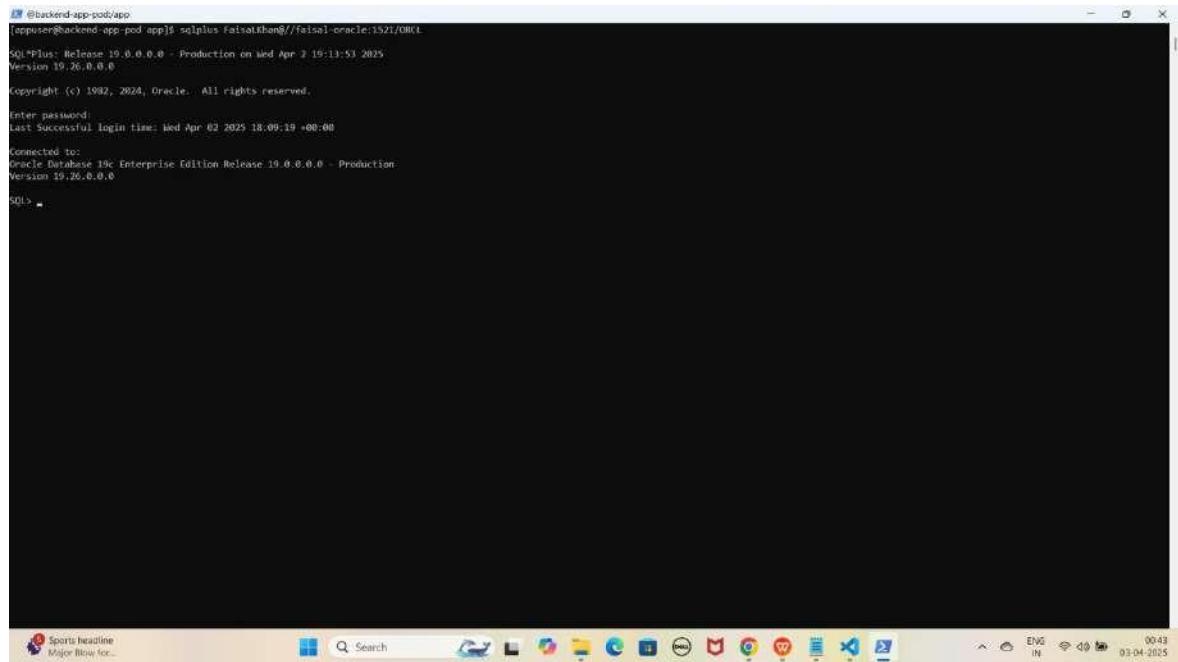
mysql> exit
Bye
[appuser@backend-app-pod app]$
```

## **Step 3: Amazon RDS (Oracle) Database Access Karo-Method 1**

### **1. Oracle database se connect karne ke liye ye command kuch istarha hogi**

```
sqlplus FaisaLKhan@//faisal-oracle:1521/ORCL
```

YE KUCH ISTARHA LAGEGA



```
[appuser@backend-app-pod app]$ sqlplus FaisaLKhan@//faisal-oracle:1521/ORCL
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Apr 2 19:13:53 2025
Version 19.26.0.0.0
Copyright (c) 1982, 2024, Oracle. All rights reserved.

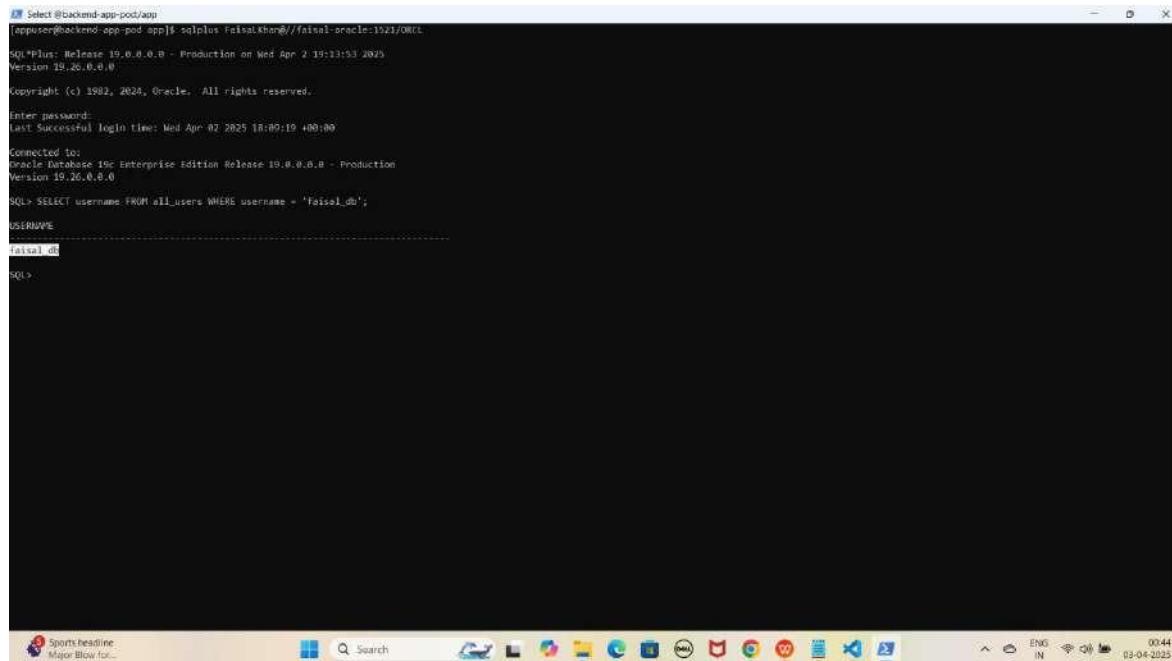
Enter password:
Last Successful login time: Wed Apr 02 2025 18:09:19 +00:00
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0
SQL>
```

**Note:** Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke rds-secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.

## 2. Databases check karne ke liye ye command run kariye

```
SELECT username FROM all_users WHERE username = 'faisal_db';
```

YE KUCH ISTARHA LAGEGA



```
[Select@backend-app-pod/app]# sqlplus FaisalKhan@//faisal-oracle:1521/ORECL
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Apr 2 19:13:53 2025
Version 19.20.0.0.0
Copyright (c) 1982, 2024, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed Apr 02 2025 18:09:19 +00:00

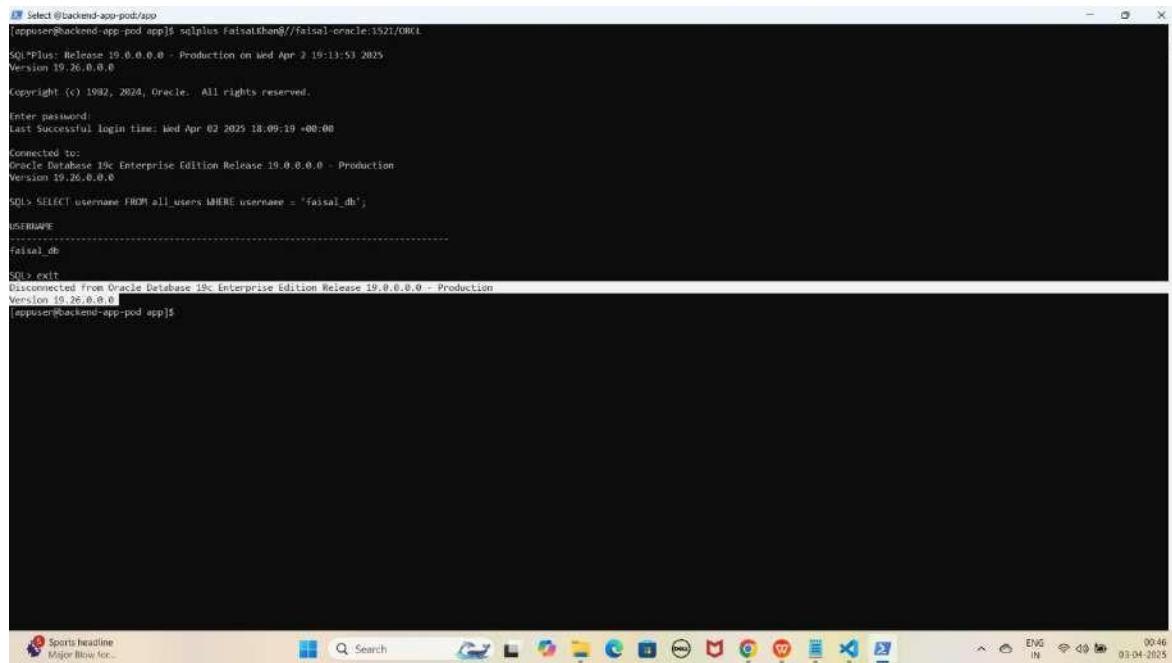
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.20.0.0.0
SQL> SELECT username FROM all_users WHERE username = 'faisal_db';
USERNAME
-----
faisal_db
SQL>
```

**NOTE: Yahan hum dobara database create nahi karenge, kyunki pehle hi Method 1 mein create kar chuke hain. Hum sirf ye check karenge ki database available hai ya nahi.**

### 3. Ab Oracle database se exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
[Select @backend-app-pod/app]# sqlplus FaisalKhan@//faisal-oracle:1521/ORCL
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Apr 2 19:13:53 2025
Version 19.26.0.0.0
Copyright (c) 1982, 2024, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed Apr 03 2025 18:09:19 +00:00

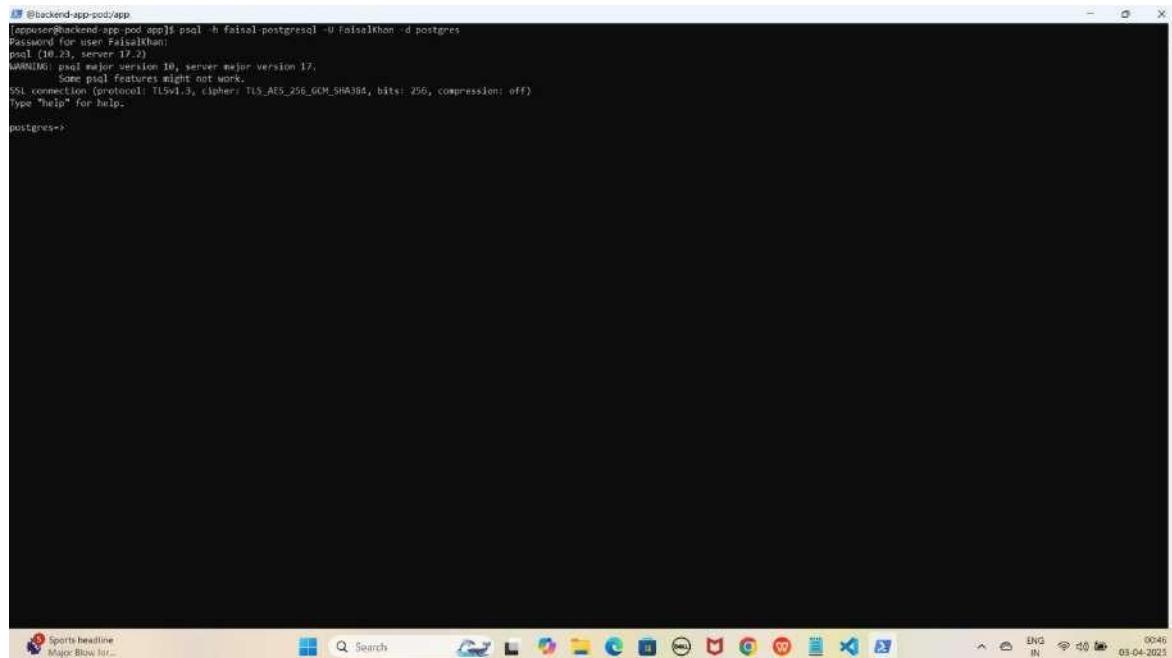
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0
SQL> SELECT username FROM all_users WHERE username = 'faisal_db';
USERNAME
-----
faisal_db
SQL> exit
Disconnected From Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0
[appuser@backend-app-pod/app]#
```

## **Step 4: Amazon RDS (PostgreSQL) Database Access Karo-Method 1**

**1. PostgreSQL database se connect karne ke liye ye command kuch istarha hogi**

```
psql -h faisal-postgresql -U FaisalKhan -d postgres
```

**YE KUCH ISTARHA LAGEGA**



```
[appuser@backend-app-pod app]$ psql -h faisal-postgresql -U Faisalkhan -d postgres
Password for user Faisalkhan:
psql (10.23, server 17.2)
WARNING: pgsql major version 10, server major version 17.
Some pgsql features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

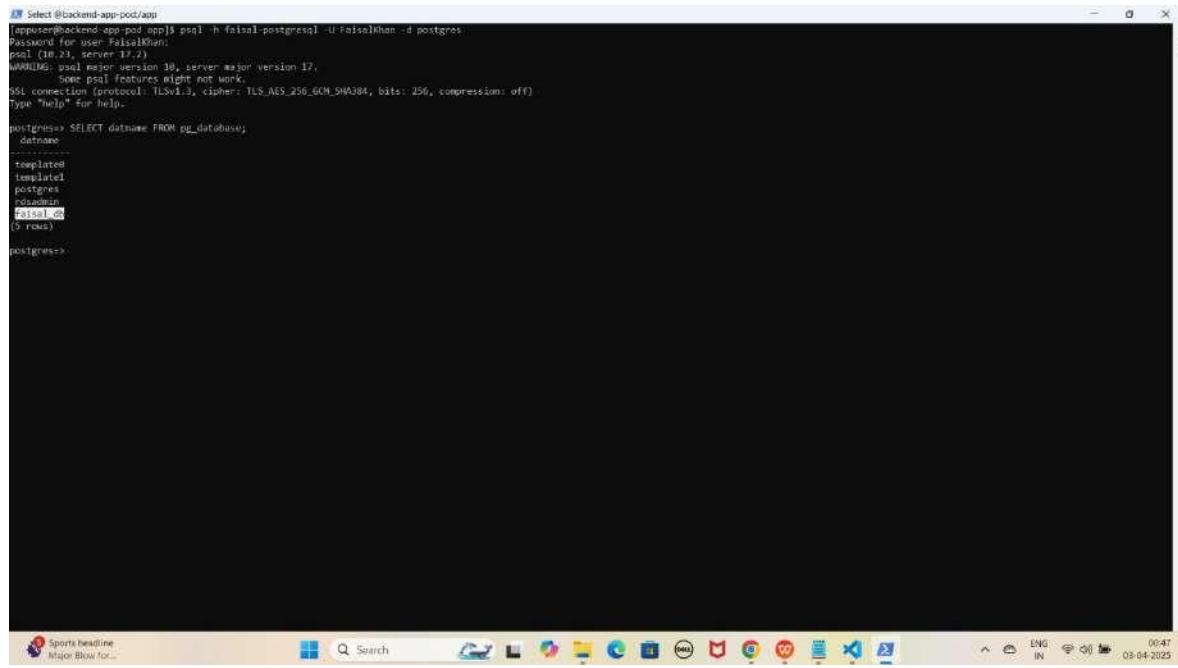
postgres>
```

**Note:** Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke rds-secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.

## 2. Databases check karne ke liye ye command run kariye

SELECT datname FROM pg\_database;

YE KUCH ISTARHA LAGEGA



```
[faisal@backend-app-pod app]$ psql -h faisal-postgresql -U faisalKhan -d postgres
Password for user faisalKhan:
psql (10.23, server 17.2)
WARNING: SSL connection features might not work.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres> SELECT datname FROM pg_database;
datname
-----
temp
template0
postgres
faisaladmin
faisal
(5 rows)

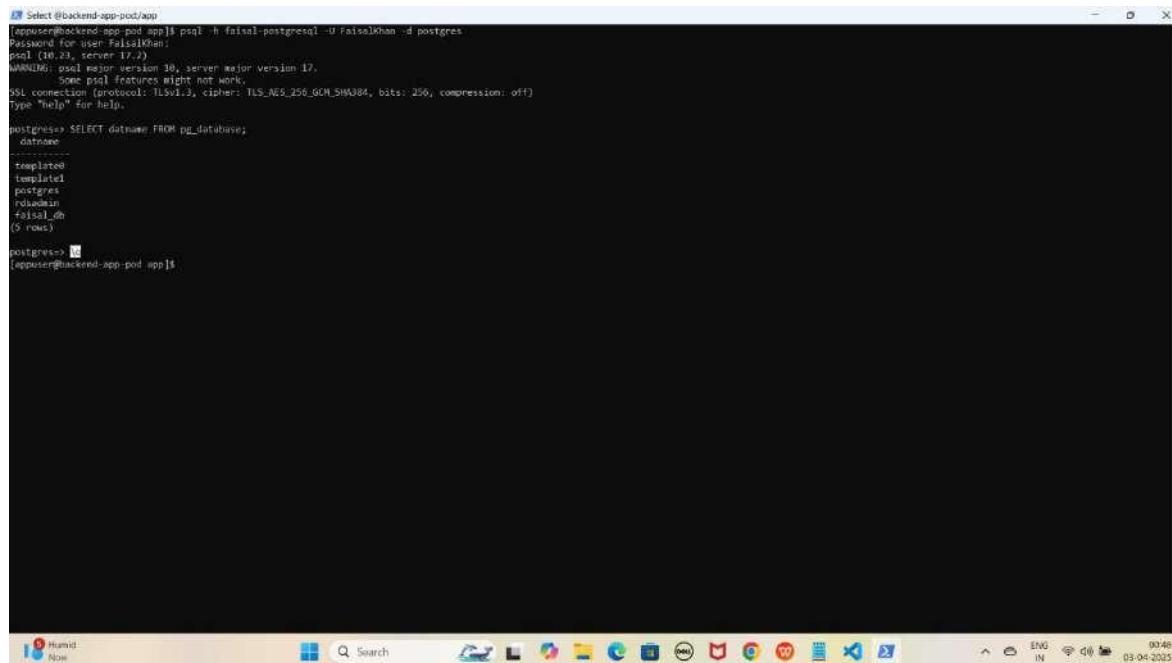
postgres>
```

**NOTE:** Yahan hum dobara database create nahi karenge, kyunki pehle hi Method 1 mein create kar chuke hain. Hum sirf ye check karenge ki database available hai ya nahi.

### **3. Ab PostgreSQL database se exit hoagaye exit hone ke liye exit type kariye**

\q

YE KUCH ISTARHA LAGEGA



```
[appuser@backend-app-pod app]$ psql -h faisal-postgresql -U faisalKhan -d postgres
Password for user faisalKhan:
psql (10.23, server 17.2)
WARNING: Using password on the command line interface can be insecure.
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

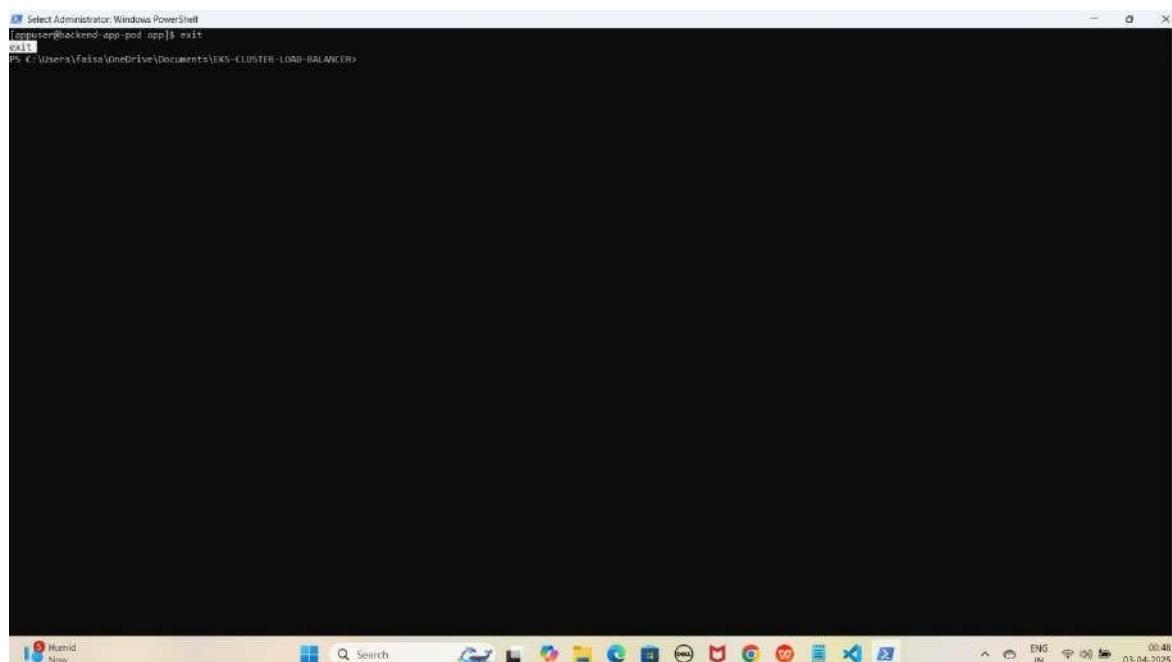
postgres> SELECT datname FROM pg_database;
 datname
-----
 template0
 template1
 postgres
 rdadmin
 faisal_db
(5 rows)

postgres> \q
[appuser@backend-app-pod app]$
```

### **4. Ab pod se bhi exit hoagaye exit hone ke liye exit type kariye**

exit

YE KUCH ISTARHA LAGEGA



```
[appuser@backend-app-pod app]$ exit
PS C:\Users\faisal\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER>
```

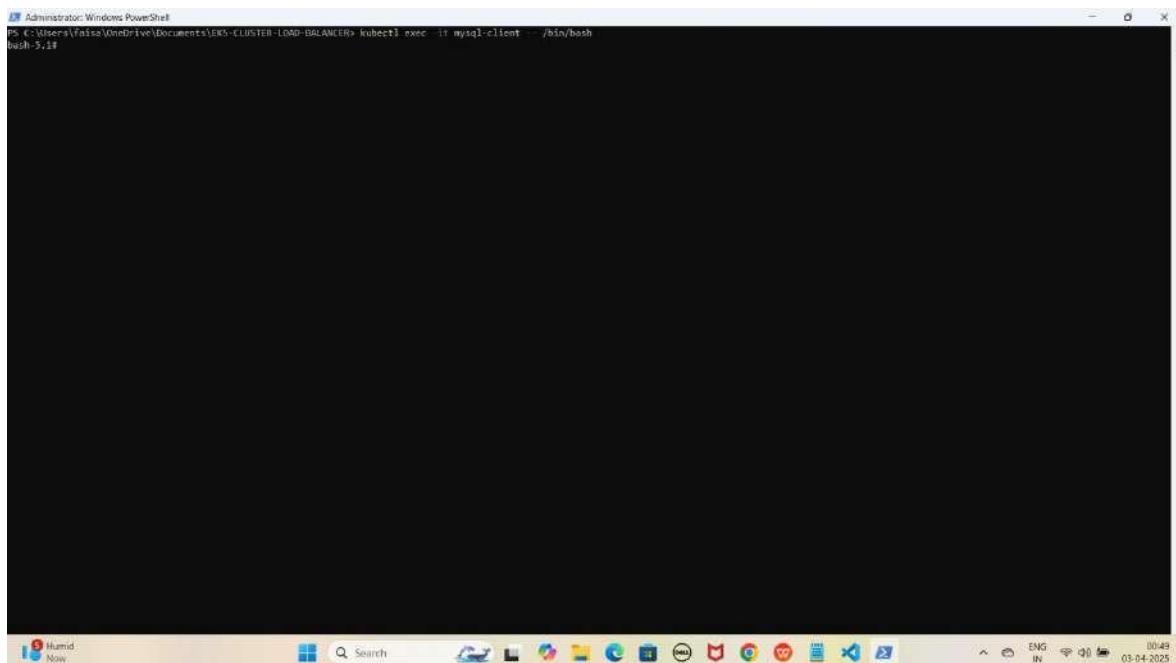
## **Method 2: Client Pods ke through Database Access Karna (Without Hardcoded Endpoint)**

### **Step 1: Amazon RDS (MySQL) Database Access Karo-Method 2**

#### **1. MySQL client pod me jane ke liye ye command run kaiye**

```
kubectl exec -it mysql-client -- /bin/bash
```

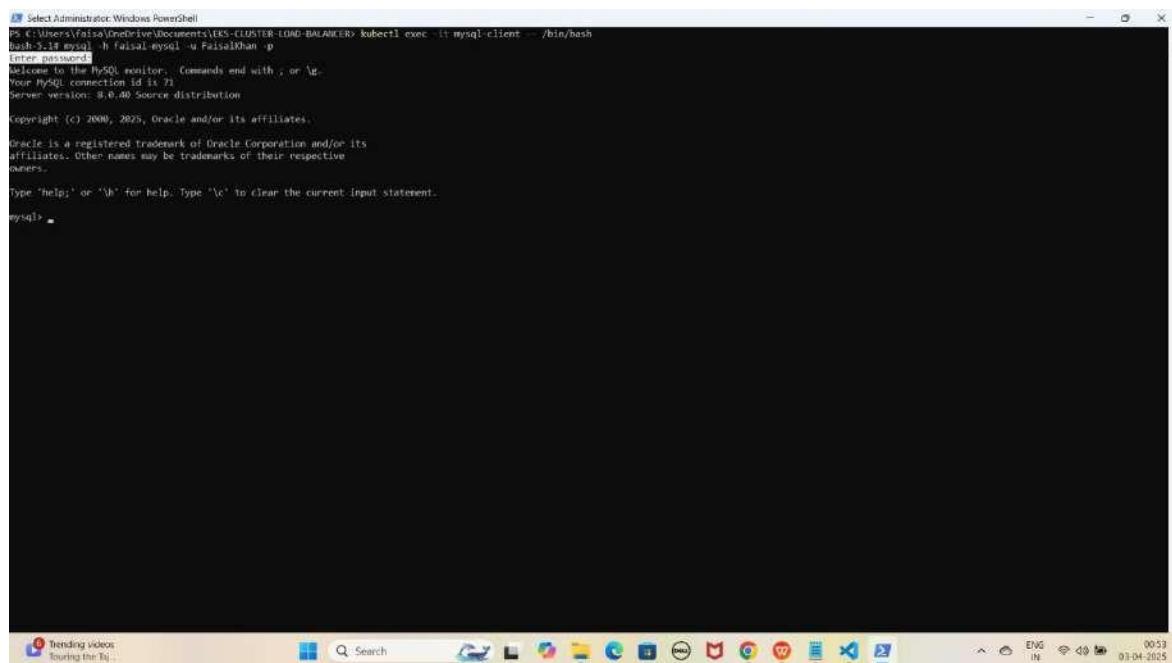
YE KUCH ISTARHA LAGEGA



## 2. MYSQL database se connect karne ke liye ye command kuch istarha hogi

```
mysql -h faisal-mysql -u FaisalKhan -p
```

### YE KUCH ISTARHA LAGEGA



The screenshot shows a Windows PowerShell window titled "Select Administrator: Windows PowerShell". The command entered is "mysql -h faisal-mysql -u FaisalKhan -p". The MySQL monitor prompt "mysql>" is visible at the bottom. The taskbar at the bottom of the screen shows various application icons.

```
PS C:\Users\faisa\OneDrive\Documents\LETS-CLUSTER-LB-MYSQL> mysql -h faisal-mysql -u FaisalKhan -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 73
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

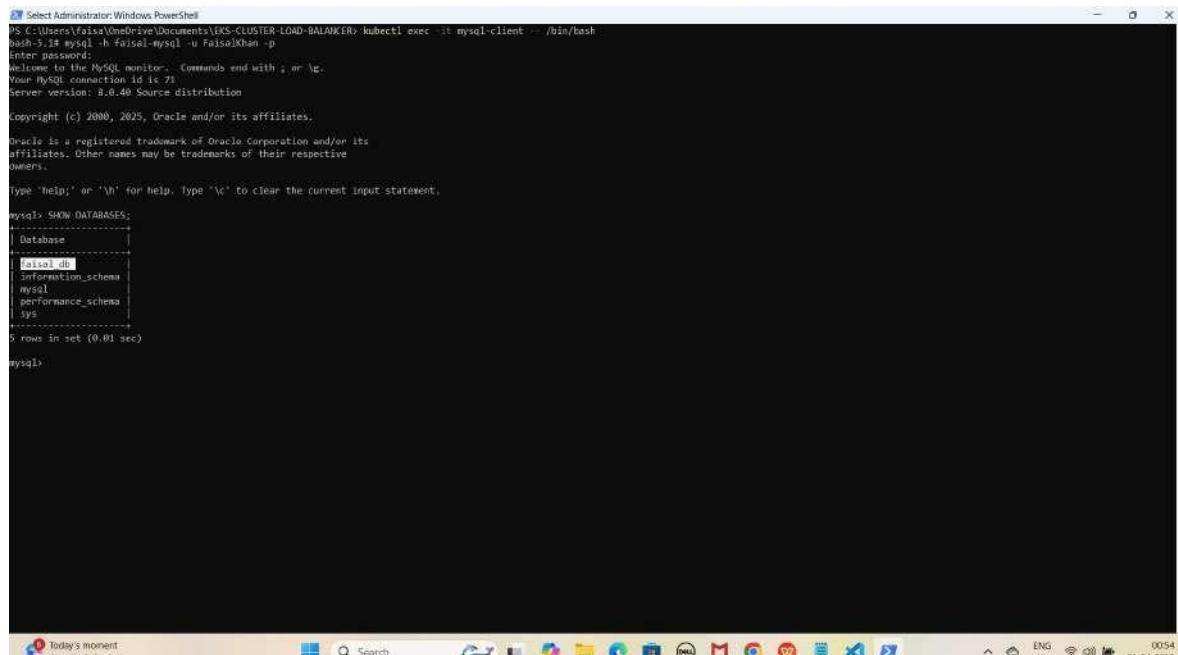
mysql> _
```

**Note:** Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke rds-secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.

### **3. Databases check karne ke liye ye command run kariye**

**SHOW DATABASES;**

**YE KUCH ISTARHA LAGEGA**



The screenshot shows a Windows PowerShell window titled "Select Administrator: Windows PowerShell". The command entered is "mysql -h Faisal-mysql -u Faisalkhan -p". The output shows the MySQL monitor welcome message, connection ID, server version, and copyright information. The "SHOW DATABASES;" command is run, displaying a list of databases: "faisal\_db", "information\_schema", "mysql", "performance\_schema", and "sys". The command "mysql>" is shown at the bottom.

```
PS C:\Users\Faisa\OneDrive\Documents\IKS-CLUSTER-LOAD-BALANCING>kubectl exec -it mysql-client -- /bin/bash
bash:5:if mysql -h Faisal-mysql -u Faisalkhan -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 78
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| faisal_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

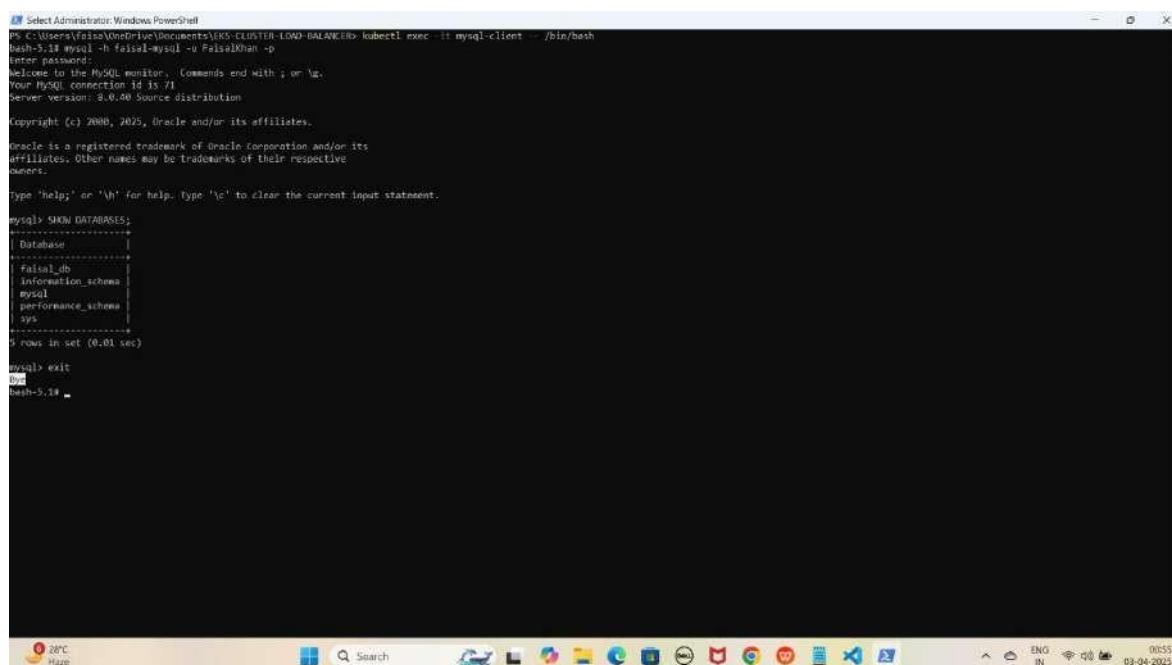
mysql>
```

**NOTE: Yahan hum dobara database create nahi karenge, kyunki pehle hi  
Method 1 mein create kar chuke hain. Hum sirf ye check karenge ki  
database available hai ya nahi.**

#### 4. MYSQL database se exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisal\OneDrive\Documents\ERS-CLUSTER-LOAD-BALANCER> kubectl exec -it mysql-client -- /bin/bash
bash-5.1# mysql -h faisal.mysql -u FaisalKhan -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 71
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

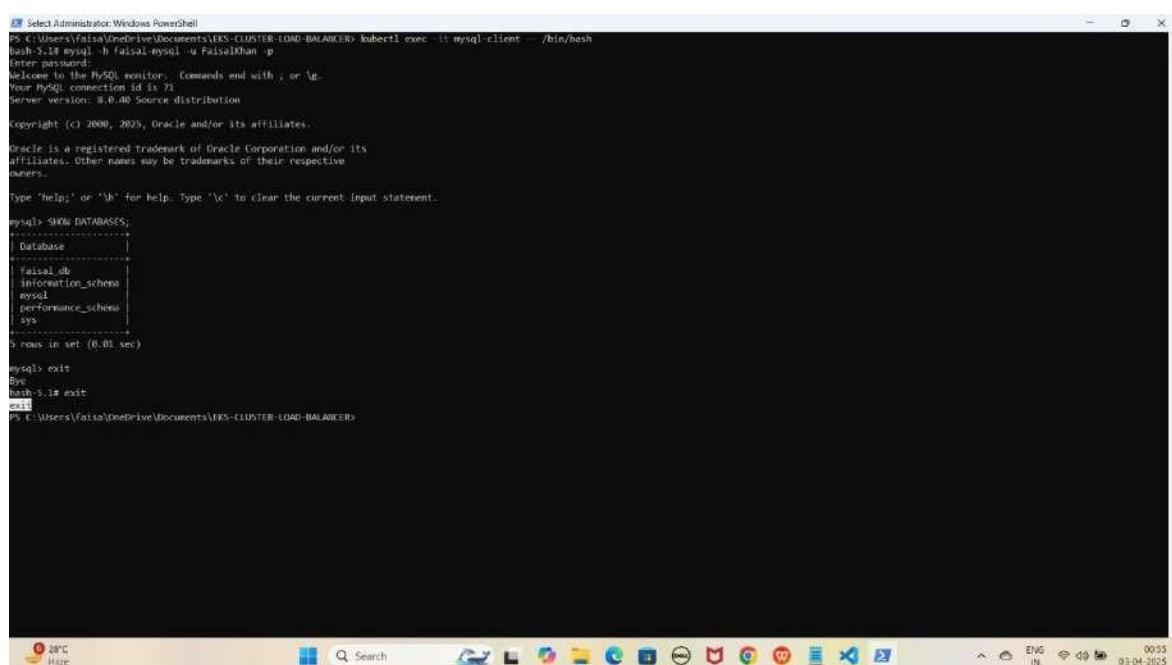
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| faisal_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> exit
Bye
bash-5.1#
```

#### 5. MySQL client pod se bhi exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisal\OneDrive\Documents\ERS-CLUSTER-LOAD-BALANCER> kubectl exec -it mysql-client -- /bin/bash
bash-5.1# mysql -h faisal.mysql -u FaisalKhan -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 71
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

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

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| faisal_db |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

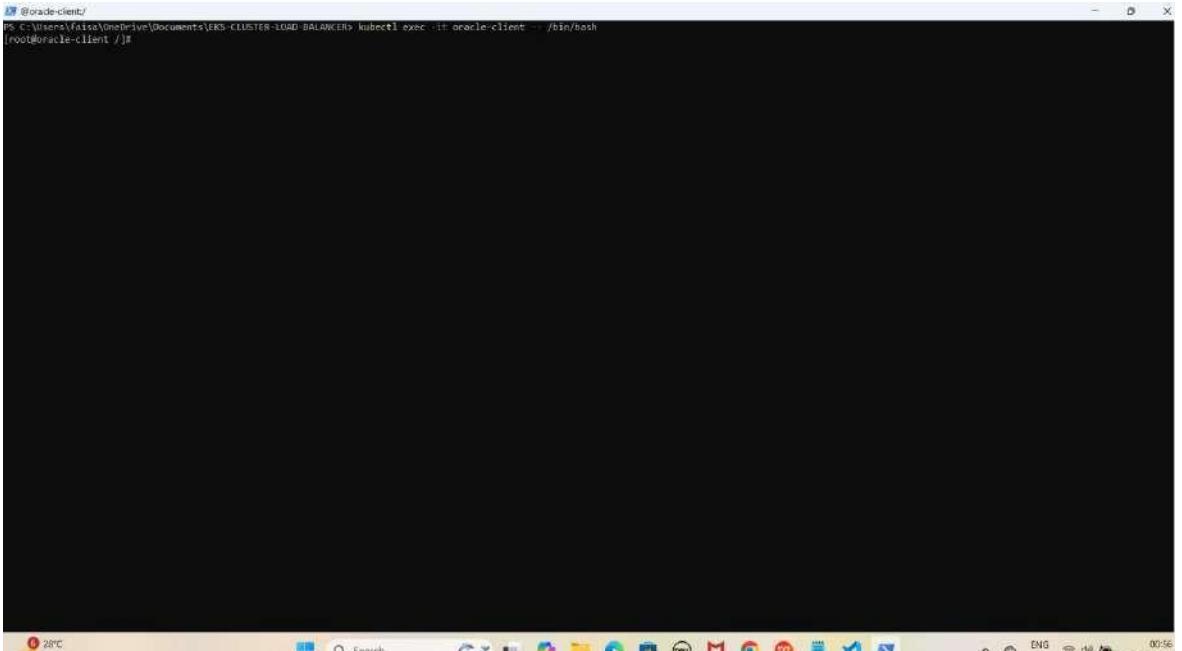
mysql> exit
Bye
bash-5.1# exit
exit
PS C:\Users\faisal\OneDrive\Documents\ERS-CLUSTER-LOAD-BALANCER>
```

## Step 2: Amazon RDS (Oracle) Database Access Karo-Method 2

### 1. Oracle client pod me jane ke liye ye command run kaiye

```
kubectl exec -it oracle-client -- /bin/bash
```

YE KUCH ISTARHA LAGEGA



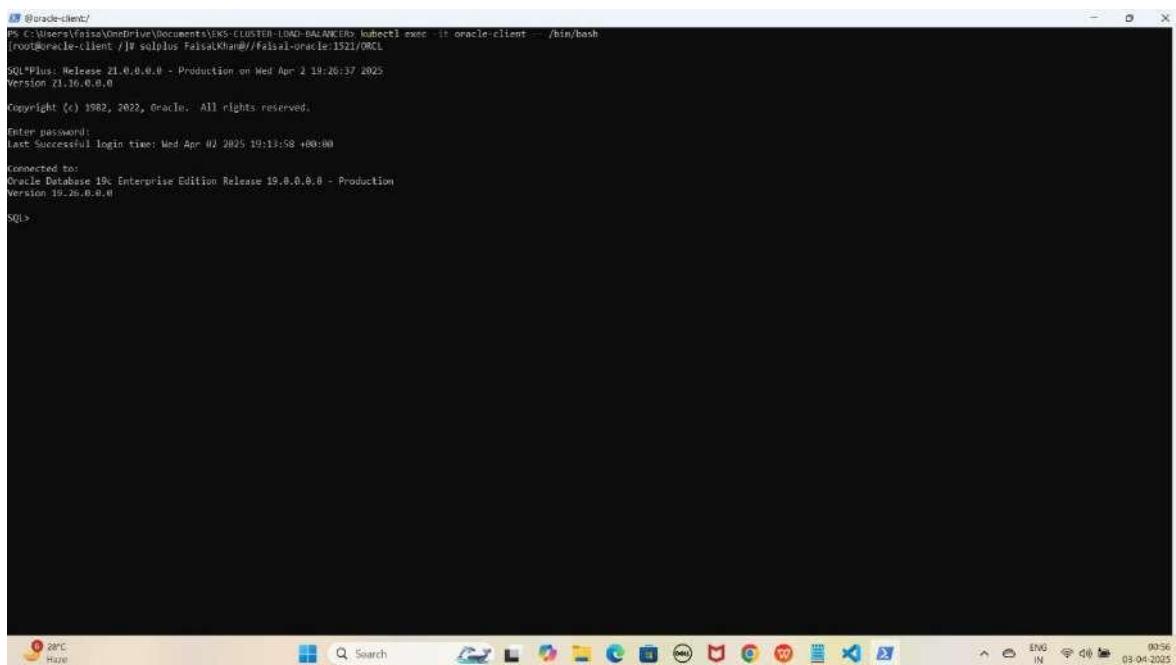
The screenshot shows a Windows terminal window titled '@oracle-client/'. The command entered is 'kubectl exec -it oracle-client -- /bin/bash'. The terminal is in a dark mode theme. The taskbar at the bottom shows various icons for applications like File Explorer, Edge, and FileZilla. The system tray indicates the date as 03-04-2025 and the time as 00:56.

```
@oracle-client/  
PS C:\Users\fatima\OneDrive\Documents\ERS-CLUSTER-LOAD-BALANCERS> kubectl exec -it oracle-client -- /bin/bash  
(root@oracle-client:~) #
```

## 2. Oracle database se connect karne ke liye ye command kuch istarha hogi

```
sqlplus FaisaLKhan@//faisal-oracle:1521/ORCL
```

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisa\OneDrive\Documents\KVS-CLUSTER-LOAD-BALANCERS>kubectl exec -it oracle-client -- /bin/bash
root@oracle-client:/# sqlplus FaisaLKhan@//faisal-oracle:1521/ORCL

SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 2 19:26:37 2025
Version 21.0.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed Apr 02 2025 19:13:58 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.0.0.0.0

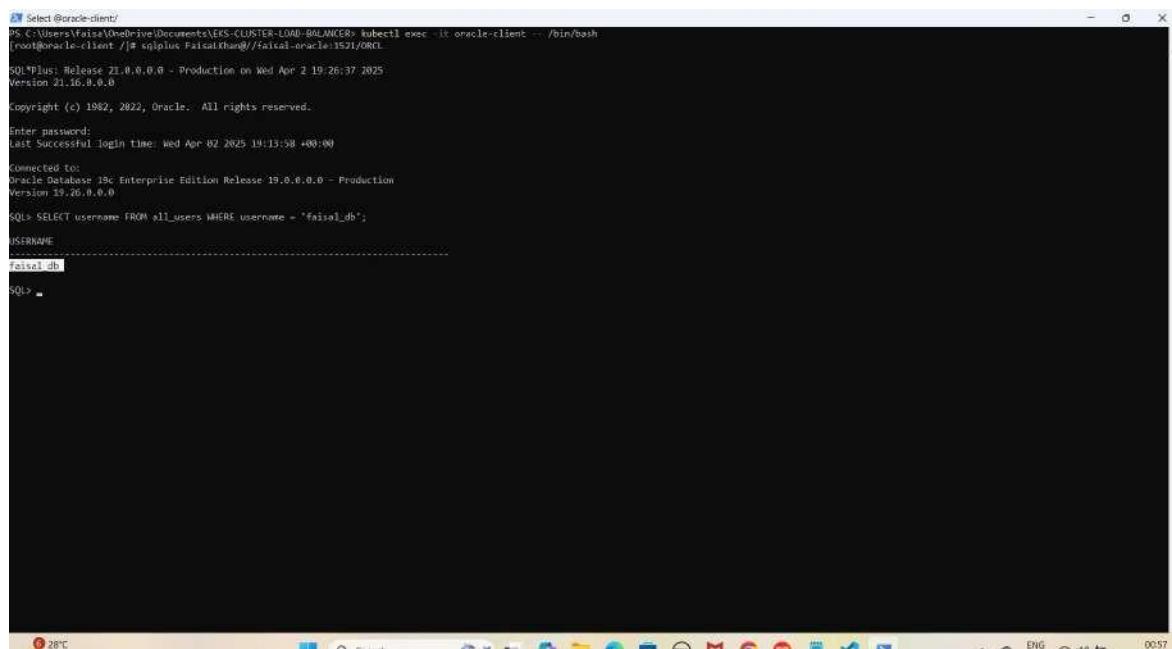
SQL>
```

**Note:** Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke rds-secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.

### **3. Databases check karne ke liye ye command run kariye**

```
SELECT username FROM all_users WHERE username = 'faisal_db';
```

**YE KUCH ISTARHA LAGEGA**



The screenshot shows a terminal window titled "Select @oracle-client". The command executed is:

```
PS C:\Users\faisal\OneDrive\Documents\KCS-CLUSTER-LOAD-BALANCER> kubectl exec -it oracle-client -- /bin/bash
root@oracle-client:/# sqlplus FaisalKhang@//faisal-oracle:1521/ORCL
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 2 19:26:37 2025
Version 21.0.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed Apr 02 2025 19:13:58 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0

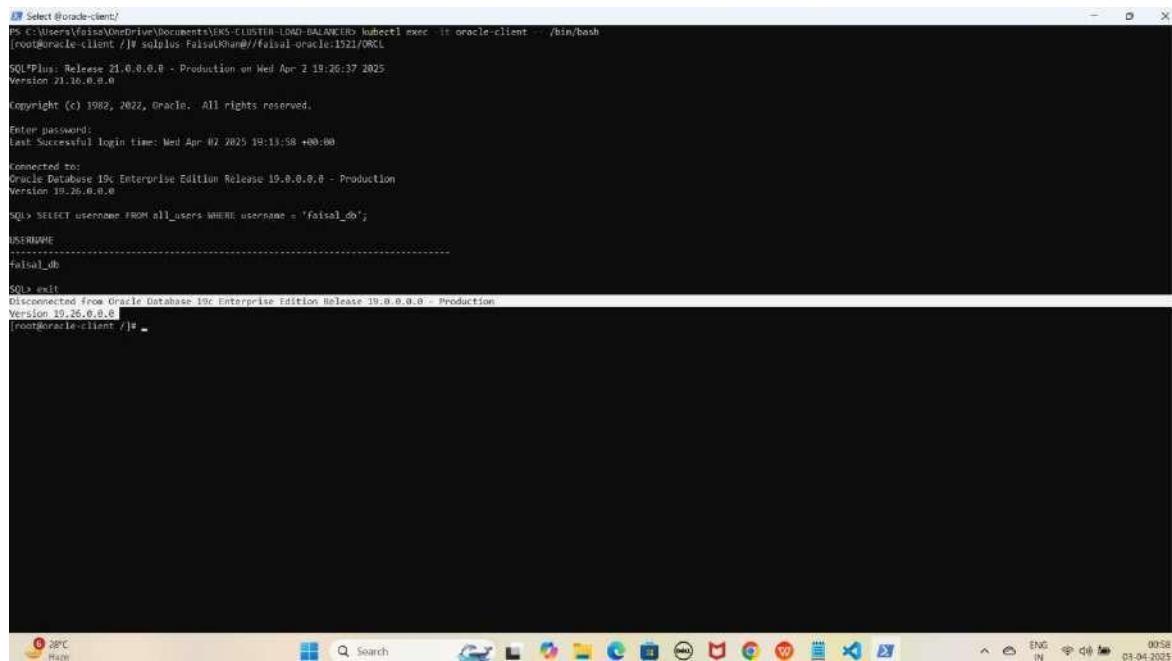
SQL> SELECT username FROM all_users WHERE username = 'faisal_db';
USERNAME
-----
faisal_db
```

**NOTE: Yahan hum dobara database create nahi karenge, kyunki pehle hi Method 1 mein create kar chuke hain. Hum sirf ye check karenge ki database available hai ya nahi.**

#### 4. Oracle database se exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisla\OneDrive\Documents\ERKS-CLUSTER-LOAD-BALANCER> kubectl exec -it oracle-client -- /bin/bash
(root@oracle-client:/# sqlplus FaisalKhan@//faisal-oracle:1521/ORCL
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 2 19:26:37 2025
Version 21.10.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed Apr 02 2025 19:13:58 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0

SQL> SELECT username FROM all_users WHERE username = 'faisal_db';

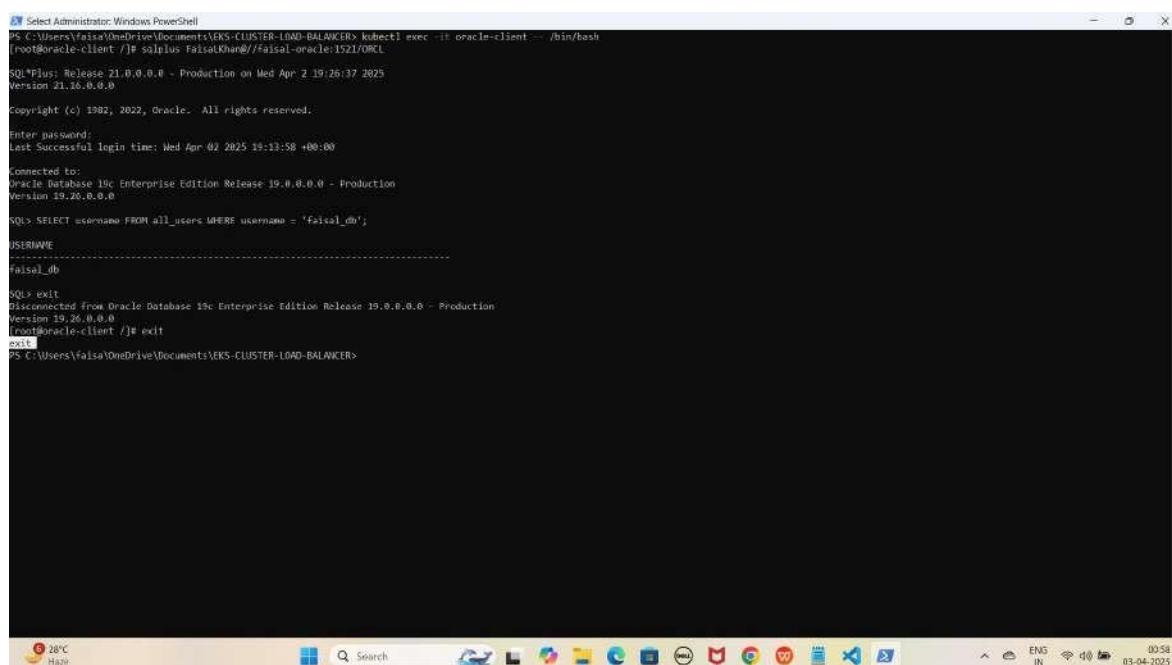
USERNAME
-----
faisal_db

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0
(root@oracle-client:/# )
```

#### 5. Oracle client pod se bhi exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisla\OneDrive\Documents\ERKS-CLUSTER-LOAD-BALANCER> kubectl exec -it oracle-client -- /bin/bash
(root@oracle-client:/# sqlplus FaisalKhan@//faisal-oracle:1521/ORCL
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 2 19:26:37 2025
Version 21.10.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed Apr 02 2025 19:13:58 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0

SQL> SELECT username FROM all_users WHERE username = 'faisal_db';

USERNAME
-----
faisal_db

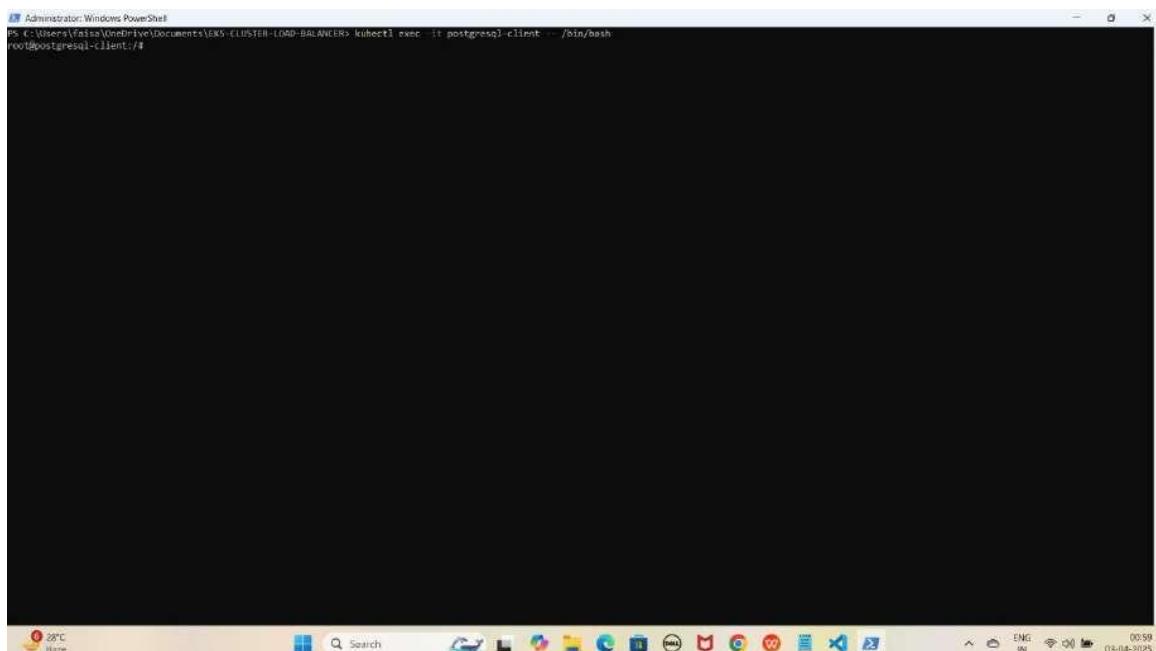
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.26.0.0.0
(root@oracle-client:/# exit
exit
PS C:\Users\faisla\OneDrive\Documents\ERKS-CLUSTER-LOAD-BALANCER>
```

## **Step 3: Amazon RDS (PostgreSQL) Database Access Karo - Method 2**

### **1. PostgreSQL client pod me jane ke liye ye command run kaiye**

```
kubectl exec -it postgresql-client -- /bin/bash
```

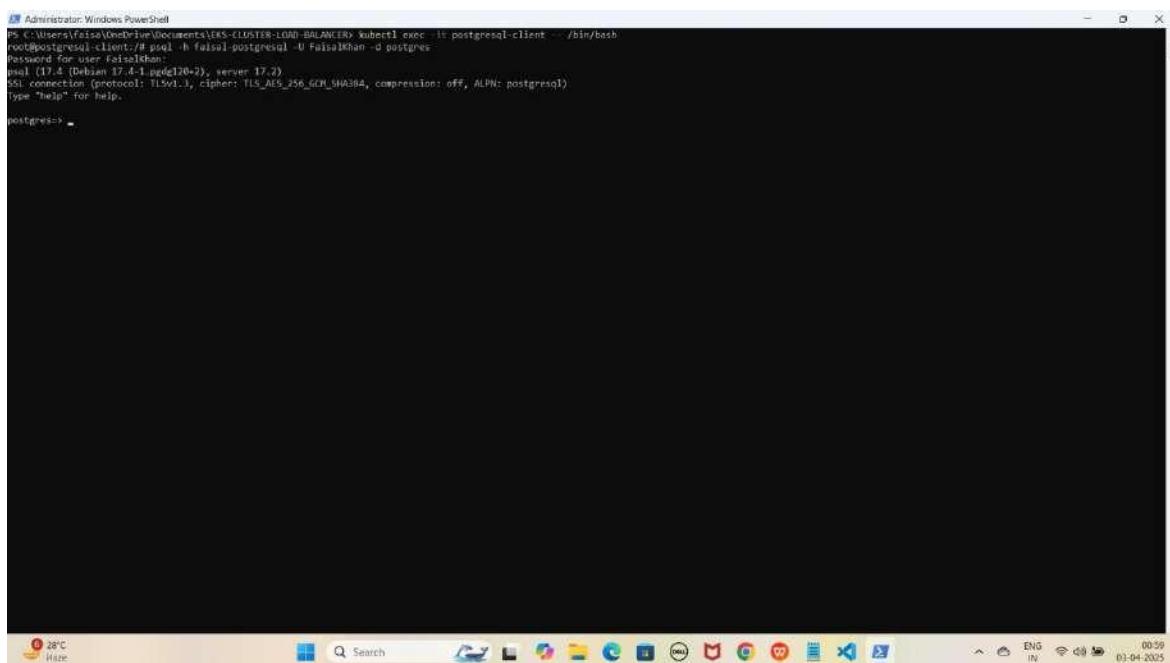
YE KUCH ISTARHA LAGEGA



**2. PostgreSQL database se connect karne ke liye ye command kuch istarha hogi**

```
psql -h faisal-postgresql -U FaisalKhan -d postgres
```

**YE KUCH ISTARHA LAGEGA**



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command entered is:

```
psql -h faisal-postgresql -U FaisalKhan -d postgres
```

The output shows the PostgreSQL prompt:

```
psql (17.4 (Debian 37.4-1.pgdg120+2), server 17.2)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgres)
Type "help" for help.
```

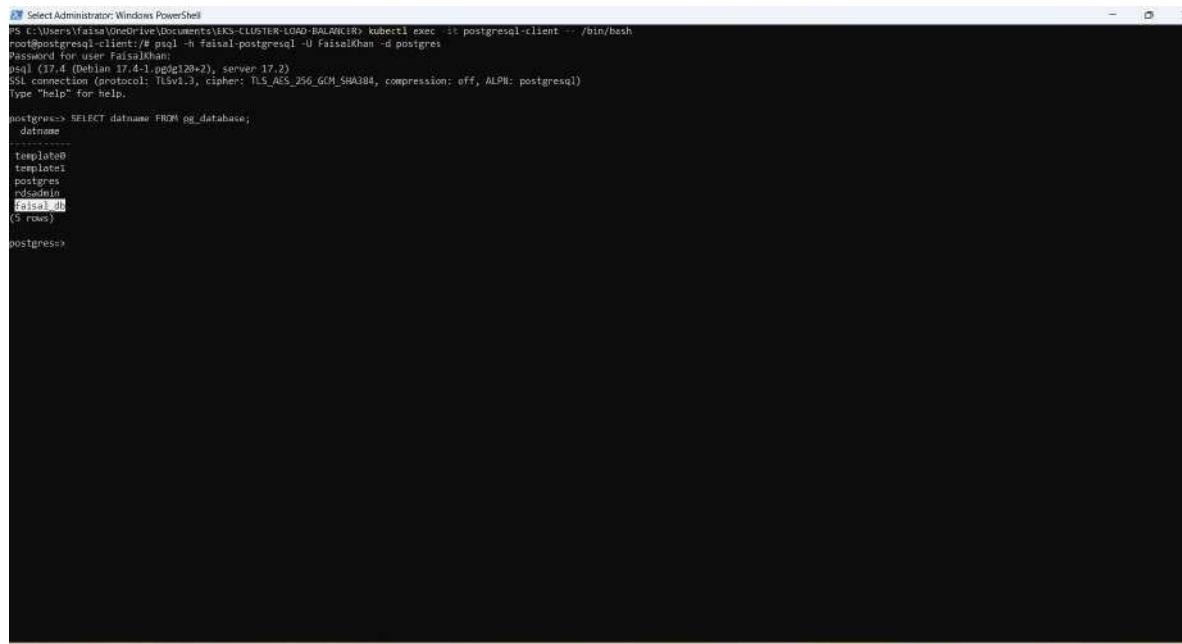
The window has a standard Windows title bar and taskbar at the bottom. The taskbar includes icons for search, file explorer, internet browser, and other system applications. The system tray shows the date (03-04-2025) and time (00:59).

**Note: Jab aap se password maanga jaaye, to wahi password use karein jo pehle Base64 encode karke rds-secrets.yaml file me store kiya gaya tha. Maine 'Faisalkhan35\$' diya tha, lekin password enter karte waqt yeh screen par show nahi hoga.**

### **3. Databases check karne ke liye ye command run kariye**

```
SELECT datname FROM pg_database;
```

**YE KUCH ISTARHA LAGEGA**



```
PS C:\Users\faisalm\OneDrive\Documents\IKS-CLUSTER-LOAD-BALANCER> kubectl exec -it postgresql-client -- /bin/bash
root@postgresql-client:/# psql -h faisal-postgresql -U FaisalKhan -d postgres
Password for user FaisalKhan:
psql (17.4 (Debian 17.4-1.postgres120+2), server 17.2)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA256, compression: off, ALPM: postgresql)
Type "help" for help.

postgres=> SELECT datname FROM pg_database;
      datname      
-----
template0
template1
postgres
rdsadmin
faisal_db
(5 rows)

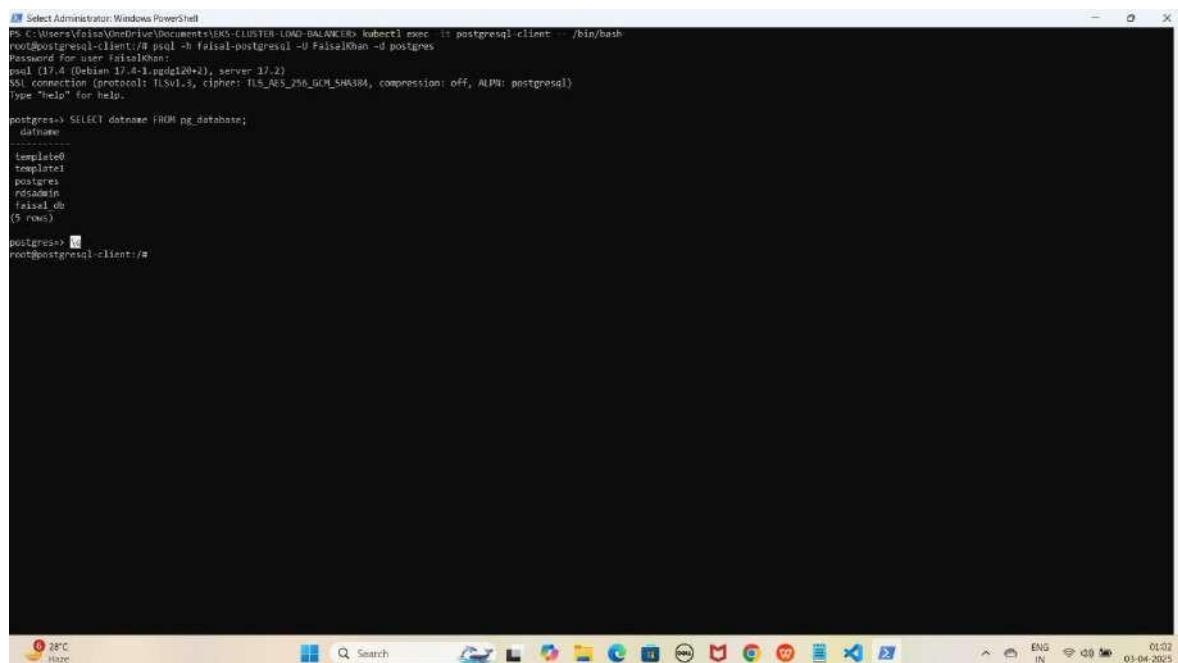
postgres>
```

**NOTE: Yahan hum dobara database create nahi karenge, kyunki pehle hi  
Method 1 mein create kar chuke hain. Hum sirf ye check karenge ki  
database available hai ya nahi.**

#### 4. PostgreSQL database se exit hoagaye exit hone ke liye exit type kariye

\q

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisal\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER> kubectl exec -it postgresql-client -- /bin/bash
root@postgresql-client:/# psql -h faisal-postgresql -U FaisalKhan -d postgres
Password for user FaisalKhan:
psql (17.4 (Debian 17.4-1+pgdg120+2), server 17.2)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgres)
Type "help" for help.

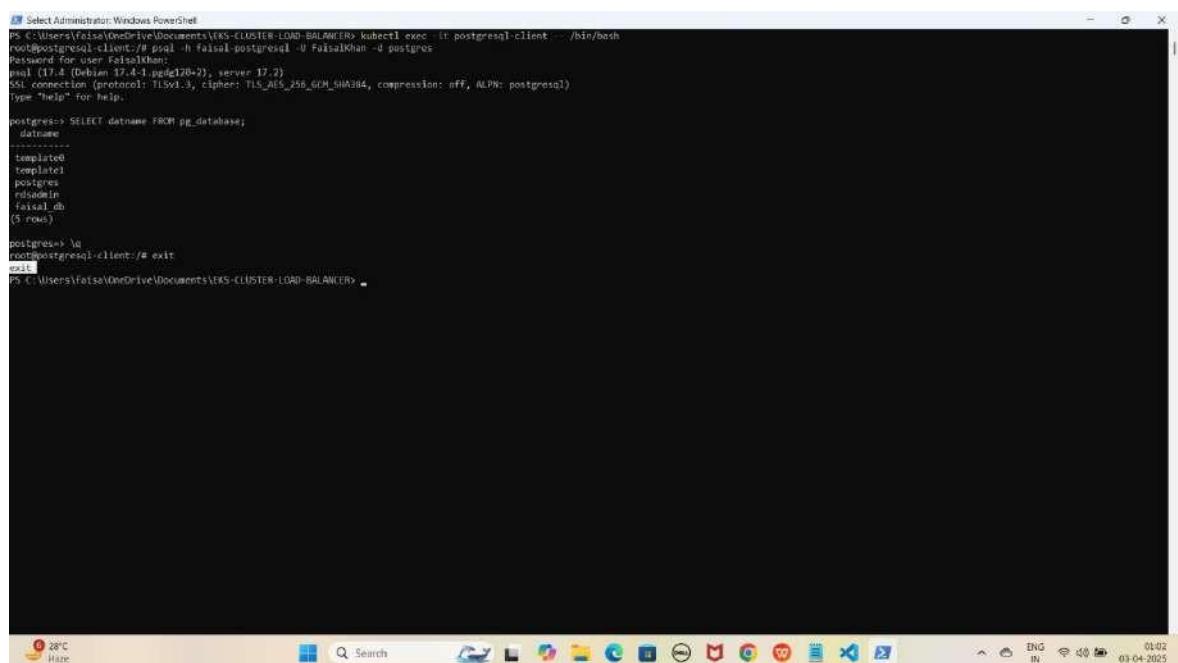
postgres> SELECT datname FROM pg_database;
datname
-----
template0
template1
postgres
faisal
faisal_db
(5 rows)

postgres> \q
root@postgresql-client:/#
```

#### 5. PostgreSQL client pod se bhi exit hoagaye exit hone ke liye exit type kariye

exit

YE KUCH ISTARHA LAGEGA



```
PS C:\Users\faisal\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER> kubectl exec -it postgresql-client -- /bin/bash
root@postgresql-client:/# psql -h faisal-postgresql -U FaisalKhan -d postgres
Password for user FaisalKhan:
psql (17.4 (Debian 17.4-1+pgdg120+2), server 17.2)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgres)
Type "help" for help.

postgres> SELECT datname FROM pg_database;
datname
-----
template0
template1
postgres
faisal
faisal_db
(5 rows)

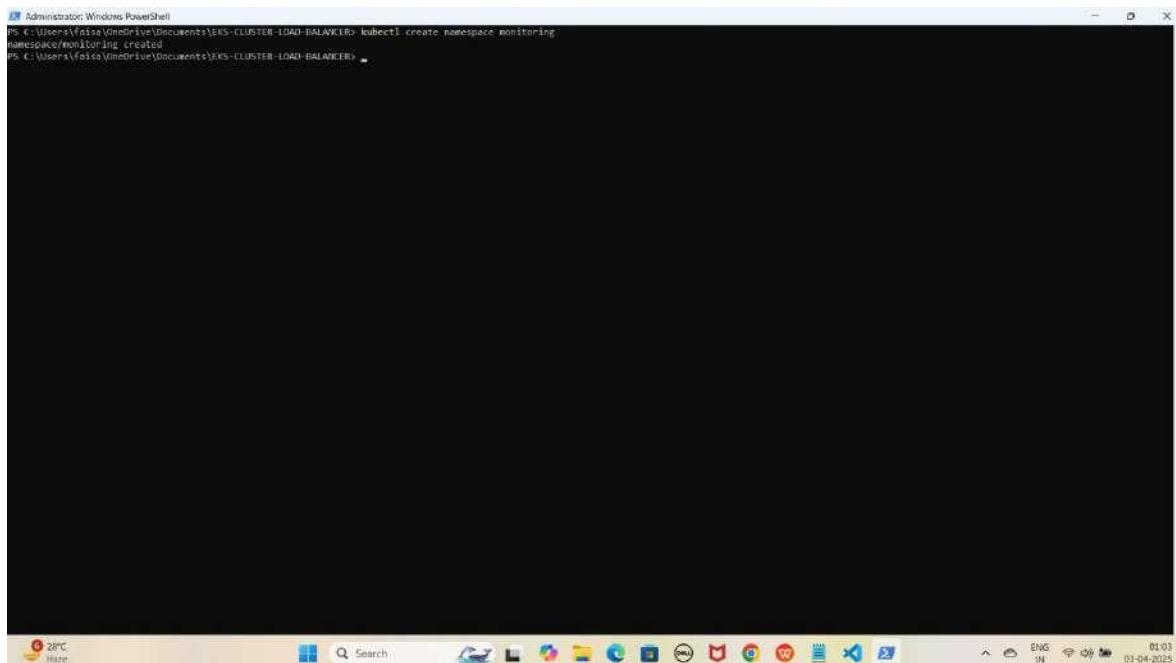
postgres> \q
root@postgresql-client:/# exit
exit
PS C:\Users\faisal\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER>
```

## Part 10: Monitoring Using Prometheus and Loki With Grafana

Kubernetes me ek naya namespace 'monitoring' create karna hoga.  
Namespace create karne ke liye ye command run kariye:

```
kubectl create namespace monitoring
```

YE KUCH ISTARHA LAGEGA



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command "kubectl create namespace monitoring" is entered and executed, resulting in the output "namespace/monitoring created". The window is set against a black background with white text. The taskbar at the bottom shows various icons for system status and application launch.

```
Administrator: Windows PowerShell
PS C:\Users\fatia\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER> kubectl create namespace monitoring
namespace/monitoring created
PS C:\Users\fatia\OneDrive\Documents\K8S-CLUSTER-LOAD-BALANCER>
```

## Step 1: prometheus-daemonset.yaml File Ka Kaam

Yeh file Prometheus monitoring tool deploy karne ke liye use hoti hai. Isko cluster ka CPU usage, memory, aur network monitor karne ke liye **Kubernetes DaemonSet** ke andar likha jata hai.

JAISE KI:-

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: prometheus
  namespace: monitoring
spec:
  selector:
    matchLabels:
      app: prometheus
  template:
    metadata:
      labels:
        app: prometheus
    spec:
      containers:
        - name: prometheus
          image: prom/prometheus:v2.31.1
          args:
            - "--config.file=/etc/prometheus/prometheus.yml"
            - "--storage.tsdb.path=/prometheus"
            - "--web.enable-lifecycle"
          volumeMounts:
            - name: prometheus-config
              mountPath: /etc/prometheus
            - name: prometheus-storage
              mountPath: /prometheus
      volumes:
        - name: prometheus-config
          configMap:
            name: prometheus-config
        - name: prometheus-storage
          emptyDir: {}

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
  namespace: monitoring
data:
  prometheus.yml: |
    global:
      scrape_interval: 15s # Default scrape interval for all jobs

    scrape_configs:
      # Kubernetes Nodes Service Discovery for Node Exporters
      - job_name: 'kubernetes-nodes'
        kubernetes_sd_configs:
          - role: node # Specify the role as 'node'
        relabel_configs:
          - source_labels: [__meta_kubernetes_node_name] # Scrape metrics from nodes
            target_label: kubernetes_node
          - source_labels: [__meta_kubernetes_node_label_kubernetes_io_hostname] # Include the
node hostname as a label
```

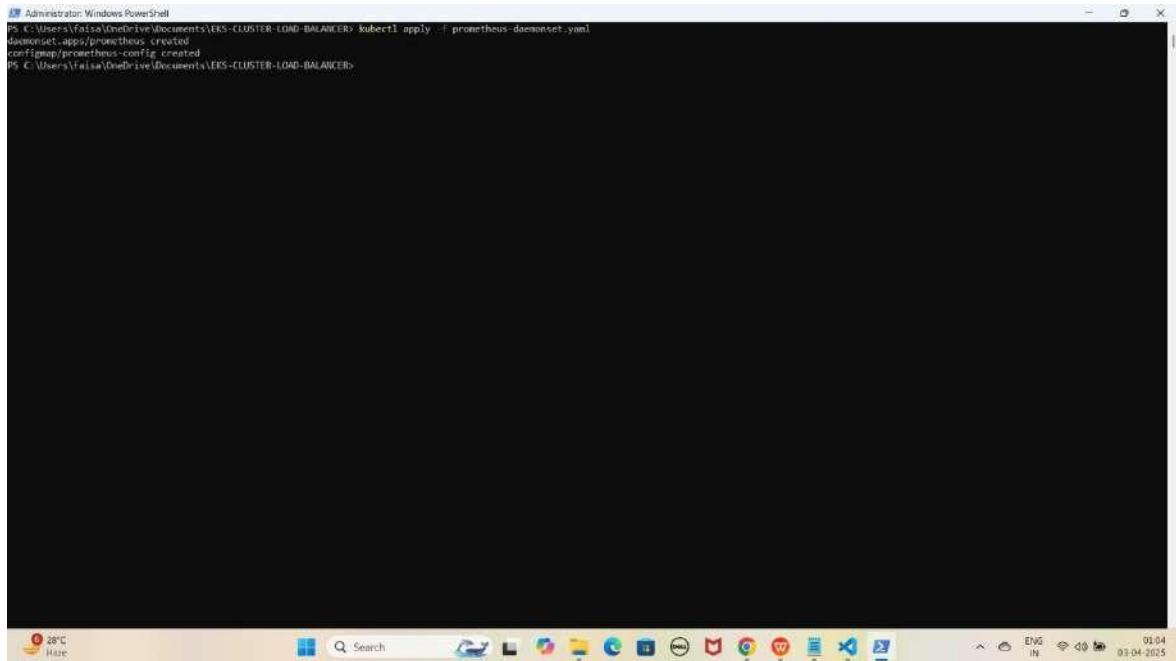
```
target_label: kubernetes_hostname

# Kubernetes Pods Service Discovery (for scraping Kubernetes pods)
- job_name: 'kubernetes-pods'
  kubernetes_sd_configs:
    - role: pod # Scrape all pods (can be further refined with relabeling)
  relabel_configs:
    - source_labels: [__meta_kubernetes_pod_label_app] # Use pod labels for further
filtering
      target_label: app
```

## Prometheus DaemonSet Apply Karo

kubectl apply -f prometheus-daemonset.yaml

YE KUCH ISTARHA LAGEGA



## Step 2: prometheus-rbac.yaml File Ka Kaam

Yeh **file Prometheus** ko required **permissions** dene ke liye use hoti hai, taake woh **cluster** ke **resources** ko access kar sake.

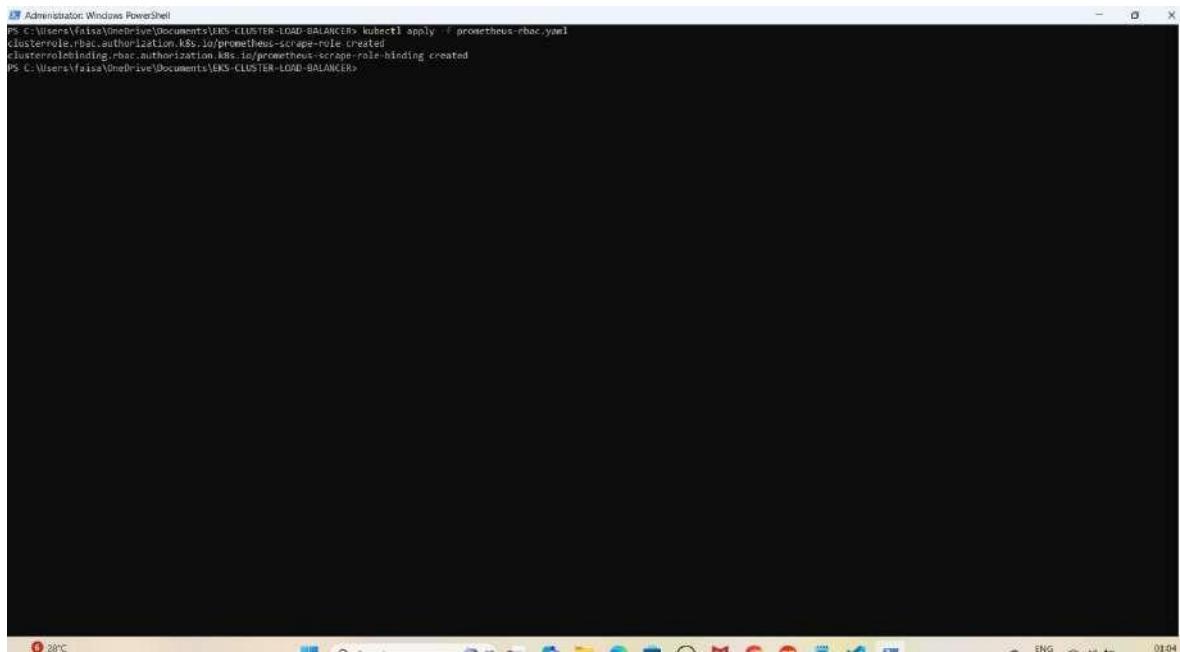
JAISE KI:-

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: prometheus-scrape-role
rules:
  - apiGroups: [""]
    resources: ["pods", "nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["metrics.k8s.io"]
    resources: ["nodes", "pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["deployments", "replicasets", "daemonsets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["extensions"]
    resources: ["ingresses"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["services", "endpoints"]
    verbs: ["get", "list", "watch"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: prometheus-scrape-role-binding
subjects:
  - kind: ServiceAccount
    name: default
    namespace: monitoring
roleRef:
  kind: ClusterRole
  name: prometheus-scrape-role
  apiGroup: rbac.authorization.k8s.io
```

## Prometheus RBAC Apply Karo

```
kubectl apply -f prometheus-rbac.yaml
```

YE KUCH ISTARHA LAGEGA



```
[Administrator: Windows PowerShell]
PS C:\Users\faiza\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f prometheus-rbac.yaml
clusterrole.rbac.authorization.k8s.io/prometheus-scrape-role created
clusterrolebinding.rbac.authorization.k8s.io/prometheus-scrape-role-binding created
PS C:\Users\faiza\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

### Step 3: prometheus-nodeport-service.yaml File Ka Kaam

Yeh file Prometheus ko port **30090** par **NodePort service** ke zariye **expose** karne ke liye use hoti hai, taake hum usko **browser** me **access** kar sakein.

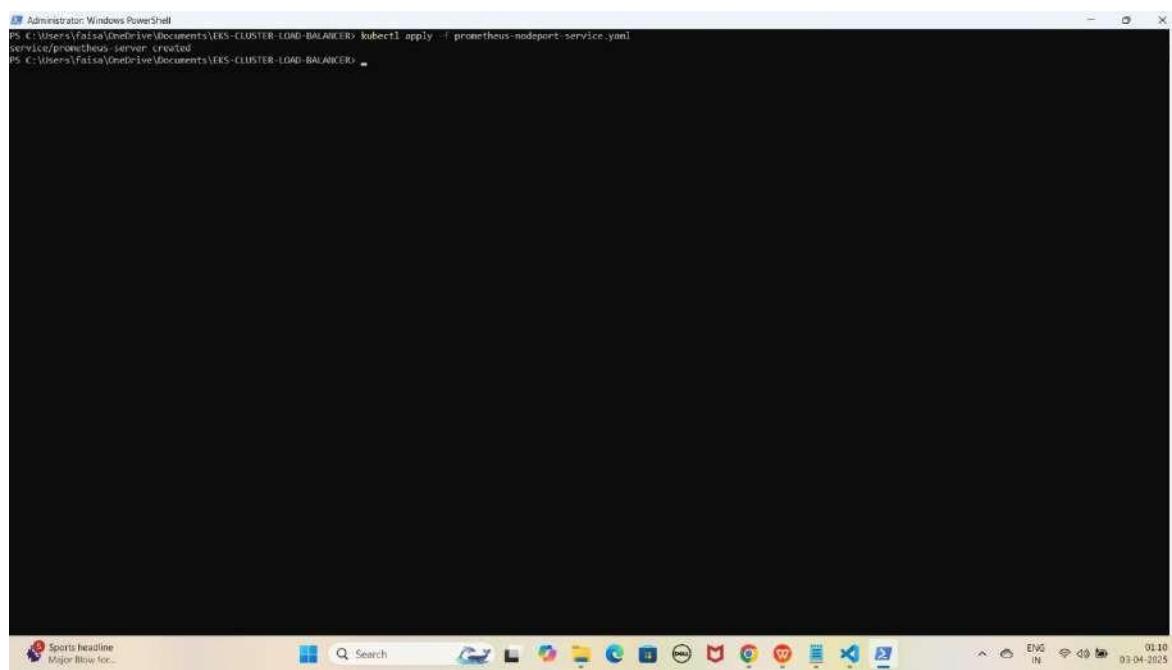
JAISE KI:-

```
apiVersion: v1
kind: Service
metadata:
  name: prometheus-server
  namespace: monitoring
spec:
  ports:
    - port: 80
      targetPort: 9090
      nodePort: 30090
  selector:
    app: prometheus
  type: NodePort
```

### Prometheus Service Apply Karo

kubectl apply -f prometheus-nodeport-service.yaml

### YE KUCH ISTARHA LAGEGA

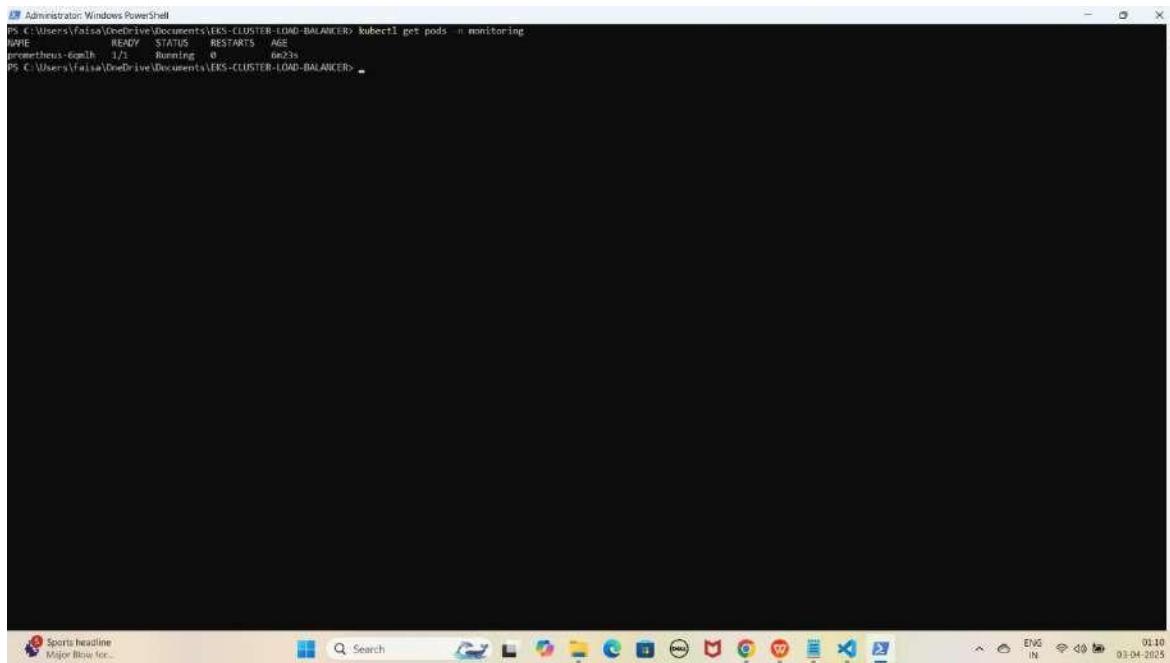


```
PS C:\Users\fatima\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f prometheus-nodeport-service.yaml
service/prometheus-server created
PS C:\Users\fatima\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## **1. Pods check karne ke liye ye command run kariye**

kubectl get pods -n monitoring

YE KUCH ISTARHA LAGEGA



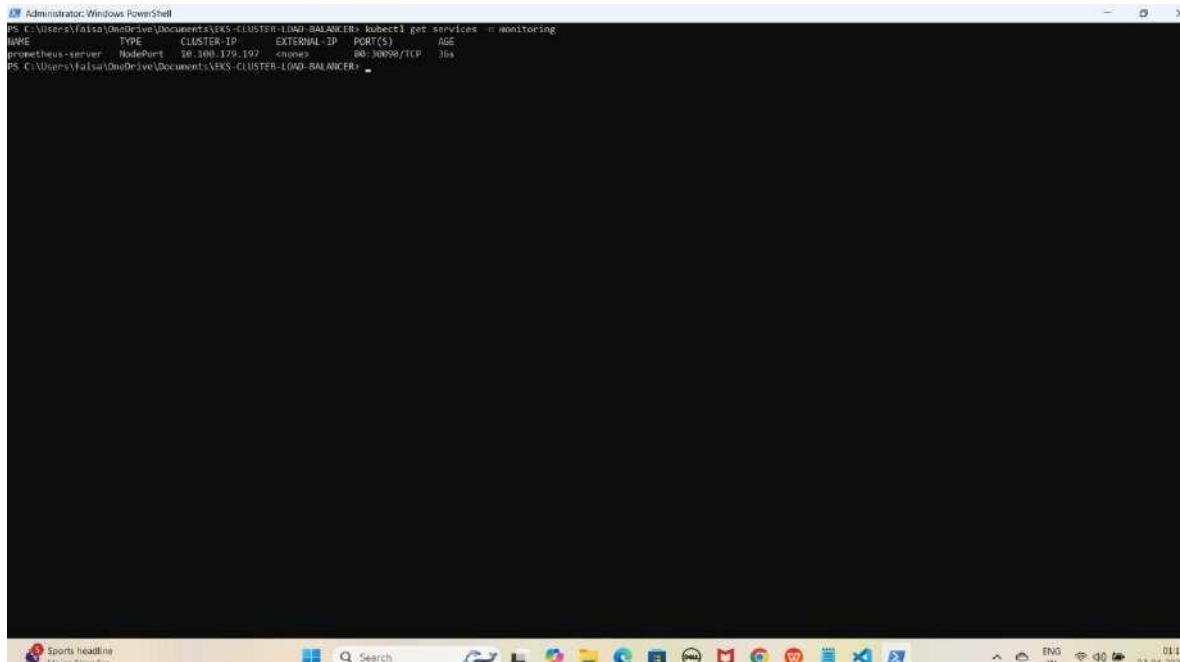
```
Administrator: Windows PowerShell
PS C:\Users\Yash\OneDrive\Documents\LENS-CLUSTER-LOAD-BALANCER> kubectl get pods -n monitoring
NAME      READY   STATUS    RESTARTS   AGE
prometheus-6qnh   1/1     Running   0          6m23s
PS C:\Users\Yash\OneDrive\Documents\LENS-CLUSTER-LOAD-BALANCER>
```

**NOTE: Agar STATUS Running show karraha hai to sab kuch sahi hai**

## 2. Services check karne ke liye ye commad run kariye

kubectl get services -n monitoring

YE KUCH ISTARHA LAGEGA



```
Administrator: Windows PowerShell
PS C:\Users\faiza\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl get services -n monitoring
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
prometheus-server   NodePort    10.100.179.192  <none>        80:30090/TCP   36s
PS C:\Users\faiza\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

**NOTE:** Ab aapka Prometheus 30090 port par expose ho chuka hai. Browser me EKS Node ki Public IP ke saath check kariye jaise ki mere case me kuch aisa hogा

Prometheus: <http://EKS-Node-Public IP:30090>

## YE KUCH ISTARHA LAGEGA

The screenshot shows the AWS EC2 Instances page. A single instance, 'Faisal-Node' (i-0aac44ac8fcc5c7cf), is listed as 'Running'. The instance details page is open, showing the following information:

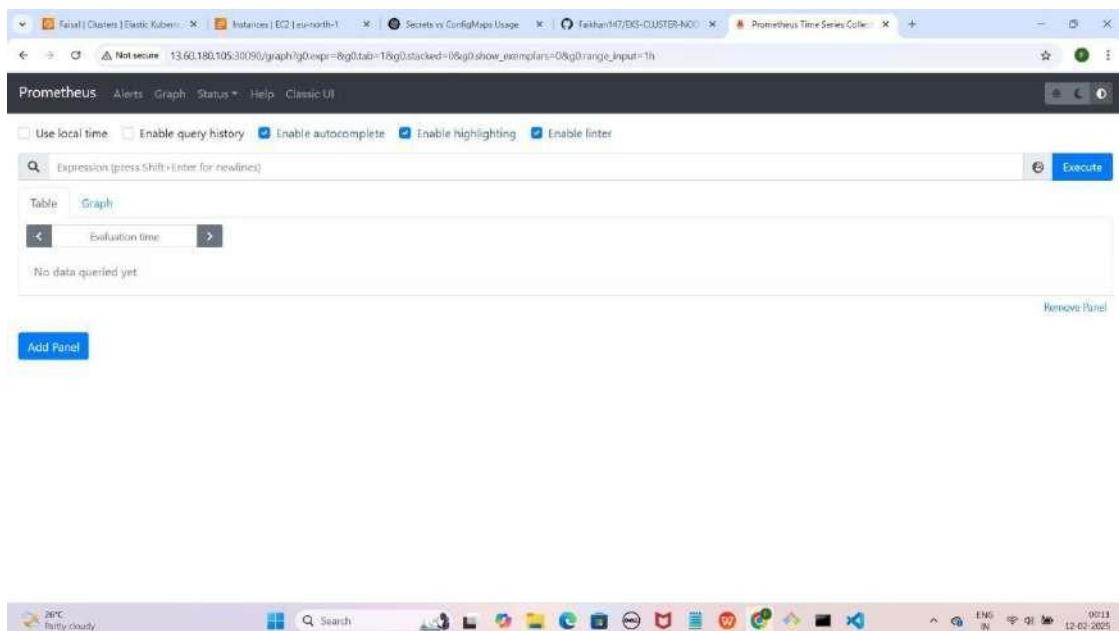
Attribute	Value
Instance ID	i-0aac44ac8fcc5c7cf
Public IPv4 address	13.60.105.106
Private IP4 addresses	172.31.37.205 172.31.35.31
Public IPv4 DNS	ec2-13-60-105-105.eu-north-1.compute.amazonaws.com
Instance type	t3.large
Instance state	Running
Private IP4 DNS name	ip-172-31-37-215.eu-north-1.compute.internal

The screenshot shows a browser window with several pinned icons on the toolbar:

- AWS ACCOUNT
- Telegram Web
- Create Account - Tr...
- ChatGPT

The main content area is mostly blank.





### 3. CPU usage check karne ke liye Prometheus me ye command run karein

```
rate(process_cpu_seconds_total[30s])
```

**Note: Command Prometheus me paste karne ke baad Execute pe click karein.**

## YE KUCH ISTARHA LAGEGA

Instance	CPU Usage
172.31.37.215.67698:job=kubernetes-pods	0.006666666666666524
172.31.37.8.8153:job=kubernetes-pods	0.001333333333333049
172.31.42.159.9153:job=kubernetes-pods	0.001333333333333049

## 4. Memory usage check karne ke liye Prometheus me ye command run karein

process\_resident\_memory\_bytes

**Note:** Command Prometheus me paste karne ke baad Execute pe click karein.

## YE KUCH ISTARHA LAGEGA

Process	Memory Usage
process_resident_memory_bytes:instance=172.31.37.215.65678:job=kubernetes-pods	68644864
process_resident_memory_bytes:instance=172.31.37.8.8153:job=kubernetes-pods	57167872
process_resident_memory_bytes:instance=172.31.42.159.9153:job=kubernetes-pods	57571472

## 5. Network Transmit aur Receive Bytes me dekhne ke liye Prometheus me ye commands run karein

- **Transmit Bytes:** process\_network\_transmit\_bytes\_total
- **Receive Bytes:** process\_network\_receive\_bytes\_total

**Note: Command Prometheus me paste karne ke baad Execute pe click karein.**

YE KUCH ISTARHA LAGEGA

The screenshot shows the Prometheus web interface with the URL `http://13.60.180.105:30090/graph?g0.expr=process_network_transmit_bytes_total&g0.tab=1&g0.stacked=0&g0.show_in_exemplars=0&g0.range_input=1h`. The results table displays a single data point for a Kubernetes pod instance: `process_network_transmit_bytes_total{instance="172.31.37.215:1678", job="kubernetes-pods"} 47341376`.

The screenshot shows the Prometheus web interface with the URL `http://13.60.180.105:30090/graph?g0.expr=process_network_receive_bytes_total&g0.tab=1&g0.stacked=0&g0.show_in_exemplars=0&g0.range_input=1h`. The results table displays a single data point for a Kubernetes pod instance: `process_network_receive_bytes_total{instance="172.31.37.215:1678", job="kubernetes-pods"} 1475239029`.



## Step 4: promtail-daemonset.yaml File Ka Kaam

Yeh **file cluster** se **logs collect** karne aur unhe **Loki** par bhejne ke liye use hoti hai.

JAISE KI:-

```
---
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: promtail
  namespace: monitoring
spec:
  selector:
    matchLabels:
      app: promtail
  template:
    metadata:
      labels:
        app: promtail
    spec:
      containers:
        - name: promtail
          image: grafana/promtail:2.3.0
          args:
            - "-config.file=/etc/promtail/promtail-config.yaml"
          volumeMounts:
            - name: promtail-config
              mountPath: /etc/promtail
            - name: varlogs
              mountPath: /var/log
            - name: positions
              mountPath: /run/promtail
          resources:
            requests:
              memory: "128Mi"
              cpu: "100m"
            limits:
              memory: "256Mi"
              cpu: "500m"
          readinessProbe:
            httpGet:
              path: /ready
              port: 9080
            initialDelaySeconds: 5
            periodSeconds: 10
          livenessProbe:
            httpGet:
              path: /ready
              port: 9080
            initialDelaySeconds: 10
            periodSeconds: 20
          securityContext:
            runAsUser: 0 # Run as root to allow access to restricted files
            runAsGroup: 0 # Run as root group
            privileged: true # Optional: Uncomment if needed to run in privileged mode
      volumes:
        - name: promtail-config
          configMap:
            name: promtail-config
        - name: varlogs
          hostPath:
            path: /var/log
```

```
        type: DirectoryOrCreate
        - name: positions
          emptyDir: {}

---
apiVersion: v1
kind: ConfigMap
metadata:
  name: promtail-config
  namespace: monitoring
data:
  promtail-config.yaml: |
    server:
      http_listen_port: 9080
      grpc_listen_port: 0

    positions:
      filename: /run/promtail/positions.yaml

clients:
  - url: http://loki.monitoring.svc.cluster.local:3100/loki/api/v1/push

scrape_configs:
  - job_name: system
    static_configs:
      - targets:
          - localhost
        labels:
          job: varlogs
          __path__: /var/log/*.log

  - job_name: kubernetes-pods
    static_configs:
      - targets:
          - localhost
        labels:
          job: podlogs
          __path__: /var/log/pods/**/* .log

# Additional configurations can be added here if needed
```

## Promtail DaemonSet Apply Karo

```
kubectl apply -f promtail-daemonset.yaml
```

YE KUCH ISTARHA LAGEGA



```
[Administrator: Windows PowerShell]
C:\Users\faizan\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f promtail-daemonset.yaml
daemonset "promtail" created
configmap "promtail-config" created
PS C:\Users\faizan\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## Step 5: loki-daemonset.yaml File Ka Kaam

Yeh file **Loki** ko ek **logs database** ke taur par **setup** karne ke liye use hoti hai, jo **Promtail** se **logs collect** karke **store** karta hai.

JAISE KI:-

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: loki
  namespace: monitoring
spec:
  selector:
    matchLabels:
      app: loki
  template:
    metadata:
      labels:
        app: loki
    spec:
      serviceAccountName: loki
      containers:
        - name: loki
          image: grafana/loki:2.3.0
          args:
            - "-config.file=/etc/loki/loki-config.yaml"
          ports:
            - containerPort: 3100
              name: http
          resources:
```

```
    requests:
      cpu: "500m"
      memory: "512Mi"
    limits:
      cpu: "1"
      memory: "1Gi"
  volumeMounts:
    - name: loki-config
      mountPath: /etc/loki/loki-config.yaml
      subPath: loki-config.yaml
  volumes:
    - name: loki-config
      configMap:
        name: loki-config
        items:
          - key: loki-config.yaml
            path: loki-config.yaml

```

---

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: loki-config
  namespace: monitoring
data:
  loki-config.yaml: |
    auth_enabled: false # Disable authentication explicitly
```

```
server:
  http_listen_port: 3100
```

```
distributor:
  ring:
    kvstore:
      store: inmemory
```

```
ingester:
  lifecycler:
    ring:
      kvstore:
        store: inmemory
        replication_factor: 1
    chunk_idle_period: 5m
    chunk_retain_period: 30s
    max_transfer_retries: 0
```

```
schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h
```

```
storage_config:
  boltdb_shipper:
    active_index_directory: /loki/index
```

```
    cache_location: /loki/cache
    shared_store: filesystem
  filesystem:
    directory: /loki/chunks
```

```
compactor:
  working_directory: /loki/compactor
  shared_store: filesystem
  compaction_interval: 5m
```

```
limits_config:
  enforce_metric_name: false
  reject_old_samples: true
  reject_old_samples_max_age: 168h
```

```
chunk_store_config:
  max_look_back_period: 0s
```

```
table_manager:
  retention_deletes_enabled: true
  retention_period: 168h
```

```
---
```

```
apiVersion: v1
kind: Service
metadata:
  name: loki
  namespace: monitoring
spec:
  type: ClusterIP
  ports:
    - port: 3100
      targetPort: 3100
  selector:
    app: loki
```

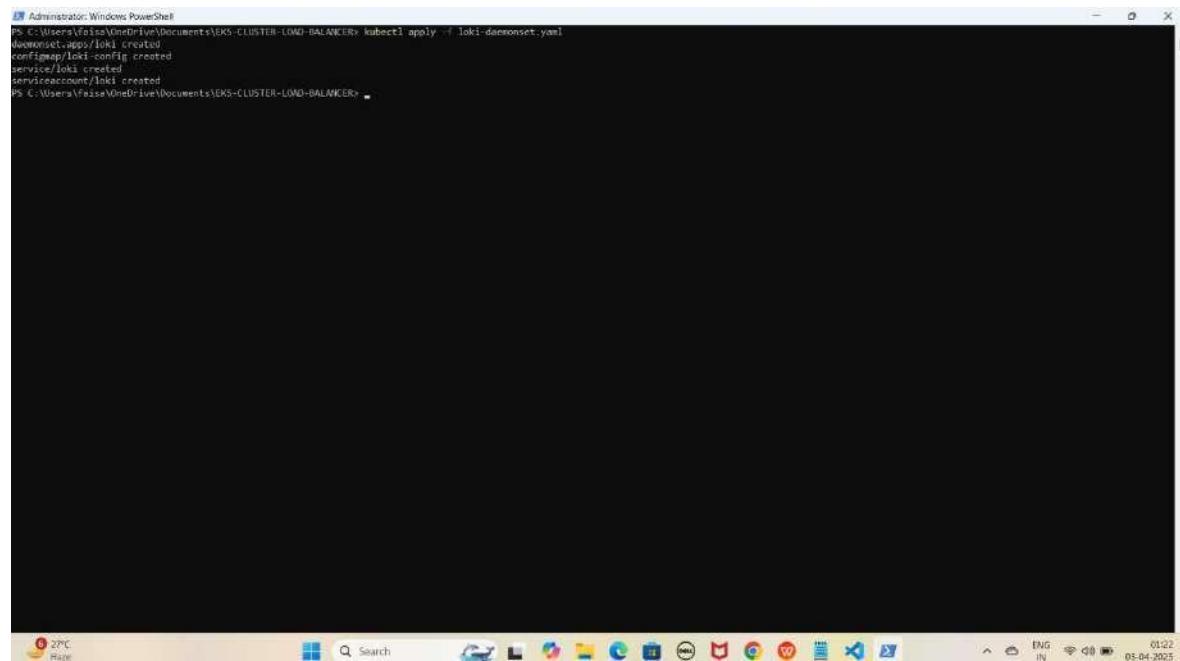
```
---
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: loki
  namespace: monitoring
```

## Loki DaemonSet Apply Karo

```
kubectl apply -f loki-daemonset.yaml
```

YE KUCH ISTARHA LAGEGA



```
[Administrator: Windows PowerShell]
PS C:\Users\faiza\OneDrive\Documents\EKS-CLUSTER-LAMBDA-BALANCER> kubectl apply -f loki-daemonset.yaml
daemonset.apps/loki created
configmap/loki-config created
service/loki created
serviceaccount/loki created
PS C:\Users\faiza\OneDrive\Documents\EKS-CLUSTER-LAMBDA-BALANCER>
```

## Step 6: loki-nodeport-service.yaml File Ka Kaam

Yeh **file Loki** ko **expose** karne ke liye use hoti hai, taake hum **port 30091** ke zariye usko **access** kar sakein. Isko **Kubernetes NodePort Service** ke andar likha jata hai.

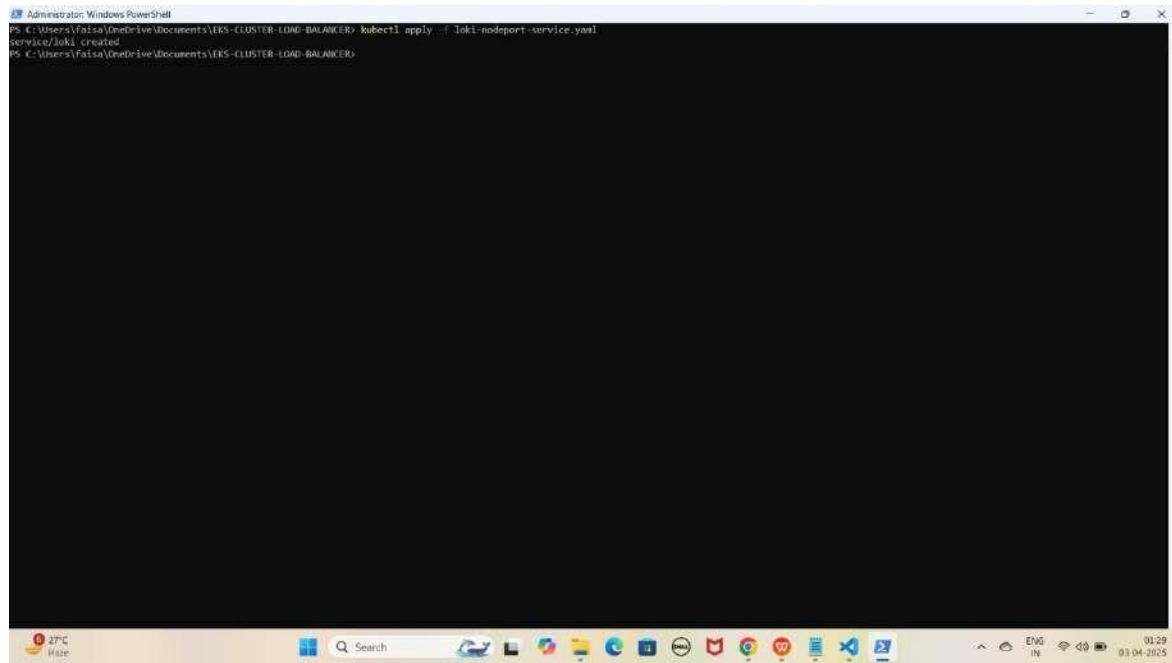
JAISE KI:-

```
apiVersion: v1
kind: Service
metadata:
  name: loki
  namespace: monitoring
spec:
  ports:
    - port: 3100
      targetPort: 3100
      nodePort: 30091 # This exposes Loki on port 30091
  selector:
    app: loki
  type: NodePort
```

## Loki Service Apply Karo

kubectl apply -f loki-nodeport-service.yaml

## YE KUCH ISTARHA LAGEGA

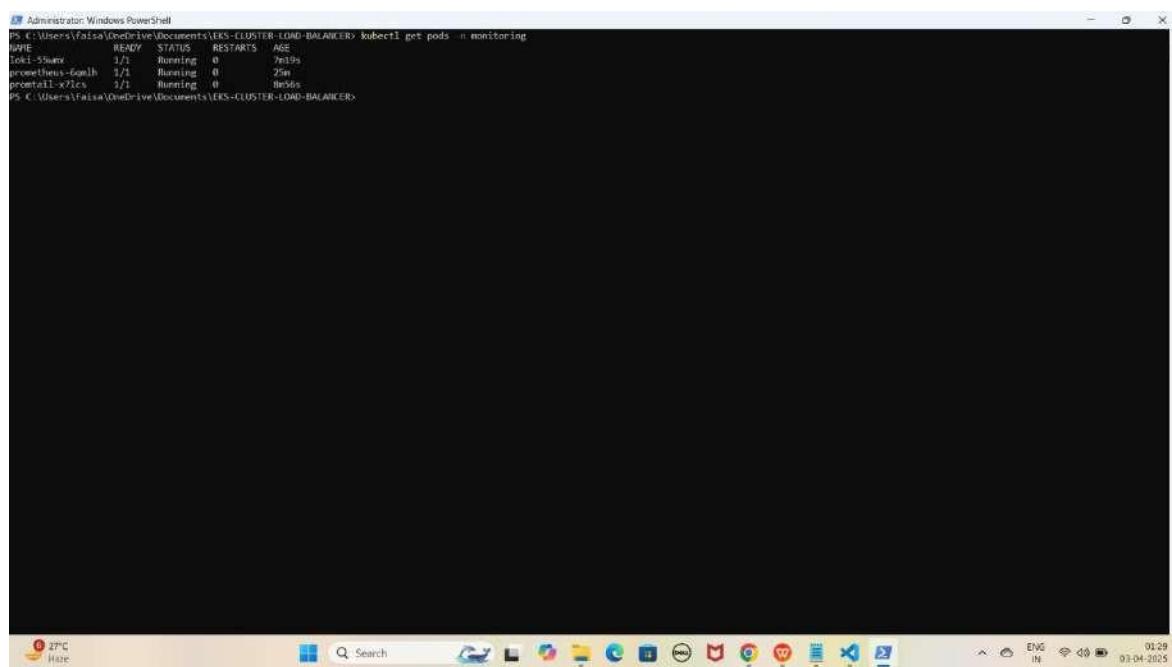


```
Administrator: Windows PowerShell
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f loki-nodeport-service.yaml
service/loki created
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## 1. Pods check karne ke liye ye command run kariye

kubectl get pods -n monitoring

## YE KUCH ISTARHA LAGEGA



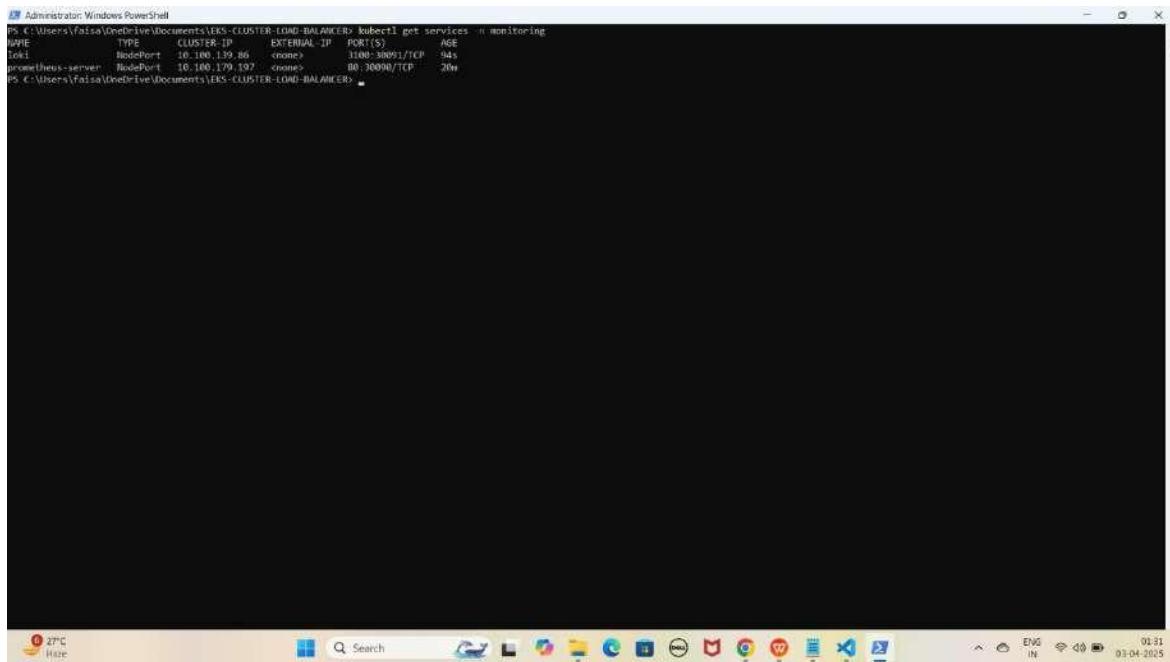
```
Administrator: Windows PowerShell
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl get pods -n monitoring
NAME          READY   STATUS    RESTARTS   AGE
loki-594cc   3/3     Running   0          7d19h
loki-594cc-h   1/1     Running   0          7d19h
promtail-x7ics   3/3     Running   0          8d23h
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

**NOTE: Agar STATE Running show karraha hai to sab kuch sahi hai**

## 2. Services check karne ke liye ye commad run kariye

kubectl get services -n monitoring

YE KUCH ISTARHA LAGEGA

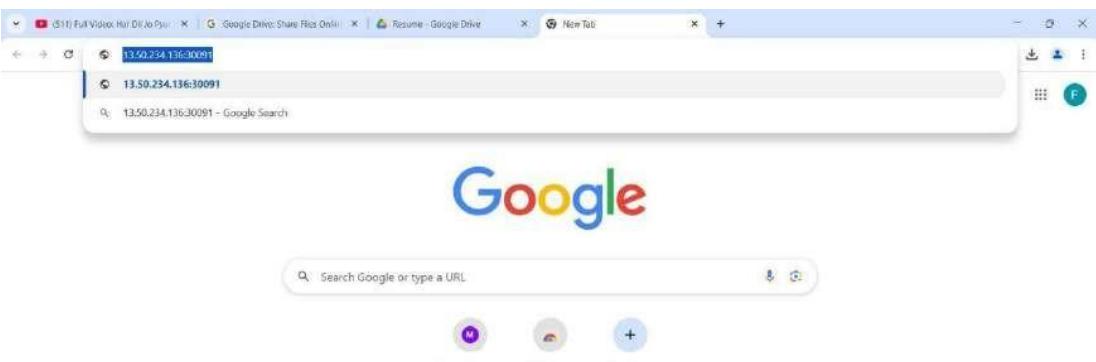
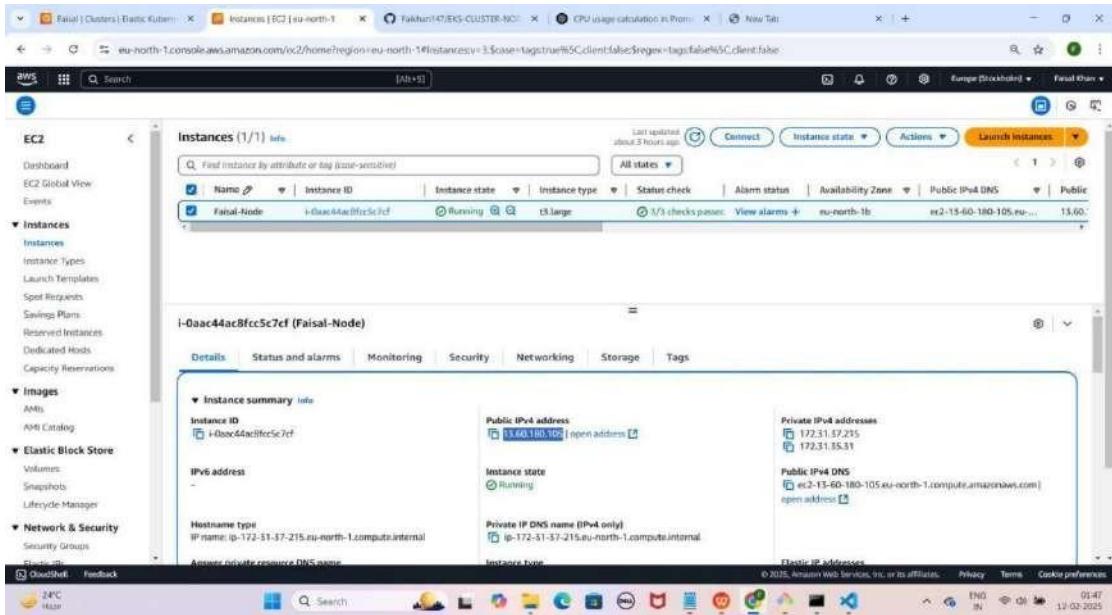


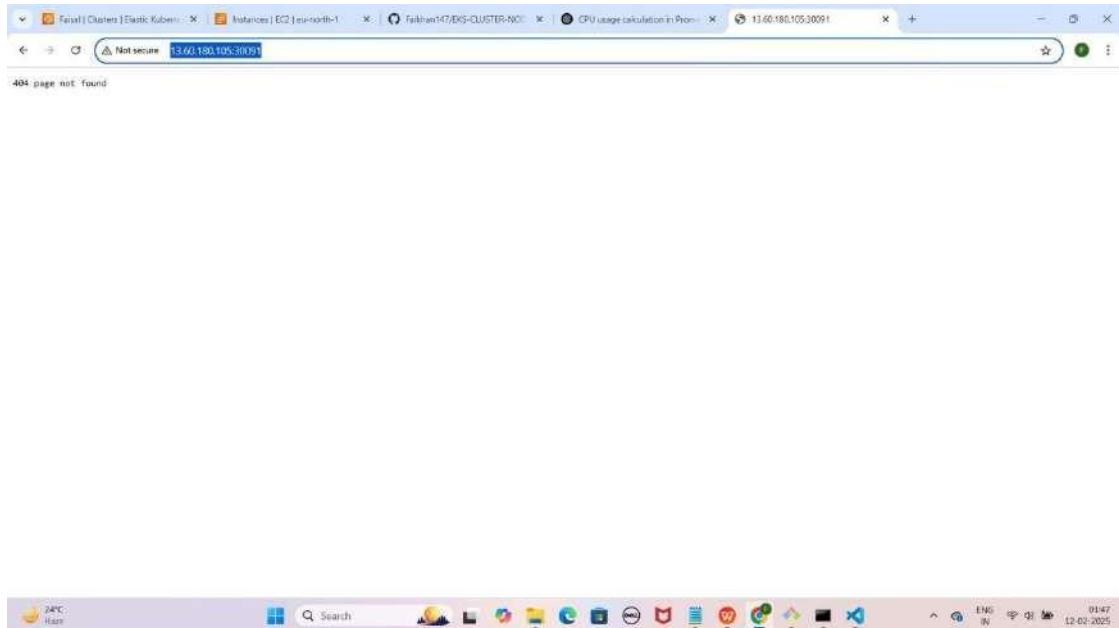
```
Administrator: Windows PowerShell
PS C:\Users\fasa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl get services -n monitoring
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
loki      NodePort    10.100.139.86  <none>        3100:30091/TCP   94s
prometheus-server  NodePort    10.100.179.197  <none>        80:30090/TCP    26s
PS C:\Users\fasa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

**NOTE:** Ab aapka Loki 30091 port par expose ho chuka hai. Browser me Cluster Node ki Public IP ke saath check karein jaise ki mere case me kuch aisa hogा.

Loki: <http://EKS-Node-Public IP:30091>

# YE KUCH ISTARHA LAGEGA

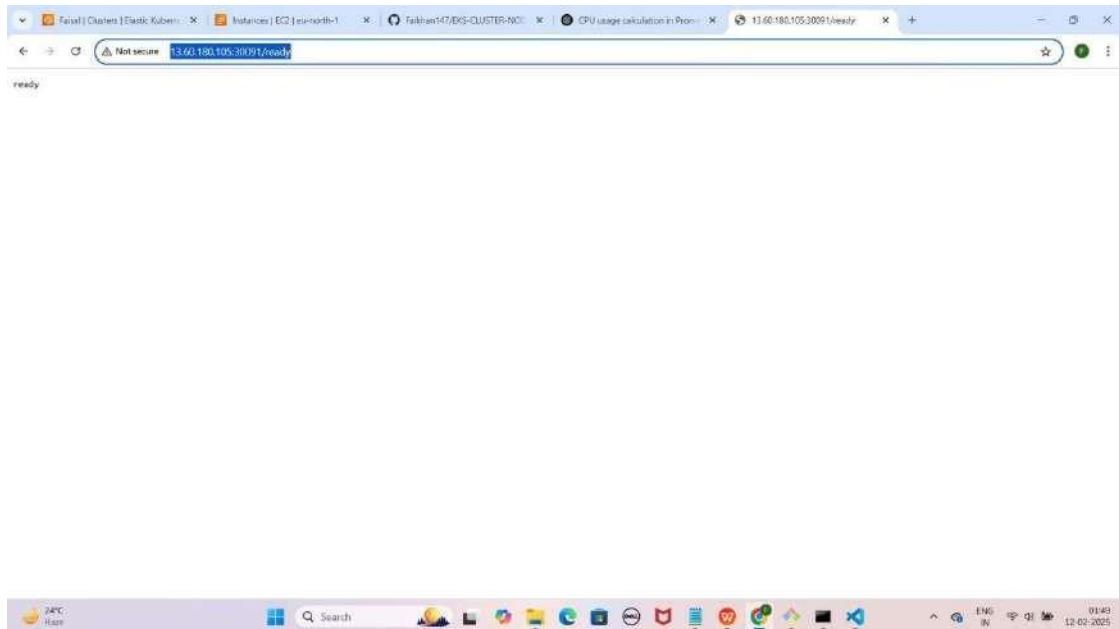




**3. Lekin aapko 404 page not found dikhayga to /ready se check kariye loki ko jaise ki mere case kuch asisa hoga.**

<http://EKS-Node-Public IP:30091/ready>

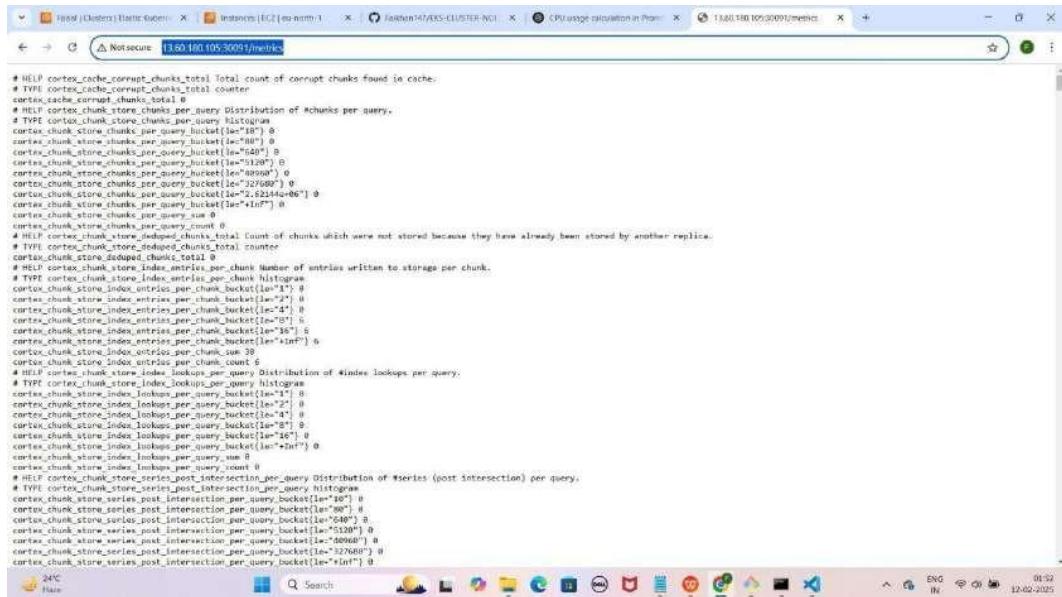
## YE KUCH ISTARHA LAGEGA



4. Agar aapko **ready show** karraha hai to iska matlab **loki successfully run** horaha hai ab **loki metrics loggs collect** karraha hai **check** karne ke liye **/metrics** run kariye jaise ki mere case me kuch aisa hogा.

<http://EKS-Node-Public IP:30091/metrics>

## YE KUCH ISTARHA LAGEGA



## Step 7: grafana-deployment.yaml File Ka Kaam

Yeh file Grafana ko **deploy** karne ke liye use hoti hai, taake hum **Prometheus metrics** aur **Loki logs** ko **visualize** kar sakein. Isme **NodePort Service** bhi use ki gayi hai, jisse **Grafana port 30080** par expose hoga.

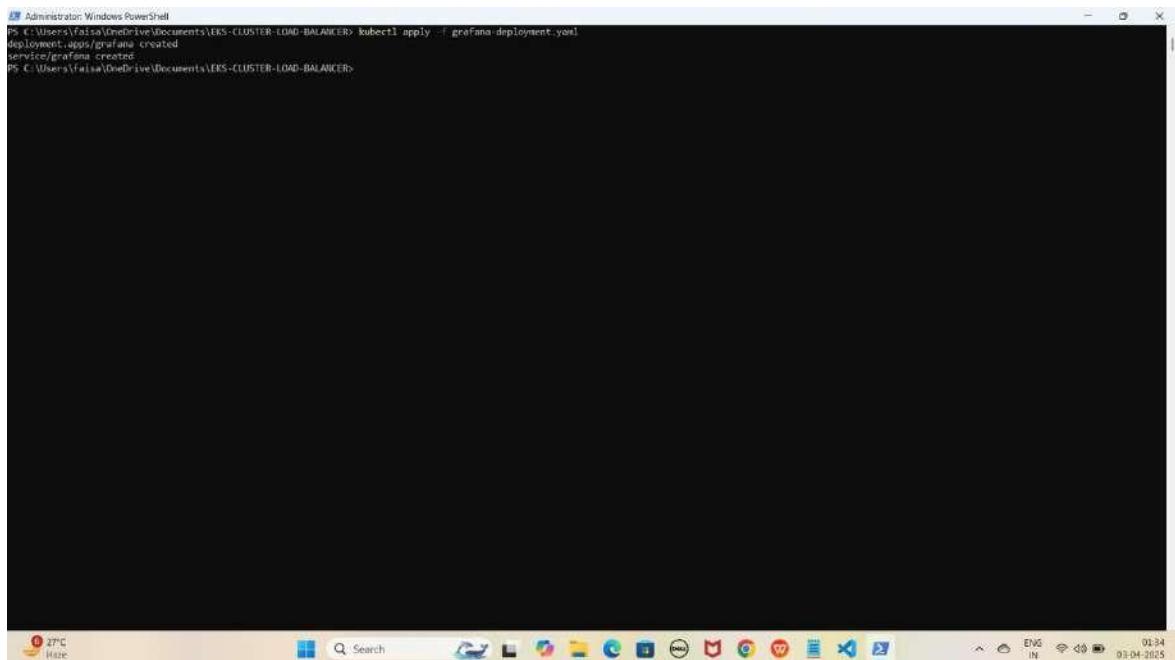
JAISE KI:-

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: grafana
  namespace: monitoring
spec:
  replicas: 1
  selector:
    matchLabels:
      app: grafana
  template:
    metadata:
      labels:
        app: grafana
    spec:
      containers:
        - name: grafana
          image: grafana/grafana:8.3.0
          ports:
            - containerPort: 3000
          env:
            - name: GF_SECURITY_ADMIN_PASSWORD
              value: "admin"
---
apiVersion: v1
kind: Service
metadata:
  name: grafana
  namespace: monitoring
spec:
  type: NodePort    # Change the type from ClusterIP to NodePort
  ports:
    - port: 80
      targetPort: 3000
      nodePort: 30080    # Specify the NodePort (e.g., 30080, any unused port in the range
30000-32767)
  selector:
    app: grafana
```

## Grafana Deployment Apply Karo

```
kubectl apply -f grafana-deployment.yaml
```

YE KUCH ISTARHA LAGEGA

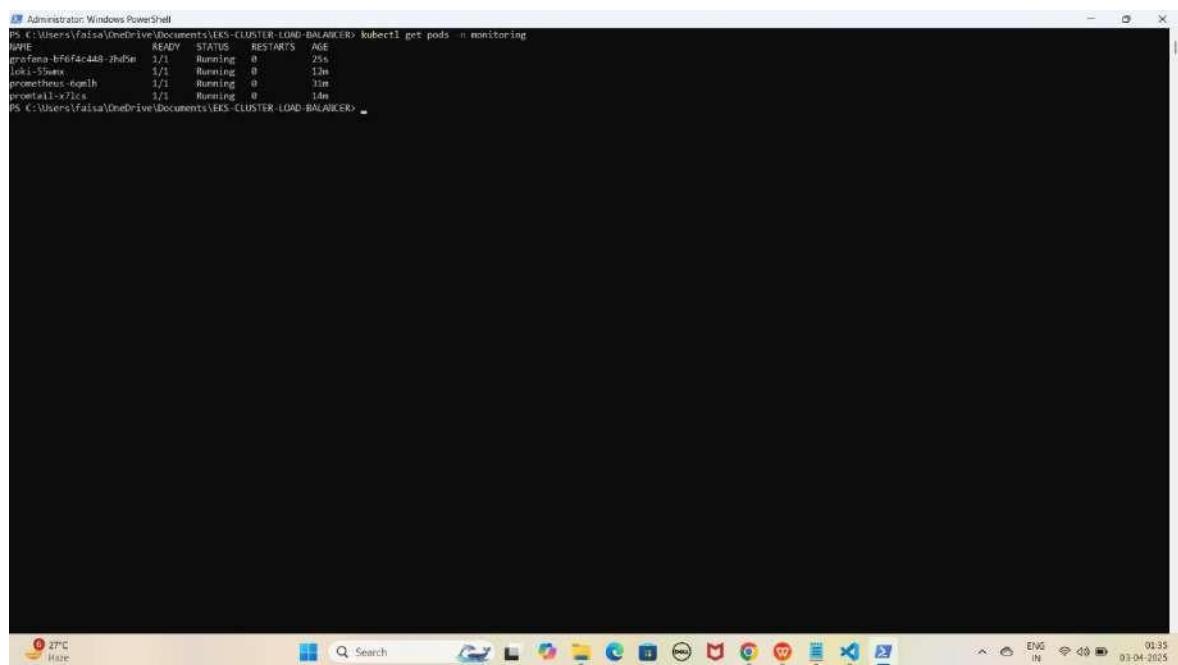


```
Administrator: Windows PowerShell
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl apply -f grafana-deployment.yaml
deployment.apps/grafana created
service/grafana created
PS C:\Users\faisa\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

## 1. Pods check karne ke liye ye command run kariye

kubectl get pods -n monitoring

YE KUCH ISTARHA LAGEGA



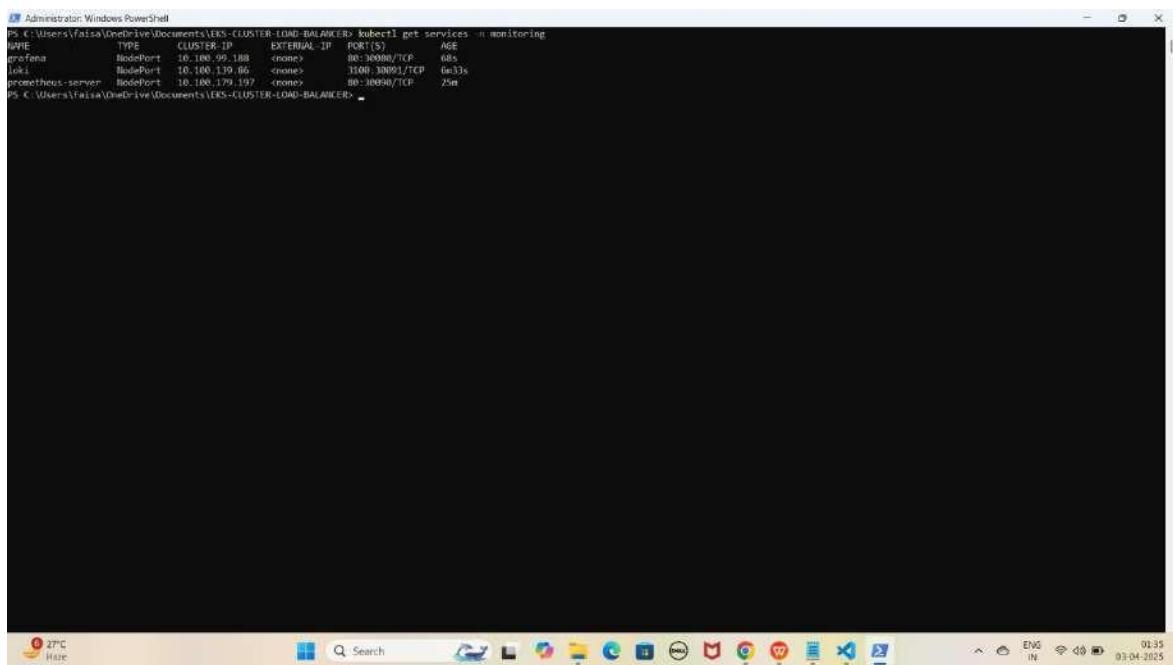
```
Administrator: Windows PowerShell
PS C:\Users\faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl get pods -n monitoring
NAME          READY   STATUS    RESTARTS   AGE
grafana-bf6f4c448-zhd8e   1/1     Running   0          25s
loki-5946      1/1     Running   0          1m
prometheus-0gjrh   1/1     Running   0          1m
telegraf-11-x7ics  1/1     Running   0          1m
PS C:\Users\faisal\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER>
```

**NOTE: Agar STATE Running show karraha hai to sab kuch sahi hai**

## 2. Services check karne ke liye ye commad run kariye

kubectl get services -n monitoring

YE KUCH ISTARHA LAGEGA

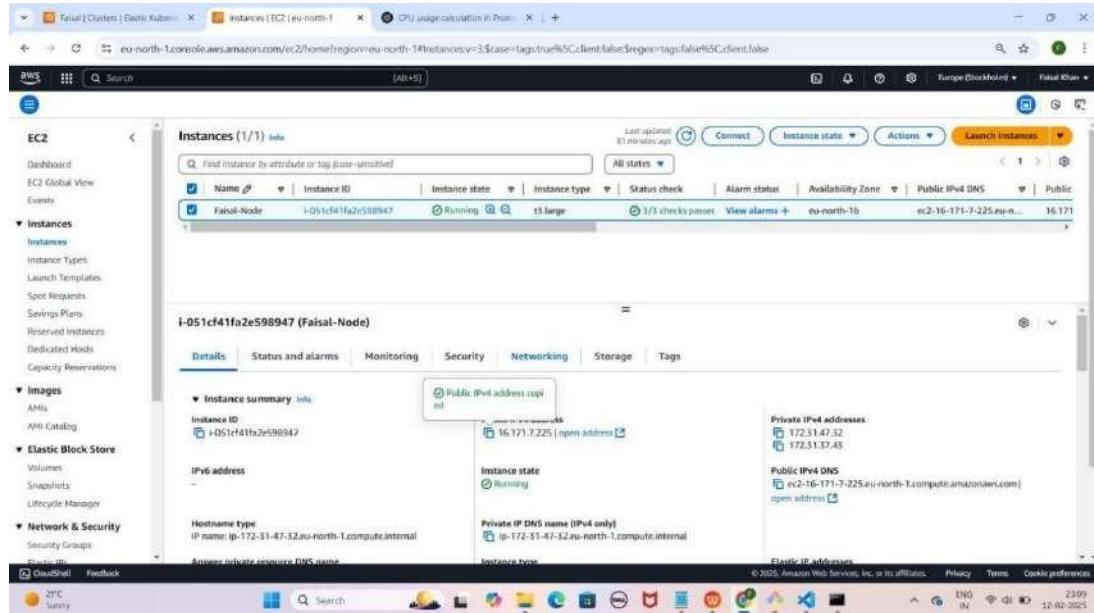


```
Administrator: Windows PowerShell
PS C:\Users\Vaibhav\OneDrive\Documents\EKS-CLUSTER-LOAD-BALANCER> kubectl get services -n monitoring
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
grafana   NodePort   10.100.99.188   <none>          80:30080/TCP   6d85s
loki      NodePort   10.100.139.86    <none>          3100:30091/TCP   6d33s
prometheus-server   NodePort   10.100.179.197   <none>          90:30090/TCP   29m
```

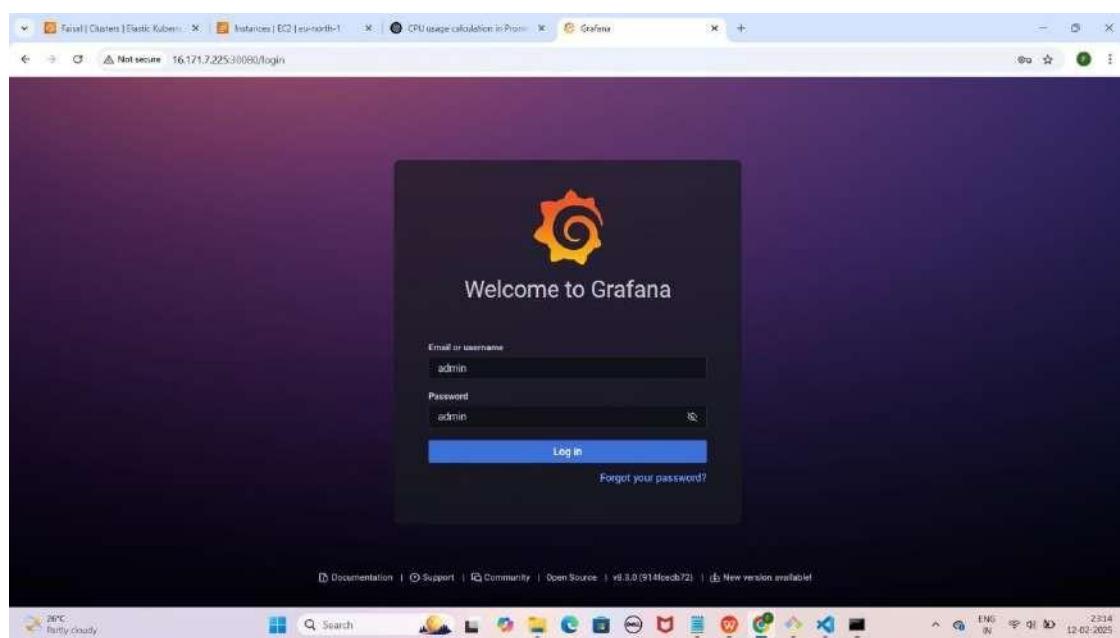
**NOTE:** Ab aapka Grafana 30080 port par expose ho chuka hai. Browser me Cluster Node ki Public IP ke saath check karein jaise ki mere case me kuch aisa hogा.

**Grafana:** <http://EKS-Node-Public IP:30080>

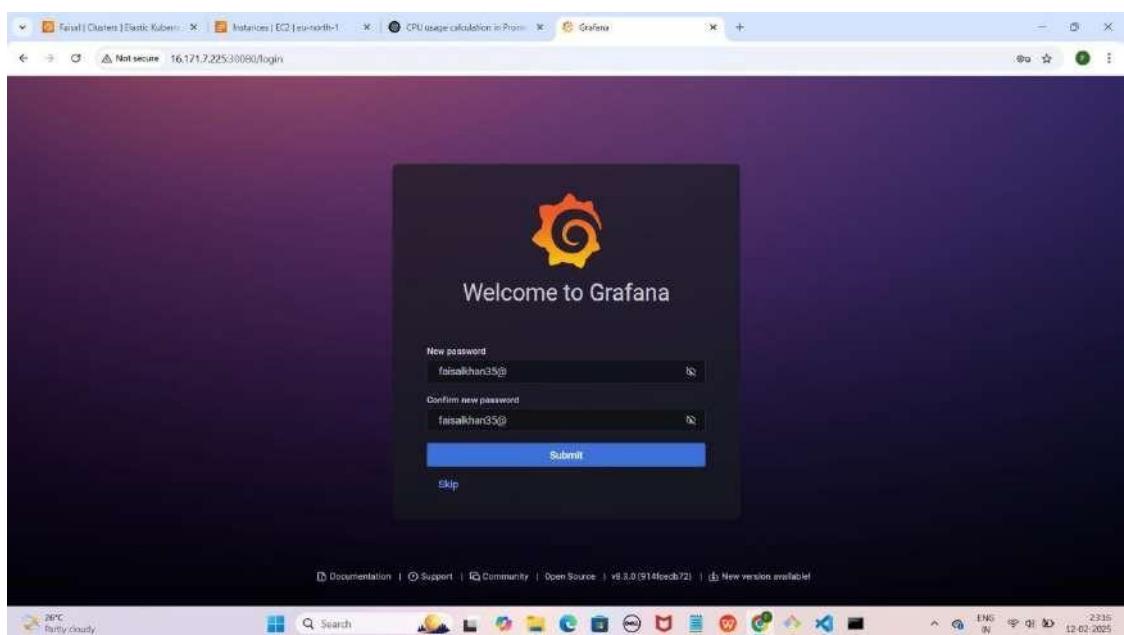
# YE KUCH ISTARHA LAGEGA



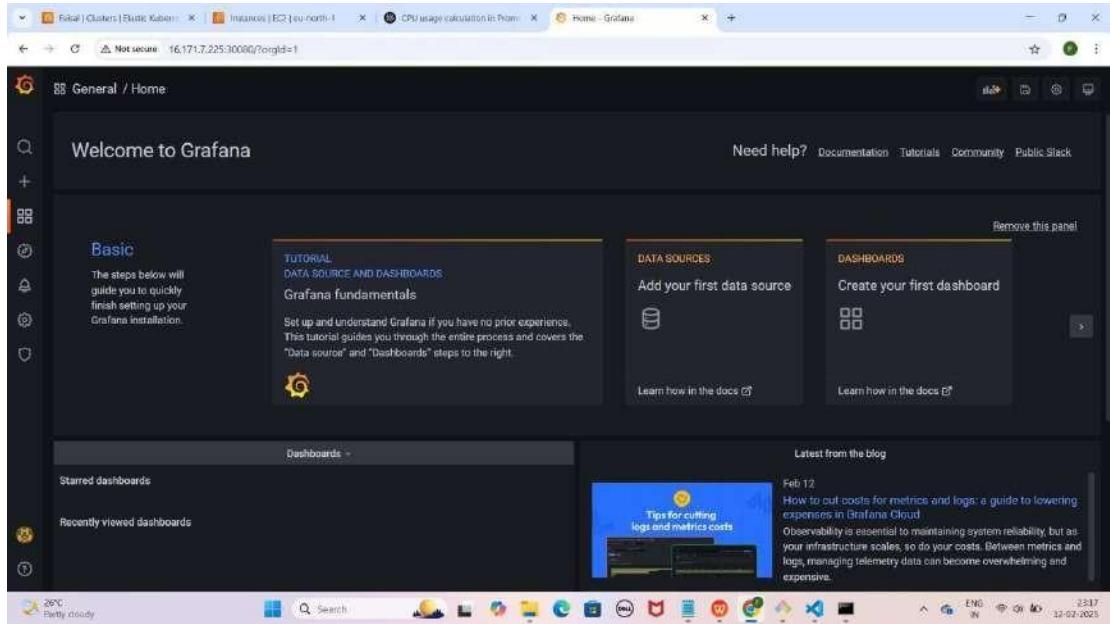
## YE KUCH ISTARHA LAGEGA



**NOTE : Grafana browser me run hone ke baad aapse login ke liye username aur password maangega. Default username aur password admin hota hai. Login ke baad aap apna password change kar sakte hain**



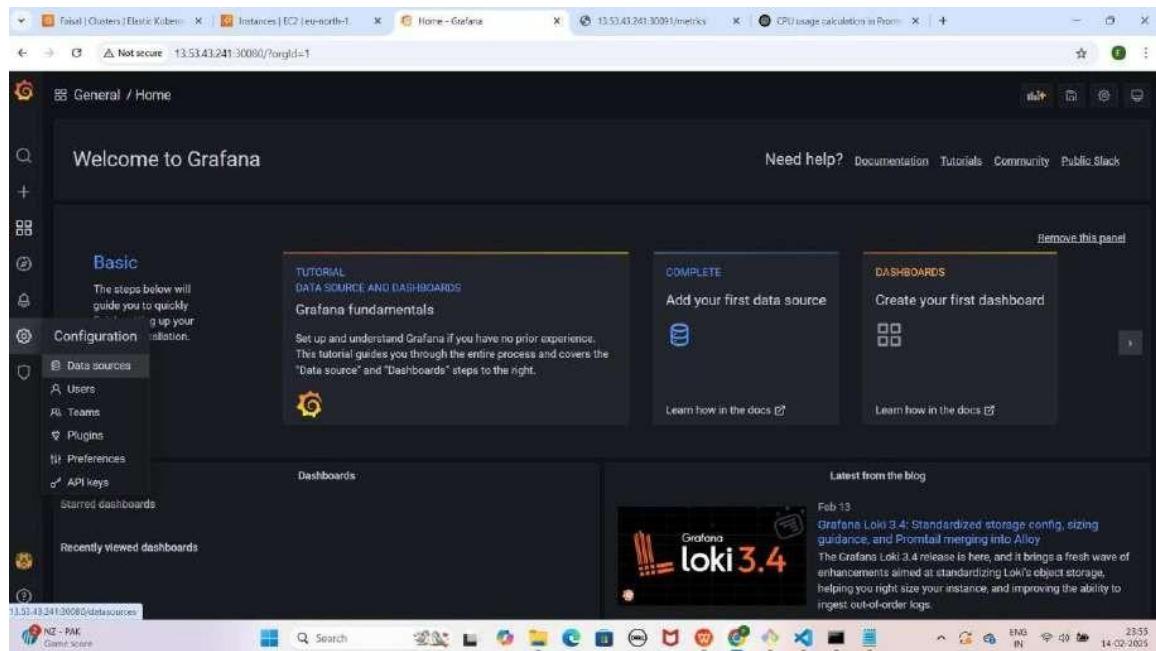
**NOTE : Is step par aapko apne pasand ka naya password set karsakte hai.  
Yaha mai faisalkhan35@ password set kar raha hoon.Password type  
karne ke baad Submit par click kariye**

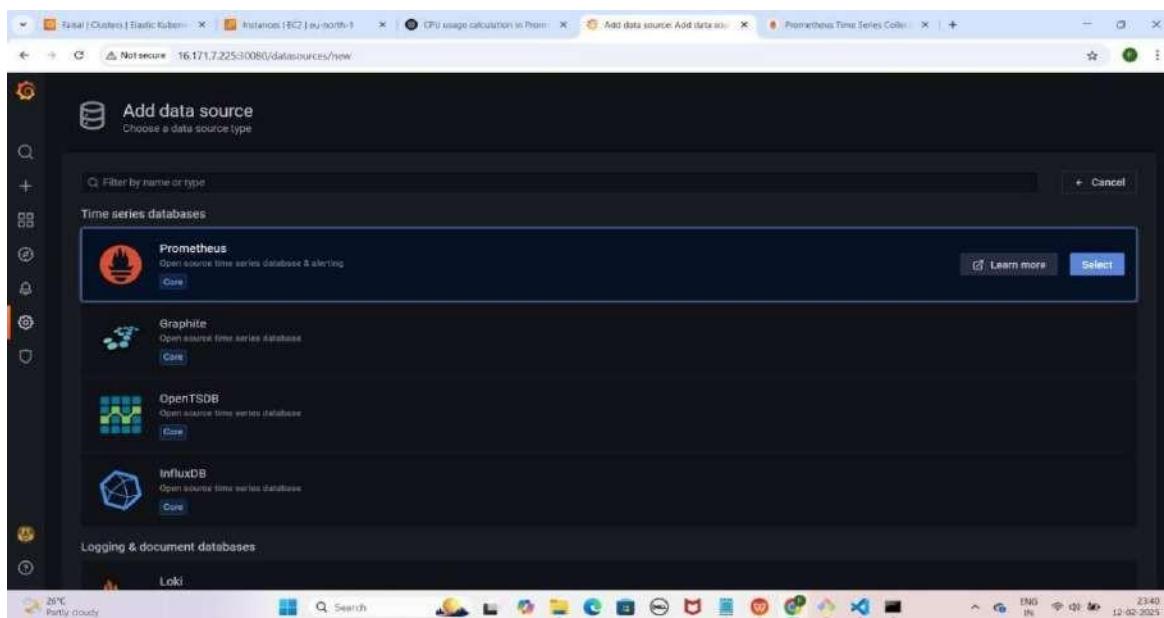
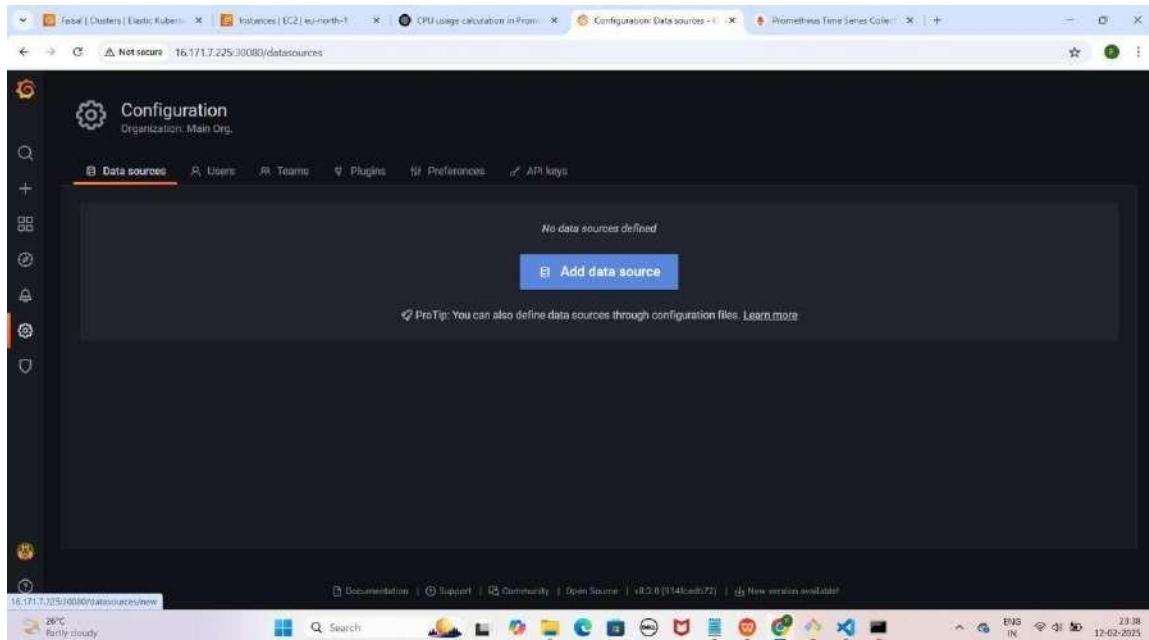


## Step 8: Grafana Mein Prometheus Data Source Add Karna, Queries Add Karna Aur Dashboard Create Karna

- **Grafana dashboard open karo.**
- **Configuration pe click karo, phir Data Sources select karo.**
- **Add data source pe click karo.**
- **Prometheus select karo.**
- **Browser open karo aur yeh URL copy karo:**
  - Mere case me yeh kuch is tarah hogi: <http://EKS-Node-Public-IP:30090/>
  - Aapke case me Public IP different ho sakti hai.
- **HTTPS section me URL box me paste karo.**
- Neeche scroll karo aur "**Save & Test**" pe click karo.

YE KUCH ISTARHA LAGEGA





The screenshot shows the 'Data Sources / Prometheus' configuration page in Grafana. The 'Settings' tab is selected. A prominent message at the top says: 'Configure your Prometheus data source below. Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the free forever Grafana Cloud plan.' Below this, the 'Name' field is set to 'Prometheus' and the 'Default' toggle is turned on. Under the 'HTTP' section, the 'URL' is set to 'http://localhost:9090'. The 'Auth' section contains three options: 'Basic auth' (disabled), 'TLS Client Auth' (disabled), and 'Skip TLS Verify' (disabled). The status bar at the bottom right shows '23:40' and '12-02-2025'.

The screenshot shows the Prometheus UI with the 'Graph' panel selected. The URL in the address bar is '16.171.7.225:9090'. The top navigation bar includes 'Prometheus', 'Alerts', 'Graph', 'Status', 'Help', and 'Classic UI'. Below the navigation, there are checkboxes for 'Use local time', 'Enable query history', 'Enable autocomplete', 'Enable highlighting', and 'Enable linter'. A search bar contains the placeholder 'Expression (press Shift+Enter for newlines)'. The main area displays the message 'No data queried yet'. At the bottom left is a blue 'Add Panel' button, and at the bottom right is a 'Remove Panel' link. The status bar at the bottom right shows '23:40' and '12-02-2025'.



Faist | Clusters | Elastic Kube... Instances | EC2 | eu-north-1 CPU usage calculation in Prometheus Settings - Grafana Prometheus Time Series Collector

Data Sources / Prometheus Type: Prometheus

Settings Dashboards

Configure your Prometheus data source below Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the free forever Grafana Cloud plan.

Name: Prometheus Default:

HTTP

URL: http://16.171.7.225:30090/ Access: Server (default) Help

Allowed cookies: New tag (enter key to add)

Timeout:

Auth

Basic auth: With Credentials With CA Cert Skip TLS Verify

29PC Fairly cloudy Search ENG IN 12-02-2025 2341

Faist | Clusters | Elastic Kube... Instances | EC2 | eu-north-1 CPU usage calculation in Prometheus Settings - Grafana Prometheus Time Series Collector

Alerting

Manage alerts via Alerting UI:

Alertmanager data source: Choose

Script interval: 1s Query timeout: 60s HTTP Method: POST

Misc

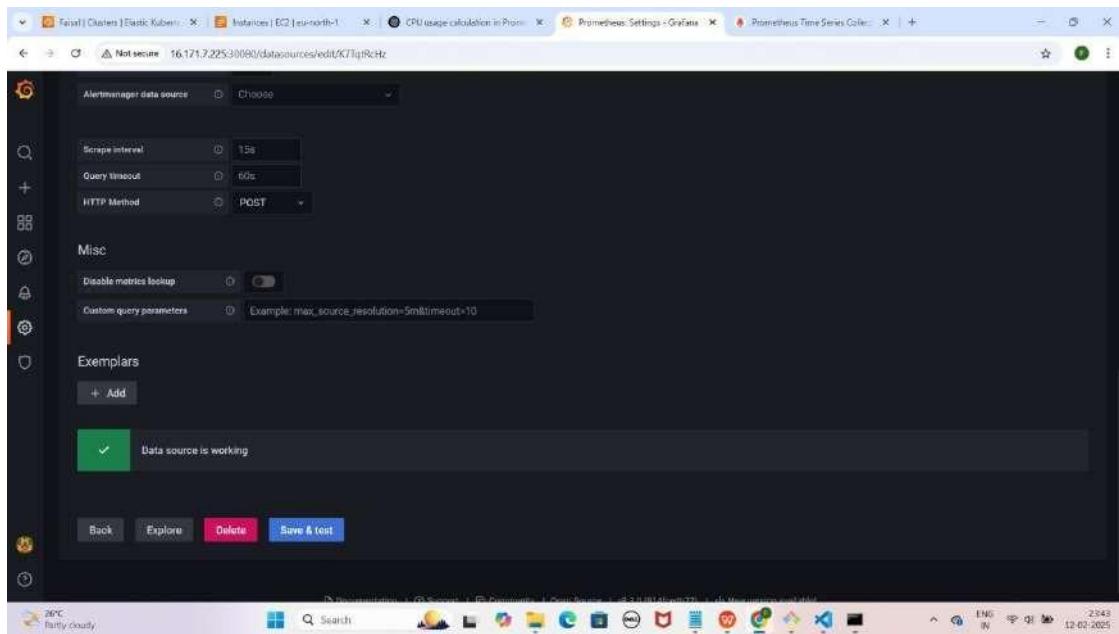
Disable metrics lookup:  Custom query parameters: Example: max\_source\_resolution=5m&timeout=10

Exemplars

+ Add Back Explore Delete Save & test

Documentation | Support | Community | Open Source | v8.3.0 (914fcd272) | New version available

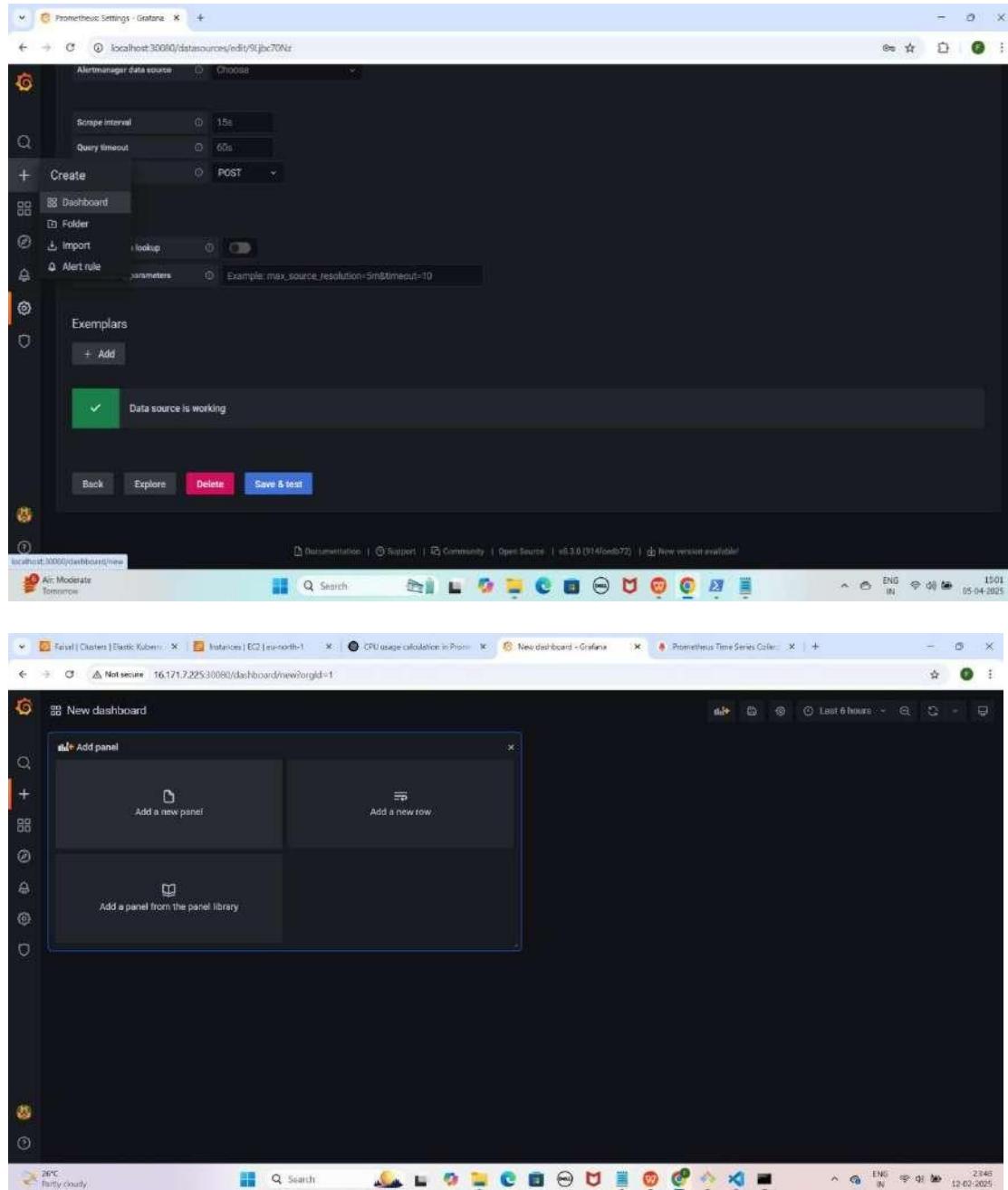
29PC Fairly cloudy Search ENG IN 12-02-2025 2342

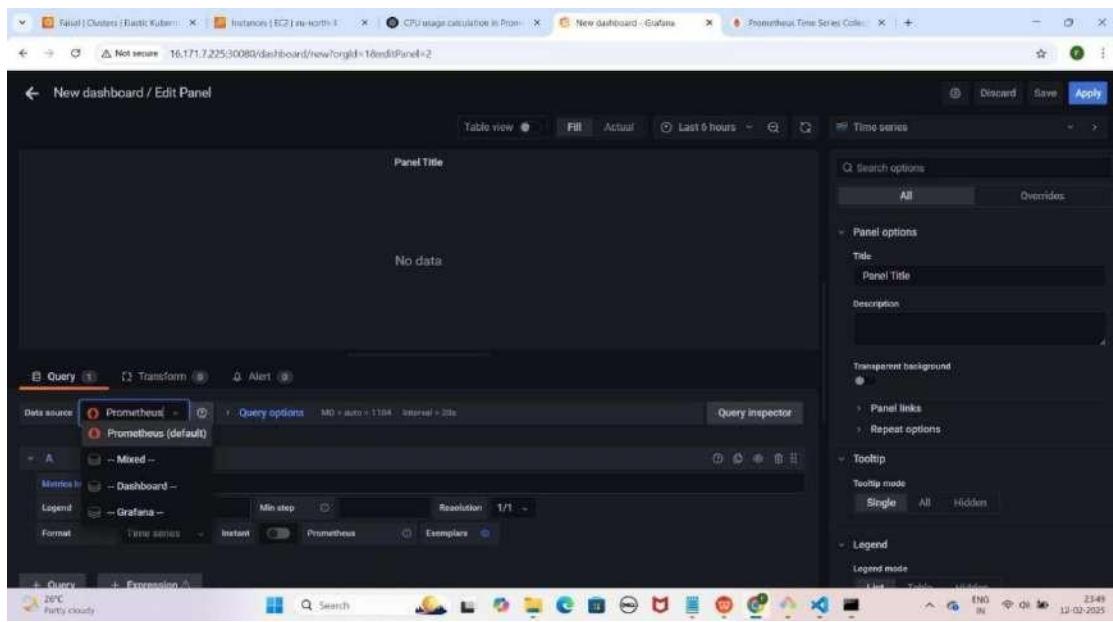


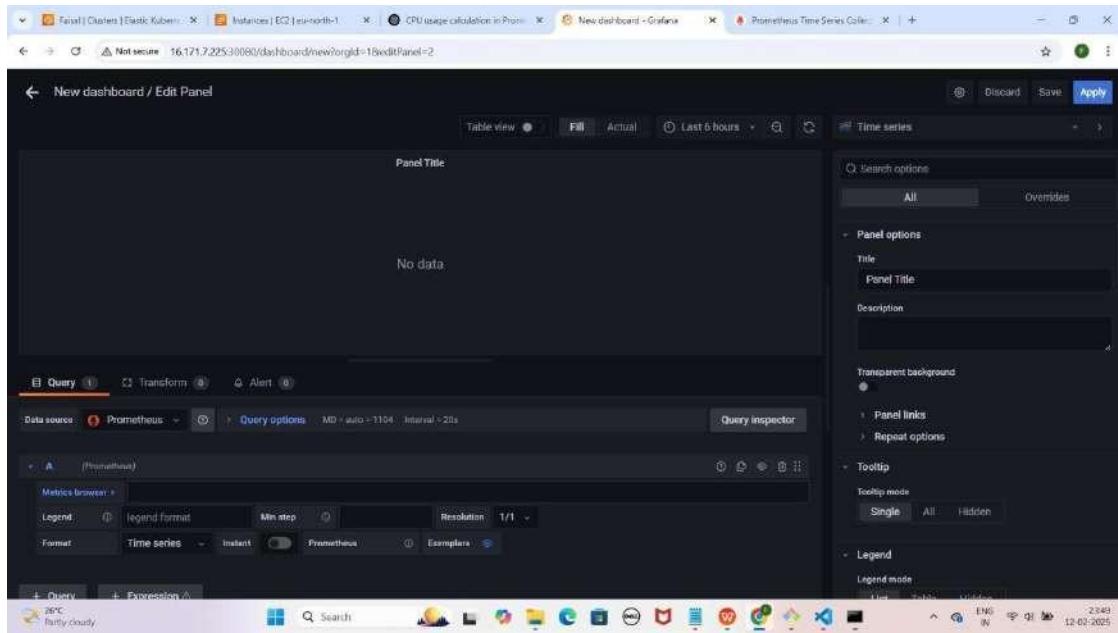
**NOTE : Save & test pe click karne ke baad aapko Data source is working  
karke pop up aayega**

- Prometheus ke Metrics visualize karne ke liye "+" (plus) icon pe click karo.
- Dashboard select karo aur "Add a new panel" pe click karo.
- Query section me Data Source ko "Prometheus" select karo.

YE KUCH ISTARHA LAGEGA





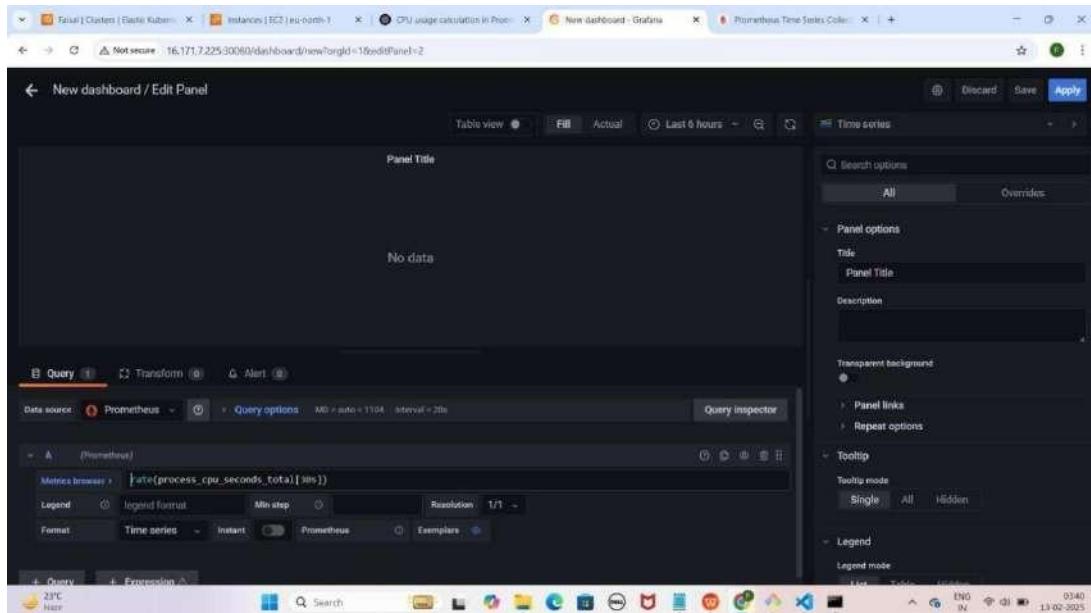


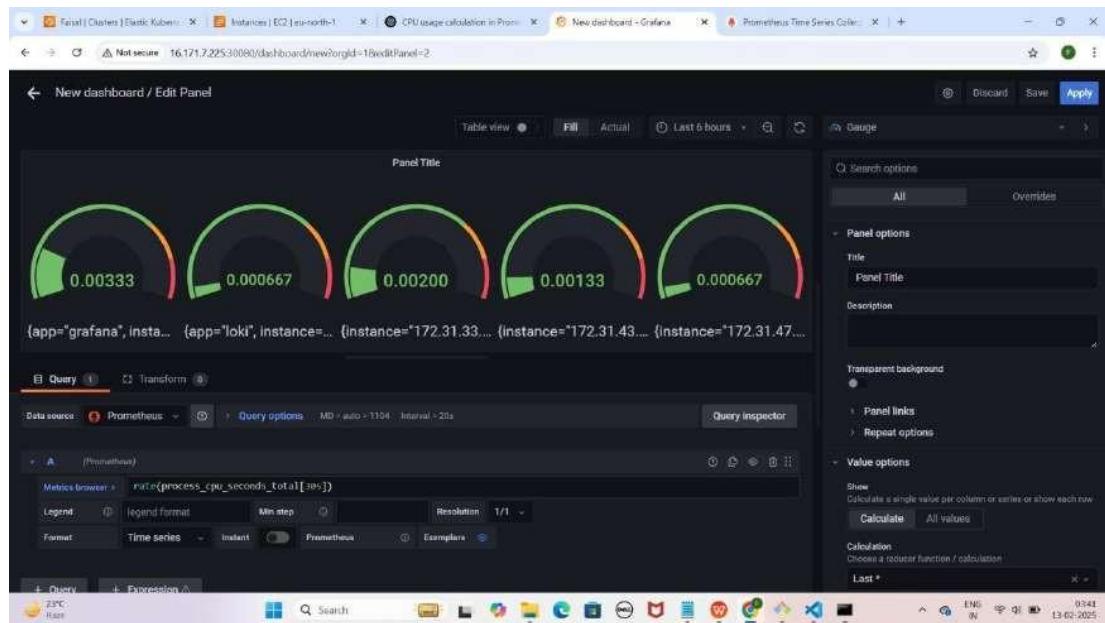
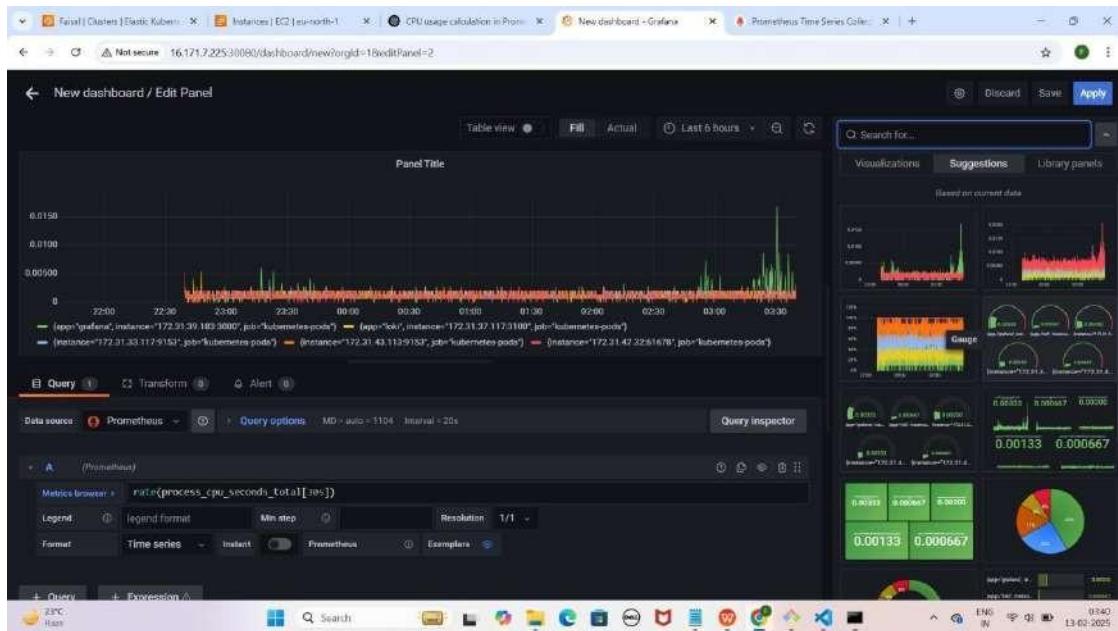
## 1. Prometheus ko data source select karne ke baad, CPU Usage dekhne ke liye Metric Browser me yeh query paste karein

rate(process\_cpu\_seconds\_total[30s])

- **"Time Series"** pe click karein.
- **Suggestions** me se **"Dashboard"** select karein jaise maine select kiya hai.

## YE KUCH ISTARHA LAGEGA

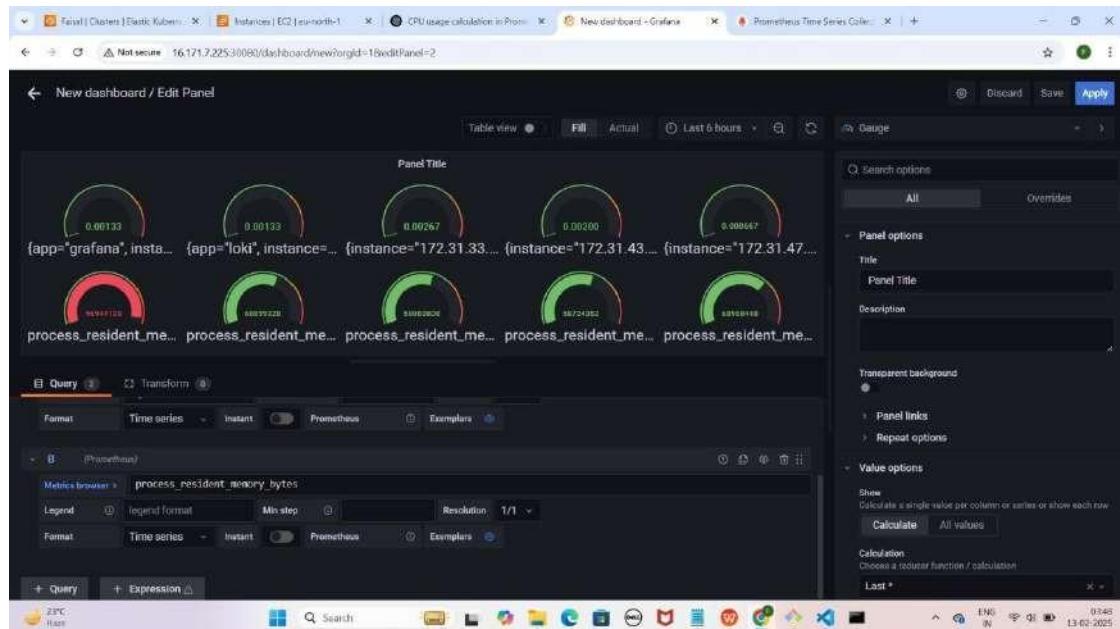




**2. Memory Usage dekhne ke liye "+ Query" par click karein.  
Yeh query paste karein taake memory usage dekh sakein**

process\_resident\_memory\_bytes

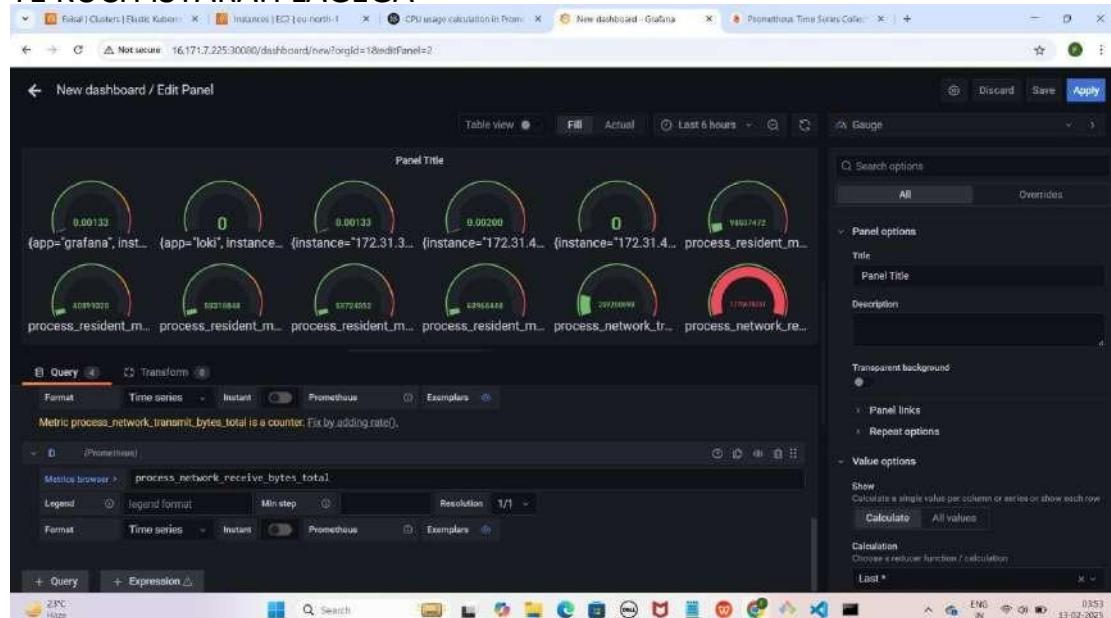
YE KUCH ISTARHA LAGEGA



**3. Network Transmit dekhne ke liye "+ Query" par click karein.  
Yeh query paste karein taake network transmit dekh sakein**

process\_network\_transmit\_bytes\_total

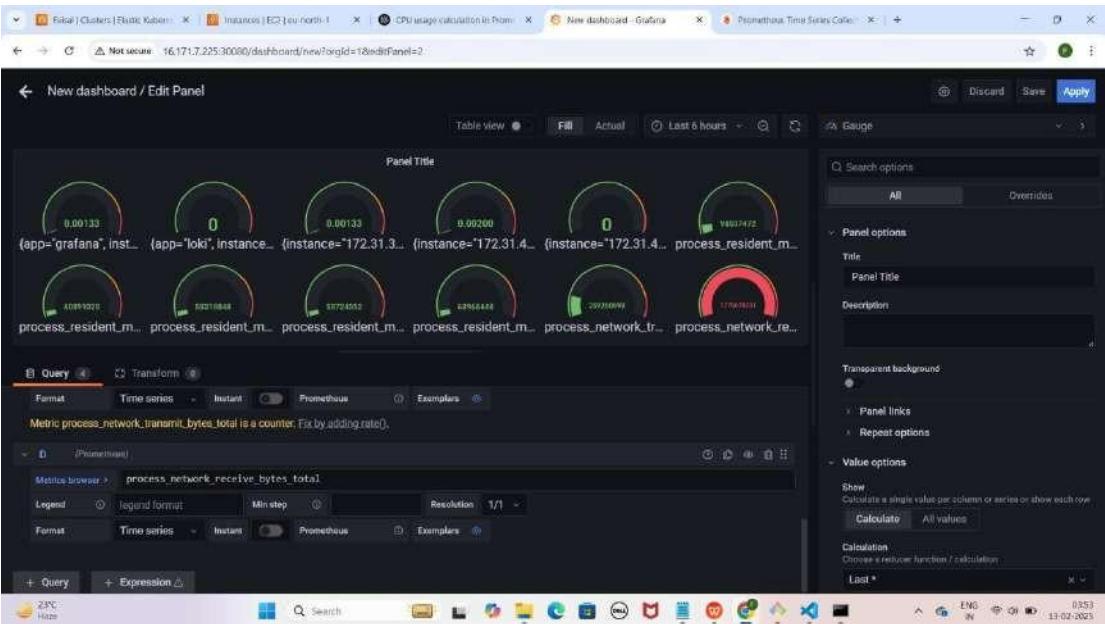
YE KUCH ISTARAH LAGEGA



**4. Network Receive dekhne ke liye "+ Query" par click karein.  
Yeh query paste karein taake network receive dekh sakein:**

process\_network\_receive\_bytes\_total

YE KUCH ISTARHA LAGEGA

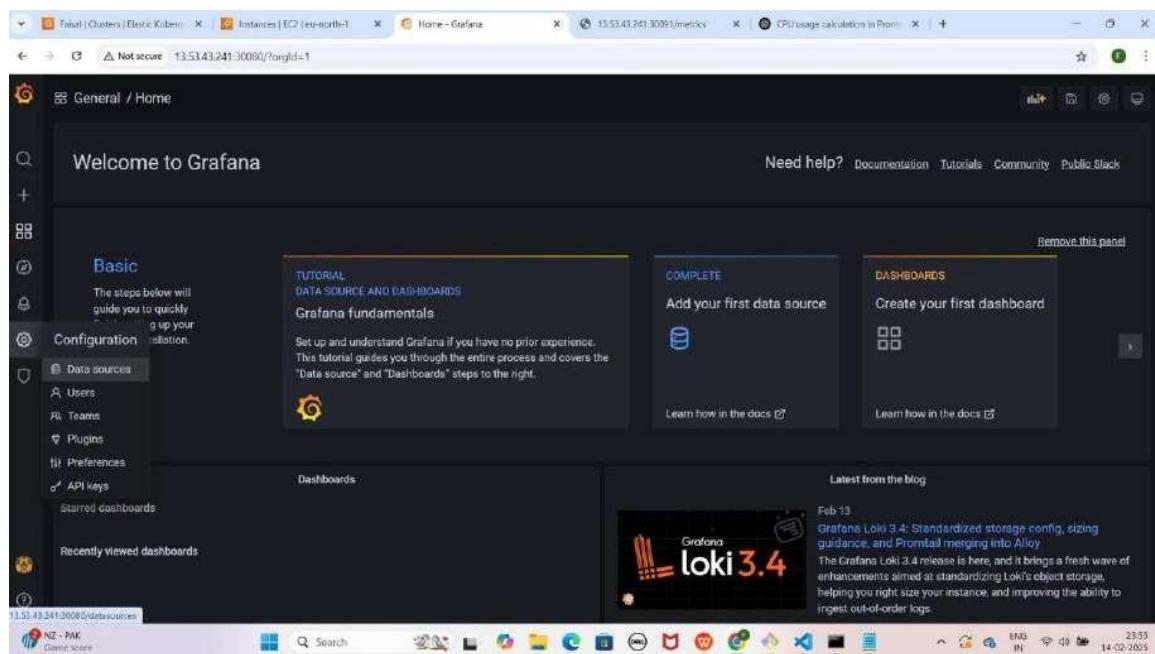


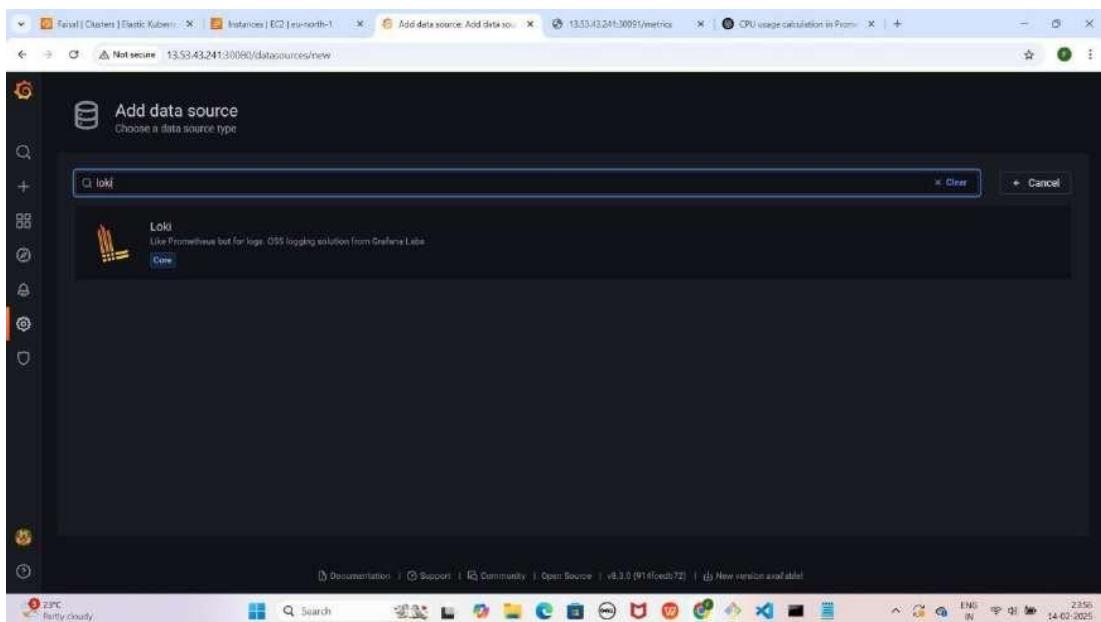
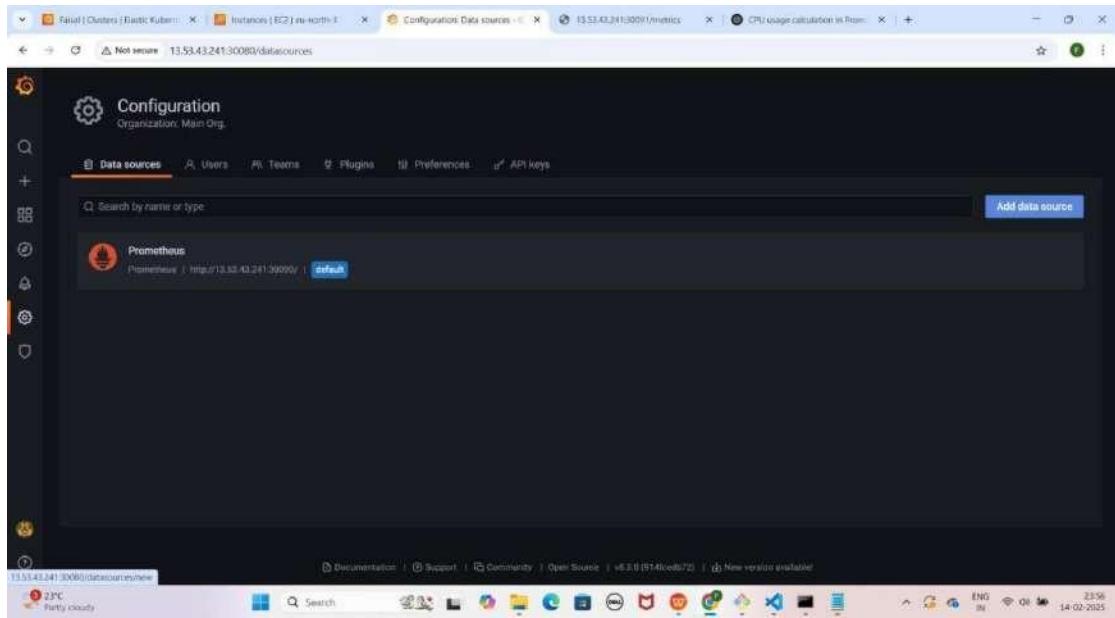
**NOTE : Itna karne ke baad aap 30 sec ke interval par CPU usage, current Memory Usage aur Network transmit/receive bytes ko Monitor kar sakte hain.**

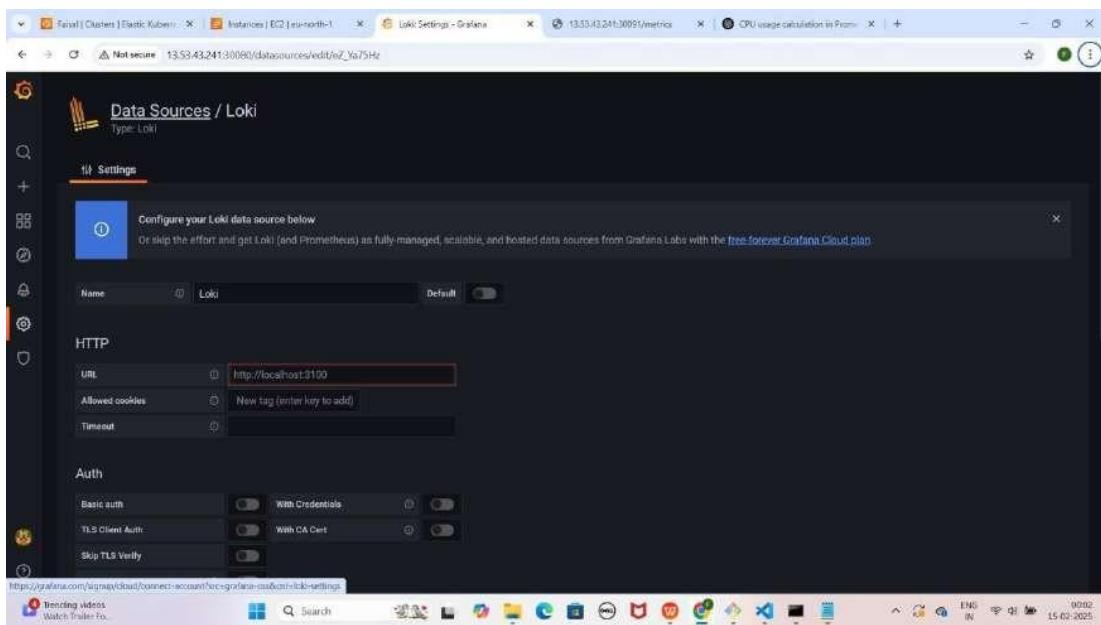
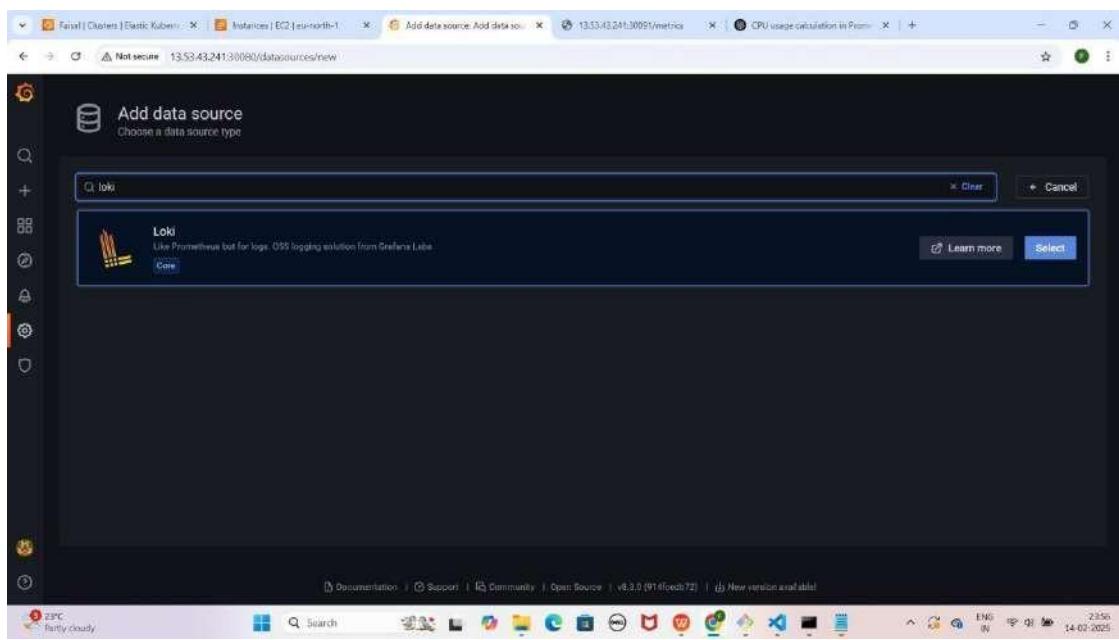
## Step 9: Grafana Mein Loki Data Source Add Karna Aur Dashboard Create Karna

- **Grafana** open karo aur **Settings** icon pe click karo.
- **Data Sources** select karo.
- **Add data source** pe click karo.
- Search bar me "**Loki**" likho aur **Loki** select karo.
- Browser open karo aur yeh URL copy karo:
  - **Mere case me yeh kuch is tarah hogi:** <http://EKS-Node-Public-IP:30091/>
  - **Aapke case me Public IP different ho sakti hai.**
- **HTTPS section** me URL box me paste karo.
- Neeche scroll karo aur "**Save & Test**" pe click karo.

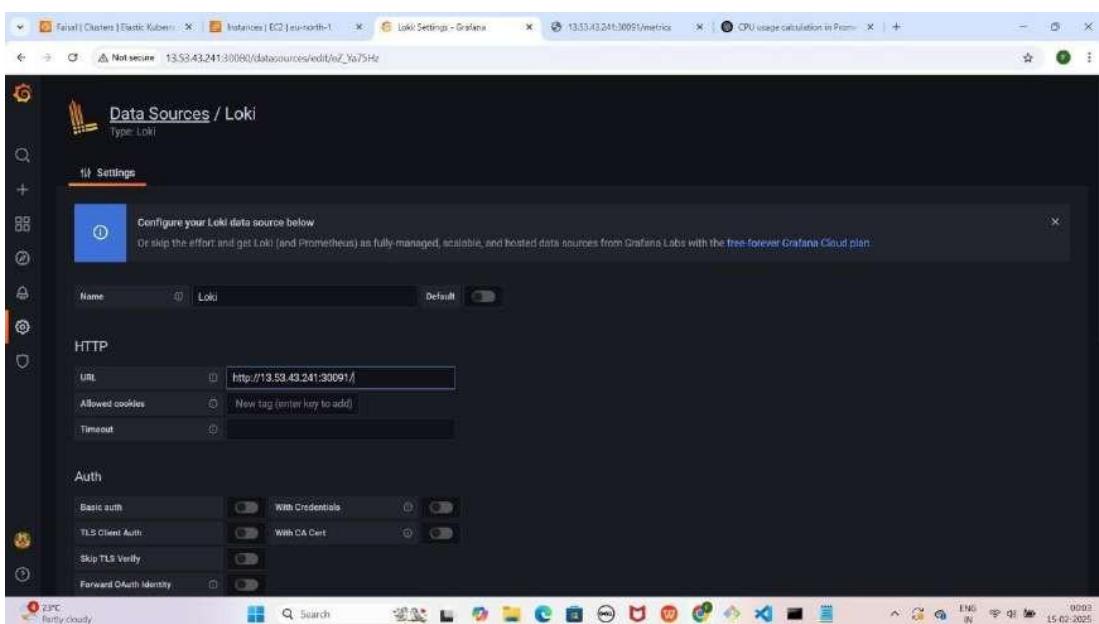
YE KUCH ISTARHA LAGEGA

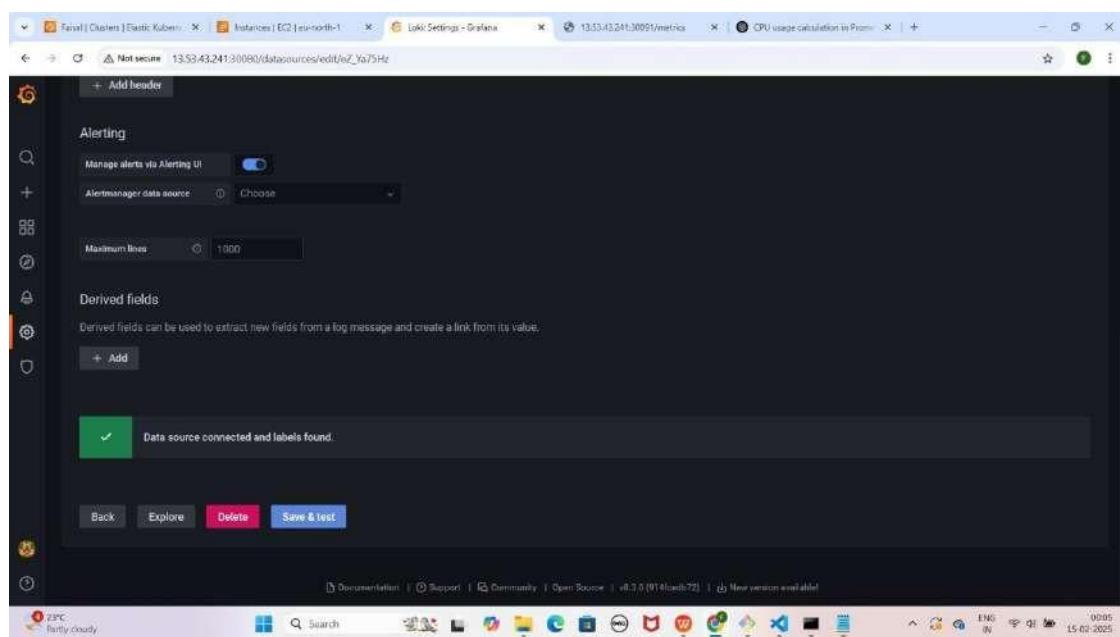
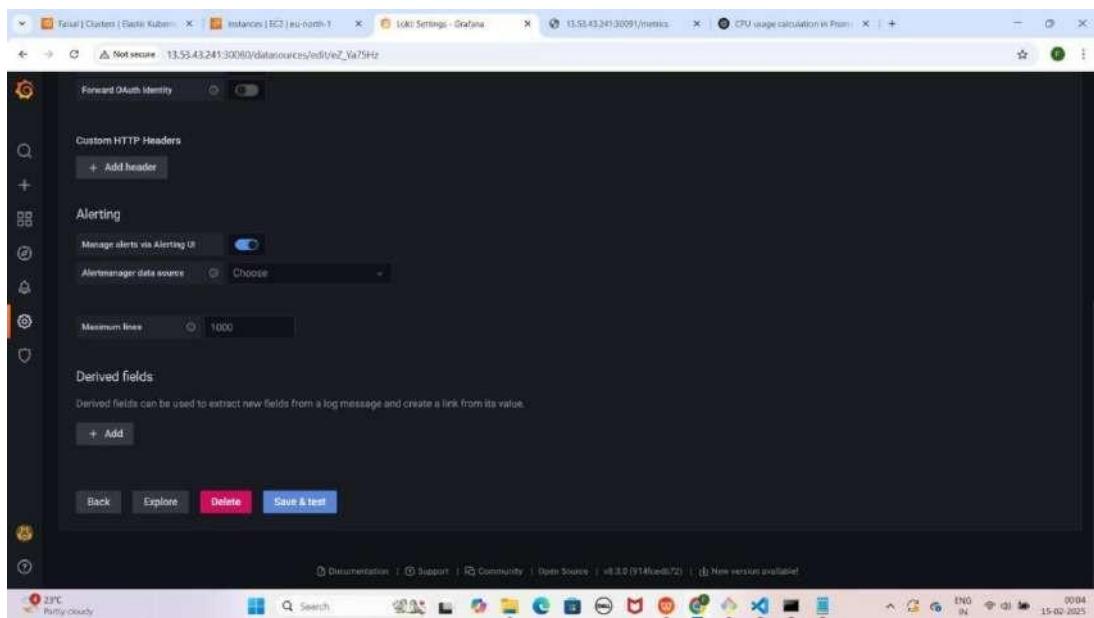






```
# HELP cortex_cortex_index_entries_per_query Distribution of #series (post_intersection) per query.
# TYPE cortex_index_entries_per_query histogram
cortex_index_entries_per_query_bucket{le="1"} 26
cortex_index_entries_per_query_bucket{le="2"} 26
cortex_index_entries_per_query_bucket{le="4"} 26
cortex_index_entries_per_query_bucket{le="8"} 26
cortex_index_entries_per_query_bucket{le="16"} 26
cortex_index_entries_per_query_bucket{le="32"} 26
cortex_index_entries_per_query_bucket{le="64"} 26
cortex_index_entries_per_query_bucket{le="128"} 26
cortex_index_entries_per_query_bucket{le="256"} 26
cortex_index_entries_per_query_bucket{le="512"} 26
cortex_index_entries_per_query_bucket{le="1024"} 26
cortex_index_entries_per_query_bucket{le="2048"} 26
cortex_index_entries_per_query_bucket{le="4096"} 26
cortex_index_entries_per_query_bucket{le="8192"} 26
cortex_index_entries_per_query_bucket{le="16384"} 26
cortex_index_entries_per_query_bucket{le="32768"} 26
cortex_index_entries_per_query_bucket{le="65536"} 26
cortex_index_entries_per_query_bucket{le="131072"} 26
cortex_index_entries_per_query_bucket{le="262144"} 26
cortex_index_entries_per_query_bucket{le="524288"} 26
cortex_index_entries_per_query_bucket{le="1048576"} 26
cortex_index_entries_per_query_bucket{le="2097152"} 26
cortex_index_entries_per_query_bucket{le="4194304"} 26
cortex_index_entries_per_query_bucket{le="8388608"} 26
cortex_index_entries_per_query_bucket{le="16777216"} 26
cortex_index_entries_per_query_bucket{le="33554432"} 26
cortex_index_entries_per_query_bucket{le="67108864"} 26
cortex_index_entries_per_query_bucket{le="134217728"} 26
cortex_index_entries_per_query_bucket{le="268435456"} 26
cortex_index_entries_per_query_bucket{le="536870912"} 26
cortex_index_entries_per_query_bucket{le="1073741824"} 26
cortex_index_entries_per_query_bucket{le="2147483648"} 26
cortex_index_entries_per_query_bucket{le="4294967296"} 26
cortex_index_entries_per_query_bucket{le="8589934592"} 26
cortex_index_entries_per_query_bucket{le="17179869184"} 26
cortex_index_entries_per_query_bucket{le="34359738368"} 26
cortex_index_entries_per_query_bucket{le="68719476736"} 26
cortex_index_entries_per_query_bucket{le="137438953472"} 26
cortex_index_entries_per_query_bucket{le="274877906944"} 26
cortex_index_entries_per_query_bucket{le="549755813888"} 26
cortex_index_entries_per_query_bucket{le="1099511627776"} 26
cortex_index_entries_per_query_bucket{le="2199023255552"} 26
cortex_index_entries_per_query_bucket{le="4398046511104"} 26
cortex_index_entries_per_query_bucket{le="8796093022208"} 26
cortex_index_entries_per_query_bucket{le="17592186044416"} 26
cortex_index_entries_per_query_bucket{le="35184372088832"} 26
cortex_index_entries_per_query_bucket{le="70368744177664"} 26
cortex_index_entries_per_query_bucket{le="140737488355328"} 26
cortex_index_entries_per_query_bucket{le="281474976710656"} 26
cortex_index_entries_per_query_bucket{le="562949953421312"} 26
cortex_index_entries_per_query_bucket{le="1125899906842640"} 26
cortex_index_entries_per_query_bucket{le="2251799813685280"} 26
cortex_index_entries_per_query_bucket{le="4503599627370560"} 26
cortex_index_entries_per_query_bucket{le="9007199254741120"} 26
cortex_index_entries_per_query_bucket{le="18014398509482240"} 26
cortex_index_entries_per_query_bucket{le="36028797018964480"} 26
cortex_index_entries_per_query_bucket{le="72057594037928960"} 26
cortex_index_entries_per_query_bucket{le="144115188075857920"} 26
cortex_index_entries_per_query_bucket{le="288230376151715840"} 26
cortex_index_entries_per_query_bucket{le="576460752303431680"} 26
cortex_index_entries_per_query_bucket{le="1152921504606863360"} 26
cortex_index_entries_per_query_bucket{le="2305843009213726720"} 26
cortex_index_entries_per_query_bucket{le="4611686018427453440"} 26
cortex_index_entries_per_query_bucket{le="9223372036854906880"} 26
cortex_index_entries_per_query_bucket{le="18446744073709817600"} 26
cortex_index_entries_per_query_bucket{le="36893488147419635200"} 26
cortex_index_entries_per_query_bucket{le="73786976294839270400"} 26
cortex_index_entries_per_query_bucket{le="147573952589678540800"} 26
cortex_index_entries_per_query_bucket{le="295147905179357081600"} 26
cortex_index_entries_per_query_bucket{le="590295810358714163200"} 26
cortex_index_entries_per_query_bucket{le="1180591620717428326400"} 26
cortex_index_entries_per_query_bucket{le="2361183241434856652800"} 26
cortex_index_entries_per_query_bucket{le="4722366482869713305600"} 26
cortex_index_entries_per_query_bucket{le="9444732965739426611200"} 26
cortex_index_entries_per_query_bucket{le="18889465931478853222400"} 26
cortex_index_entries_per_query_bucket{le="37778931862957706444800"} 26
cortex_index_entries_per_query_bucket{le="75557863725915412889600"} 26
cortex_index_entries_per_query_bucket{le="151115727458230825779200"} 26
cortex_index_entries_per_query_bucket{le="302231454916461651558400"} 26
cortex_index_entries_per_query_bucket{le="604462909832923203116800"} 26
cortex_index_entries_per_query_bucket{le="1208925819665846406233600"} 26
cortex_index_entries_per_query_bucket{le="2417851639331692812467200"} 26
cortex_index_entries_per_query_bucket{le="4835703278663385624934400"} 26
cortex_index_entries_per_query_bucket{le="9671406557326771249868800"} 26
cortex_index_entries_per_query_bucket{le="19342813114653542497737600"} 26
cortex_index_entries_per_query_bucket{le="38685626229307084995475200"} 26
cortex_index_entries_per_query_bucket{le="77371252458614169980950400"} 26
cortex_index_entries_per_query_bucket{le="154742504917228339961900800"} 26
cortex_index_entries_per_query_bucket{le="309485009834456679923801600"} 26
cortex_index_entries_per_query_bucket{le="618970019668913359847603200"} 26
cortex_index_entries_per_query_bucket{le="1237940039337826719695206400"} 26
cortex_index_entries_per_query_bucket{le="2475880078675653439390412800"} 26
cortex_index_entries_per_query_bucket{le="4951760157351306878780825600"} 26
cortex_index_entries_per_query_bucket{le="9903520314702613757561651200"} 26
cortex_index_entries_per_query_bucket{le="19807040629405227515123202400"} 26
cortex_index_entries_per_query_bucket{le="39614081258810455030246404800"} 26
cortex_index_entries_per_query_bucket{le="79228162517620910060492809600"} 26
cortex_index_entries_per_query_bucket{le="158456325035241820120956019200"} 26
cortex_index_entries_per_query_bucket{le="316912650070483640241912038400"} 26
cortex_index_entries_per_query_bucket{le="633825300140967280483824076800"} 26
cortex_index_entries_per_query_bucket{le="1267650600281935560967648153600"} 26
cortex_index_entries_per_query_bucket{le="2535301200563871121935296307200"} 26
cortex_index_entries_per_query_bucket{le="5070602401127742243870592614400"} 26
cortex_index_entries_per_query_bucket{le="10141204802455484487741185228800"} 26
cortex_index_entries_per_query_bucket{le="20282409604910968975482370457600"} 26
cortex_index_entries_per_query_bucket{le="40564819209821937950964740915200"} 26
cortex_index_entries_per_query_bucket{le="81129638419643875901929481830400"} 26
cortex_index_entries_per_query_bucket{le="162259276839287551803858963660800"} 26
cortex_index_entries_per_query_bucket{le="324518553678575103607717927321600"} 26
cortex_index_entries_per_query_bucket{le="649037107357150207215435854643200"} 26
cortex_index_entries_per_query_bucket{le="1298074214714300414430871709286400"} 26
cortex_index_entries_per_query_bucket{le="2596148429428600828861743418572800"} 26
cortex_index_entries_per_query_bucket{le="5192296858857201657723486837145600"} 26
cortex_index_entries_per_query_bucket{le="10384593717714403315446973674291200"} 26
cortex_index_entries_per_query_bucket{le="20769187435428806630893947348582400"} 26
cortex_index_entries_per_query_bucket{le="41538374870857613261787894697164800"} 26
cortex_index_entries_per_query_bucket{le="83076749741715226523575789394329600"} 26
cortex_index_entries_per_query_bucket{le="166153499483430453047151578788659200"} 26
cortex_index_entries_per_query_bucket{le="332306998966860906094303157577318400"} 26
cortex_index_entries_per_query_bucket{le="664613997933721812188606315154636800"} 26
cortex_index_entries_per_query_bucket{le="1329227995867443624377212630309353600"} 26
cortex_index_entries_per_query_bucket{le="2658455991734887248754425260618707200"} 26
cortex_index_entries_per_query_bucket{le="5316911983469774497508850521237414400"} 26
cortex_index_entries_per_query_bucket{le="10633823966939548995017701042474828800"} 26
cortex_index_entries_per_query_bucket{le="21267647933879097990035402084949657600"} 26
cortex_index_entries_per_query_bucket{le="42535295867758195980070804169899315200"} 26
cortex_index_entries_per_query_bucket{le="85070591735516391960141608339798630400"} 26
cortex_index_entries_per_query_bucket{le="170141183471032783920283216679597260800"} 26
cortex_index_entries_per_query_bucket{le="340282366942065567840566433359194521600"} 26
cortex_index_entries_per_query_bucket{le="680564733884131135681132866718389043200"} 26
cortex_index_entries_per_query_bucket{le="1361129467768262271362265733436778086400"} 26
cortex_index_entries_per_query_bucket{le="2722258935536524542724531466873556172800"} 26
cortex_index_entries_per_query_bucket{le="5444517871073049085449062933747112345600"} 26
cortex_index_entries_per_query_bucket{le="10889035742146098170898125867942224612800"} 26
cortex_index_entries_per_query_bucket{le="21778071484292196341796251735884449225600"} 26
cortex_index_entries_per_query_bucket{le="43556142968584392683592503471768898451200"} 26
cortex_index_entries_per_query_bucket{le="87112285937168785367185006943537796802400"} 26
cortex_index_entries_per_query_bucket{le="174224571874337570734370013887075593604800"} 26
cortex_index_entries_per_query_bucket{le="34844914374867514146874002777415118729600"} 26
cortex_index_entries_per_query_bucket{le="69689828749735028293748005554830237459200"} 26
cortex_index_entries_per_query_bucket{le="13937965749947005658749601110966047498400"} 26
cortex_index_entries_per_query_bucket{le="27875931499894011317499202221932094996800"} 26
cortex_index_entries_per_query_bucket{le="55751862999788022634998404443864189936000"} 26
cortex_index_entries_per_query_bucket{le="111503725999576045269968088877728378720000"} 26
cortex_index_entries_per_query_bucket{le="223007451999152090539936177755456757440000"} 26
cortex_index_entries_per_query_bucket{le="446014903998304181079872355510913514880000"} 26
cortex_index_entries_per_query_bucket{le="892029807996608362159744711021827029760000"} 26
cortex_index_entries_per_query_bucket{le="1784059615993216724319488422043654059520000"} 26
cortex_index_entries_per_query_bucket{le="3568119231986433448638976844087308119040000"} 26
cortex_index_entries_per_query_bucket{le="7136238463972866897277953688174616238080000"} 26
cortex_index_entries_per_query_bucket{le="14272476927945733794555907376349232561600000"} 26
cortex_index_entries_per_query_bucket{le="28544953855891467589111814752698465123200000"} 26
cortex_index_entries_per_query_bucket{le="57089907711782935178223629505396930246400000"} 26
cortex_index_entries_per_query_bucket{le="114179815423565870356447259010793860492800000"} 26
cortex_index_entries_per_query_bucket{le="228359630847131740712894518021587720985600000"} 26
cortex_index_entries_per_query_bucket{le="456719261694263481425789036043175441971200000"} 26
cortex_index_entries_per_query_bucket{le="913438523388526962851578072086350883942400000"} 26
cortex_index_entries_per_query_bucket{le="182687704677705392570315614417270176784800000"} 26
cortex_index_entries_per_query_bucket{le="365375409355410785140631228834540353569600000"} 26
cortex_index_entries_per_query_bucket{le="730750818710821570281262457669080707139200000"} 26
cortex_index_entries_per_query_bucket{le="1461501637421643140562524915338161414278400000"} 26
cortex_index_entries_per_query_bucket{le="2923003274843286281125049830676322828556800000"} 26
cortex_index_entries_per_query_bucket{le="5846006549686572562250099661352645657113600000"} 26
cortex_index_entries_per_query_bucket{le="11692013099373145124500199322705291314227200000"} 26
cortex_index_entries_per_query_bucket{le="23384026198746290249000398645410582628454400000"} 26
cortex_index_entries_per_query_bucket{le="46768052397492580498000797290821165256908800000"} 26
cortex_index_entries_per_query_bucket{le="93536104794985160996001594581642330513817600000"} 26
cortex_index_entries_per_query_bucket{le="187072209589970321992003189163284661027635200000"} 26
cortex_index_entries_per_query_bucket{le="374144419179940643984006378326569322055270400000"} 26
cortex_index_entries_per_query_bucket{le="748288838359881287968012756653138644110540800000"} 26
cortex_index_entries_per_query_bucket{le="149657767671976257593602551330627728821081600000"} 26
cortex_index_entries_per_query_bucket{le="299315535343952515187205102661255457642163200000"} 26
cortex_index_entries_per_query_bucket{le="598631070687905030374410205322510915284326400000"} 26
cortex_index_entries_per_query_bucket{le="1197262141375810060748820410645021825568652800000"} 26
cortex_index_entries_per_query_bucket{le="2394524282751620121497640821290043651137305600000"} 26
cortex_index_entries_per_query_bucket{le="4789048565503240242995281642580087302274611200000"} 26
cortex_index_entries_per_query_bucket{le="957809713100648048598556328516016460449322400000"} 26
cortex_index_entries_per_query_bucket{le="1915619426201296096197112657032032920898644800000"} 26
cortex_index_entries_per_query_bucket{le="3831238852402592192394225314064065841797289600000"} 26
cortex_index_entries_per_query_bucket{le="7662477704805184384788450628128131683954579200000"} 26
cortex_index_entries_per_query_bucket{le="15324955409610368769576901256256263367859198400000"} 26
cortex_index_entries_per_query_bucket{le="30649910819220737539153802512512526735718396800000"} 26
cortex_index_entries_per_query_bucket{le="61299821638441475078307605025025553471436793600000"} 26
cortex_index_entries_per_query_bucket{le="122599643276882950156615210050051106942743587200000"} 26
cortex_index_entries_per_query_bucket{le="245199286553765900313230420100102213885487154400000"} 26
cortex_index_entries_per_query_bucket{le="490398573107531800626460840200204427770974308800000"} 26
cortex_index_entries_per_query_bucket{le="980797146215063601252921680400408855541948617600000"} 26
cortex_index_entries_per_query_bucket{le="1961594292430127202505843360800817711083897235200000"} 26
cortex_index_entries_per_query_bucket{le="3923188584860254405011686721601635422167794470400000"} 26
cortex_index_entries_per_query_bucket{le="7846377169720508810023373443203270844335988940800000"} 26
cortex_index_entries_per_query_bucket{le="15692754339441017620046746886406541688671977881600000"} 26
cortex_index_entries_per_query_bucket{le="31385508678882035240093493772813083377343955763200000"} 26
cortex_index_entries_per_query_bucket{le="62771017357764070480186987545626166754687911526400000"} 26
cortex_index_entries_per_query_bucket{le="125542034715528140960373975091252333503379223052800000"} 26
cortex_index_entries_per_query_bucket{le="251084069430556281920747950182504667006758446105600000"} 26
cortex_index_entries_per_query_bucket{le="502168138861112563841495900365009334013516892211200000"} 26
cortex_index_entries_per_query_bucket{le="1004336277722241127682991800730018668027033784422400000"} 26
cortex_index_entries_per_query_bucket{le="2008672555444482255365983601460037336054067568844800000"} 26
cortex_index_entries_per_query_bucket{le="401734511088896451073196720292007467210813513769600000"} 26
cortex_index_entries_per_query_bucket{le="803469022177792902146393440584014934421627027539200000"} 26
cortex_index_entries_per_query_bucket{le="1606938044355985804292786881168028788443254055078400000"} 26
cortex_index_entries_per_query_bucket{le="3213876088711971608585573762336057576886508110156800000"} 26
cortex_index_entries_per_query_bucket{le="6427752177423943217171147524672115153773016220313600000"} 26
cortex_index_entries_per_query_bucket{le="12855504354847886434342295049344230307546032440627200000"} 26
cortex_index_entries_per_query_bucket{le="25711008709695772868684590098688460615092064881254400000"} 26
cortex_index_entries_per_query_bucket{le="51422017419391545737369180197376921230184129762508800000"} 26
cortex_index_entries_per_query_bucket{le="102844034838783091474738360394753842460368255325177600000"} 26
cortex_index_entries_per_query_bucket{le="205688069677566182949476720789507684920736510650355200000"} 26
cortex_index_entries_per_query_bucket{le="411376139355132365898953441578515369841473021300710400000"} 26
cortex_index_entries_per_query_bucket{le="822752278710264731797906883157030739682946042601420800000"} 26
cortex_index_entries_per_query_bucket{le="1645504557420529463595813766344601473658892085202841600000"} 26
cortex_index_entries_per_query_bucket{le="3291009114841058927191627532689202947317784170405683200000"} 26
cortex_index_entries_per_query_bucket{le="6582018229682117854383255065378405894635568340811366400000"} 26
cortex_index_entries_per_query_bucket{le="1316403645936423570876651013075681178927113668162273200000"} 26
cortex_index_entries_per_query_bucket{le="2632807291872847141753302026151362357854227336324546400000"} 26
cortex_index_entries_per_query_bucket{le="5265614583745694283506604052302724715708454672649092800000"} 26
cortex_index_entries_per_query_bucket{le="1053122916749138856701320810460544943541690934529185600000"} 26
cortex_index_entries_per_query_bucket{le="2106245833498277713402641620921089887083381869058371200000"} 26
cortex_index_entries_per_query_bucket{le="4212491666996555426805283241842179774166763738116742400000"} 26
cortex_index_entries_per_query_bucket{le="8424983333993110853610566483684359548333527476233484800000"} 26
cortex_index_entries_per_query_bucket{le="16849966667986221707221132967368718966670554732469689600000"} 26
cortex_index_entries_per_query_bucket{le="33699933335972443414442265934737437933341109464939379200000"} 26
cortex_index_entries_per_query_bucket{le="6739986667194488682888453186947475586668221892987878400000"} 26
cortex_index_entries_per_query_bucket{le="13479973334388977365776906373894951173336443785775756800000"} 26
cortex_index_entries_per_query_bucket{le="26959946668777954731553812747789852346672887571551513600000"} 26
cortex_index_entries_per_query_bucket{le="53919893337555909463107625495579704693345775143073027200000"} 26
cortex_index_entries_per_query_bucket{le="107839786675111818926215250981159409386791502860146054400000"} 26
cortex_index_entries_per_query_bucket{le="21567957335022363785243050196231881877358300572029210800000"} 26
cortex_index_entries_per_query_bucket{le="43135914670044727570486100392463763754716601144058421600000"} 26
cortex_index_entries_per_query_bucket{le="86271829340089455140972200784927527509433202288116843200000"} 26
cortex_index_entries_per_query_bucket{le="172543658680178910281944401569855055018866404576233686400000"} 26
cortex_index_entries_per_query_bucket{le="345087317360357820563888803139710110037732809152467372800000"} 26
cortex_index_entries_per_query_bucket{le="690174634720715641127777606279420220075465618304934745600000"} 26
cortex_index_entries_per_query_bucket{le="1380349269441431282255552012558840440150931236609869211200000"} 26
cortex_index_entries_per_query_bucket{le="2760698538882862564511104025117680880301862673219738422400000"} 26
cortex_index_entries_per_query_bucket{le="5521397077765725129022208050235361760603725346439476844800000"} 26
cortex_index_entries_per_query_bucket{le="1104279415553145025804441610047073352120745068878995369600000"} 26
cortex_index_entries_per_query_bucket{le="2208558831106290051608883220094146704241490137779990739200000"} 26
cortex_index_entries_per_query_bucket{le="4417117662212580103217766440188293408482980275559881478400000"} 26
cortex_index_entries_per_query_bucket{le="8834235324425160206435532880376586816965960551119762956800000"} 26
cortex_index_entries_per_query_bucket{le="17668470648850320412871065760753173633931921102239525113600000"} 26
cortex_index_entries_per_query_bucket{le="35336941297700640825742131521506347267863842204478050227200000"} 26
cortex_index_entries_per_query_bucket{le="70673882595401281651484263043012694535727684408956100454400000"} 26
cortex_index_entries_per_query_bucket{le="141347765190802563302968526086025389074455368817912200908800000"} 26
cortex_index_entries_per_query_bucket{le="282695530381605126605937052172050778148910737635824401817600000"} 26
cortex_index_entries_per_query_bucket{le="565391060763210253211874104344101556297821475271648803635200000"} 26
cortex_index_entries_per_query_bucket{le="1130782121526420506423742208688203112595642
```

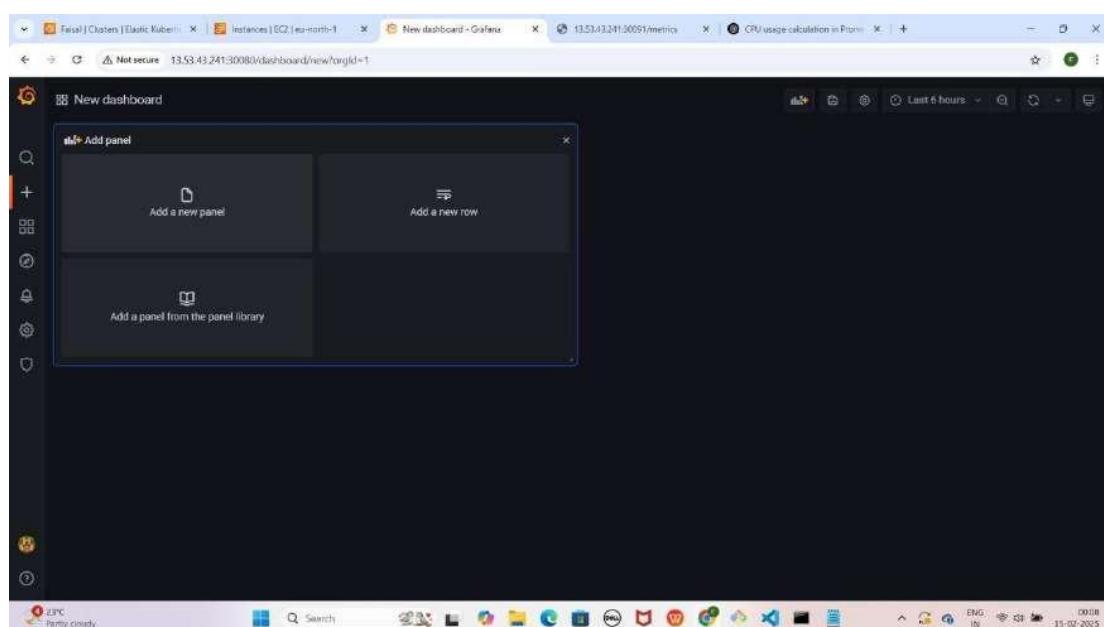
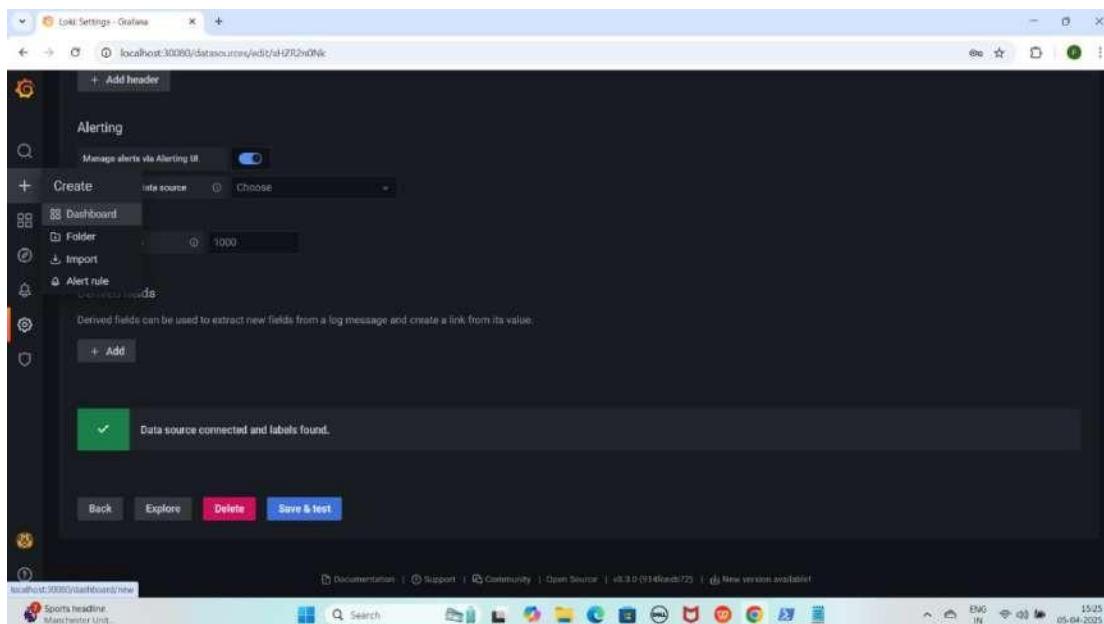


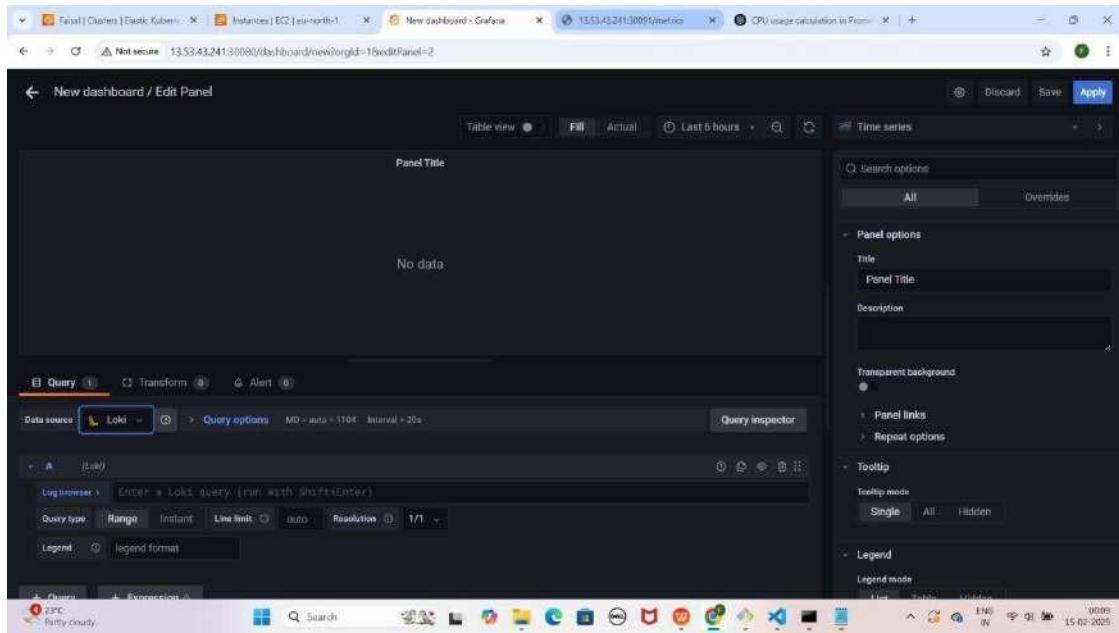


**NOTE : Save & test pe click karne ke baad aapko Data source connected and label found karke pop up aayega**

- Loki ke Loggs visualize karne ke liye "+" (plus) icon pe click karo.
- Dashboard select karo aur "Add a new panel" pe click karo.
- Query section me Data Source ko "Loki" select karo.

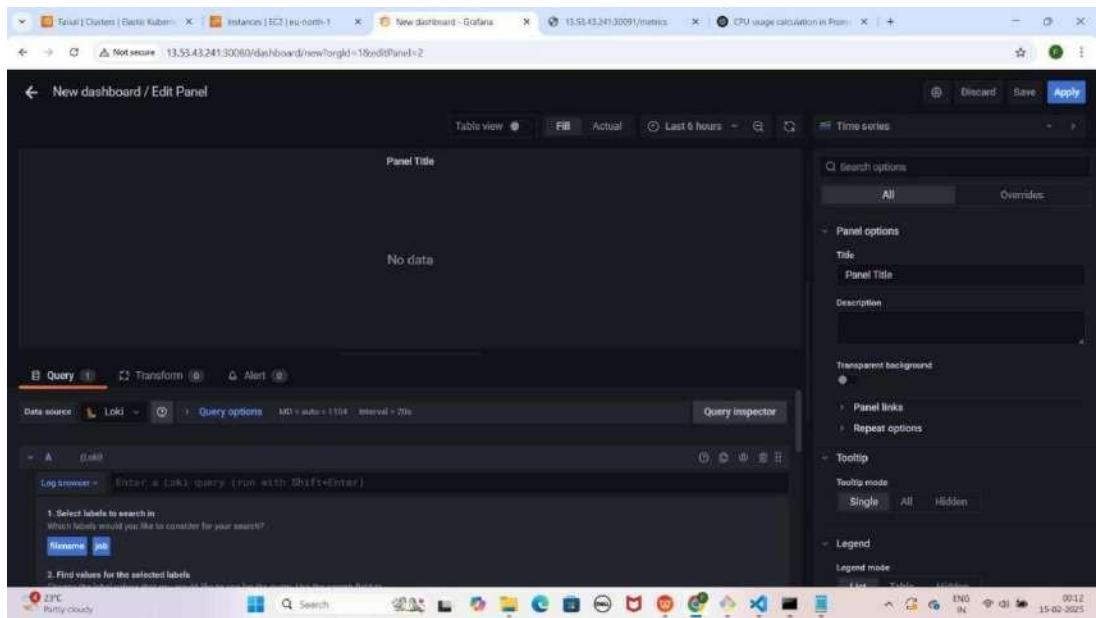
YE KUCH ISTARHA LAGEGA

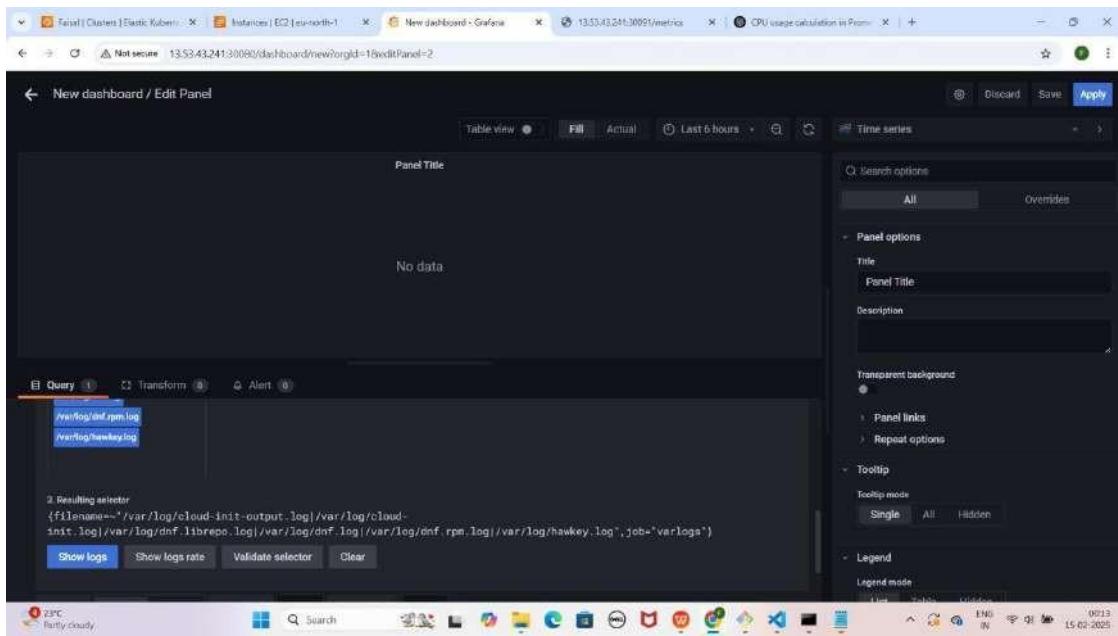
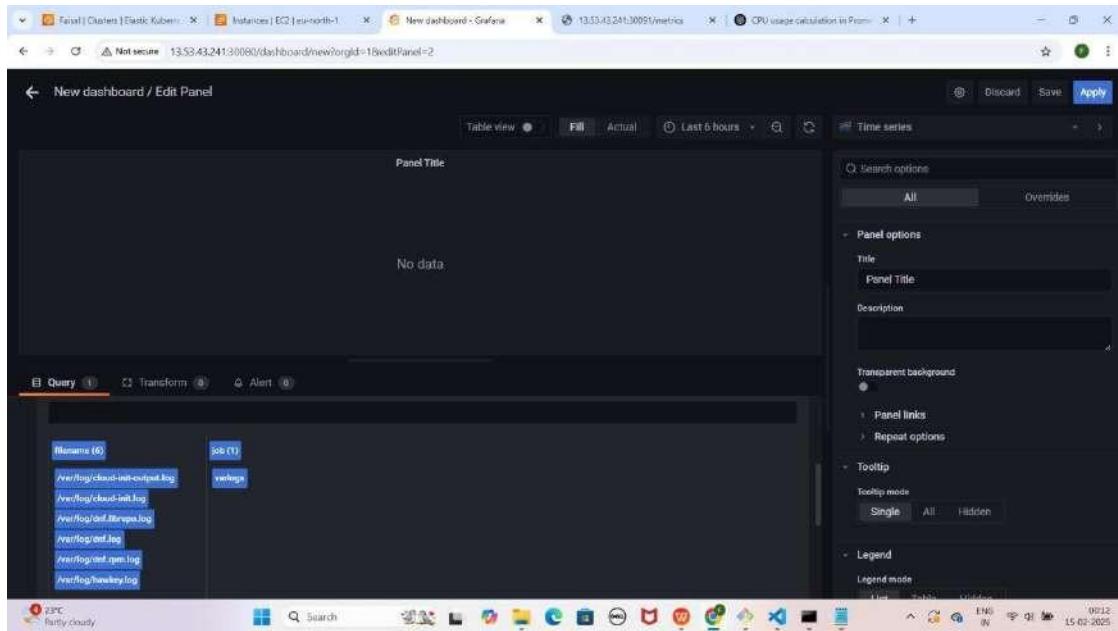




1. "Log Browser" pe click karo.
2. Pehle "filename" select karo, phir "job" (jaise ki varlogs) select karo.
3. Neeche scroll karo aur "Show logs" pe click karo.

YEKUCH ISTARHA LAGEGA





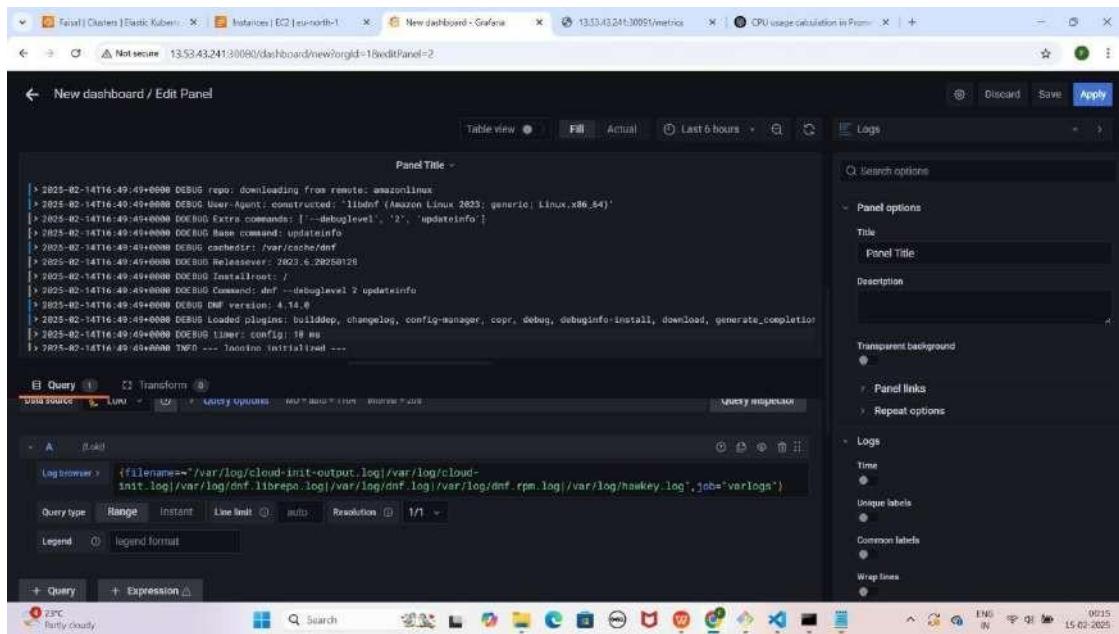
**4. Time Series pe click karo**

**5. Suggestions me jao aur "Dashboard" select karo Jaise ki maine kiya hai**

## YE KUCH ISTARHA LAGEGA

The screenshot shows the Grafana 'Edit Panel' interface. At the top, there are tabs for 'Table view' (selected), 'Fill', 'Actual', and a time range selector 'Last 6 hours'. On the right, there are 'Discard', 'Save', and 'Apply' buttons. A 'Time series' dropdown is open. Below the tabs, a message says 'Data is missing a number field'. There are two buttons: 'Switch to table' and 'Open visualization suggestions'. The main area is titled 'Panel Title' and contains a query editor. The query browser shows a log query: `(filename=~"/var/log/cloud-init-output.log")/var/log/cloud-init.log|/var/log/dnf.librepo.log|/var/log/dnf.log|/var/log/dnf.rpm.log|/var/log/hawkey.log",job="varlogs"`. The 'Query type' is set to 'Range'. On the right side, there are 'Panel options' (Title, Description, Transparent background, Panel links, Repeat options), 'Tooltip' (Tooltip mode: Single, All, Hidden), and 'Legend' (Legend mode: List, Table, Grid). The status bar at the bottom shows '23°C Partly cloudy' and a system tray with various icons.

This screenshot is nearly identical to the one above, showing the 'Edit Panel' interface in Grafana. The layout, error message 'Data is missing a number field', and the log query in the editor are all the same. The right sidebar with panel options, tooltip settings, and legend modes is also present. The status bar at the bottom shows '23°C Partly cloudy' and a system tray with various icons.



**NOTE : Is setup se hum cluster ke saare Logs Monitor kar sakte hain.**

# Part 11: Accessing Prometheus, Loki, and Grafana Using a Domain Name (Hostinger Setup)

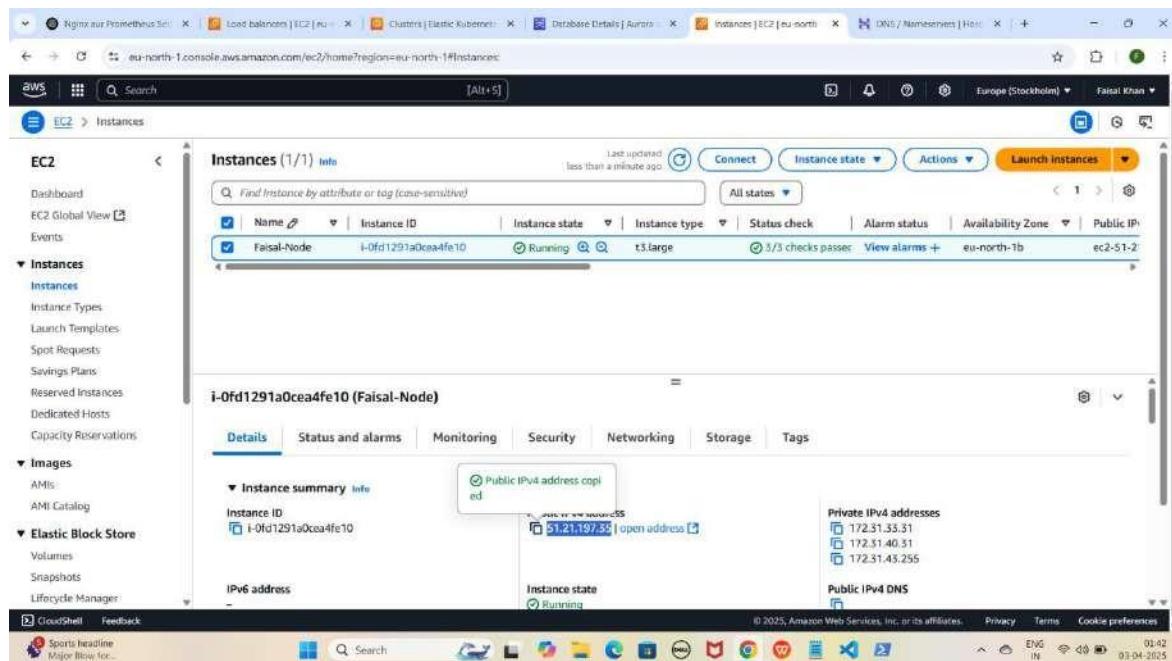
## Step 1: Hostinger me A Record Add Karo.

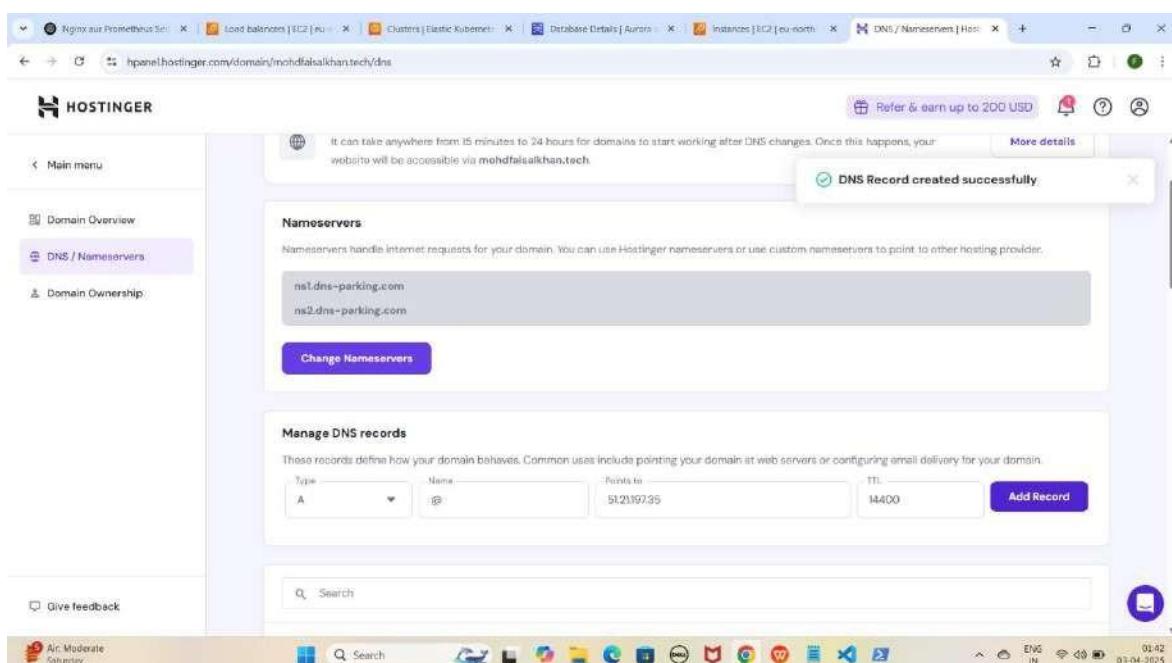
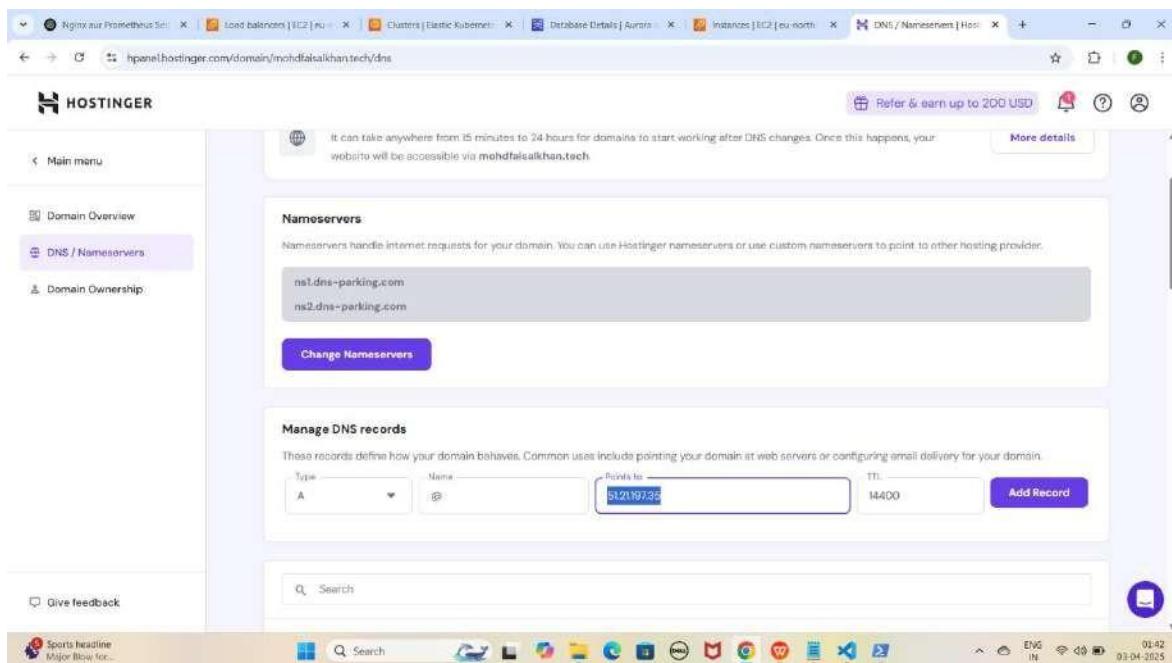
1. Phele Cluster Node ki Public IP Address Copy Karo
2. Hostinger me A Record add karne ke liye ye details fill karo

- Type: A
- Name: @
- Points to: Tumhara Cluster Node ki Public IP Address
- TTL: 14400
- Phir Add Record pe click karo.

**NOTE : Add Record pe click karne ke baad DNS Record created successfully karke Pop-up ayega**

YE KUCH ISTARHA LAGEGA





**NOTE 1:** Humne www pehle se Load Balancer ke liye map kiya hai, to monitoring apps ke liye dobara use karne par conflict error aya. Isliye monitoring ke liye sirf <http://EKS-Node-Public-IP:Port> use karein.

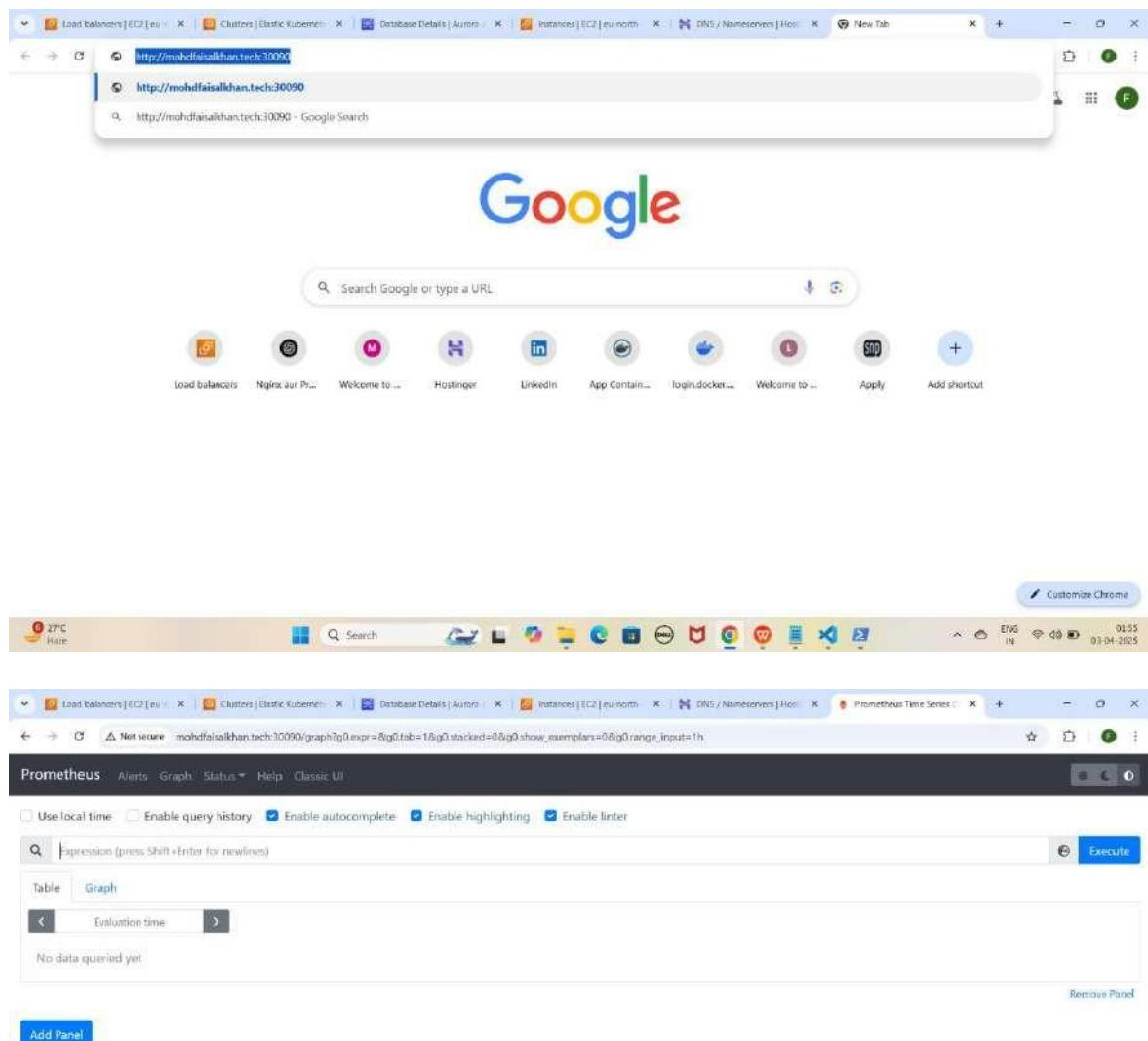
**NOTE 2:** Jab tum Cluster Node ke liye records add kar loge, toh kuch der baad apni application ko in URLs par access kar paoge. Ye process 15-20 minutes ya usse zyada bhi le sakta hai

## Step 2 : Records Add Karne Ke Baad Websites Ko Domain Name Se Access Karna

1. Apni Prometheus application ko Domain Name aur NodePort ke saath access karo.

- **Prometheus:** <http://mohdfaiskhan.tech:30090>

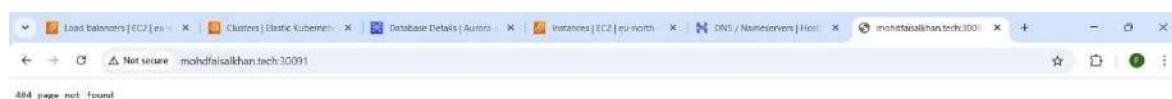
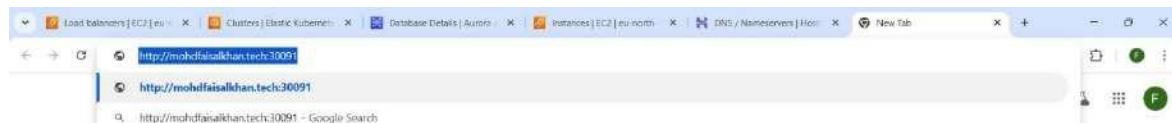
YE KUCH ISTARHA LAGEGA



## 2. Apni Loki application ko Domain Name aur NodePort ke saath access karo.

- **Loki:** <http://mohdfaiskhan.tech:30091>

YE KUCH ISTARHA LAGEGA



1. Lekin aapko **404 page not found** dikhayga to **/ready** se check kariye loki ko jaise ki mere case kuch asisa hogा.

<http://mohdfaiskhan.tech:30091/ready>

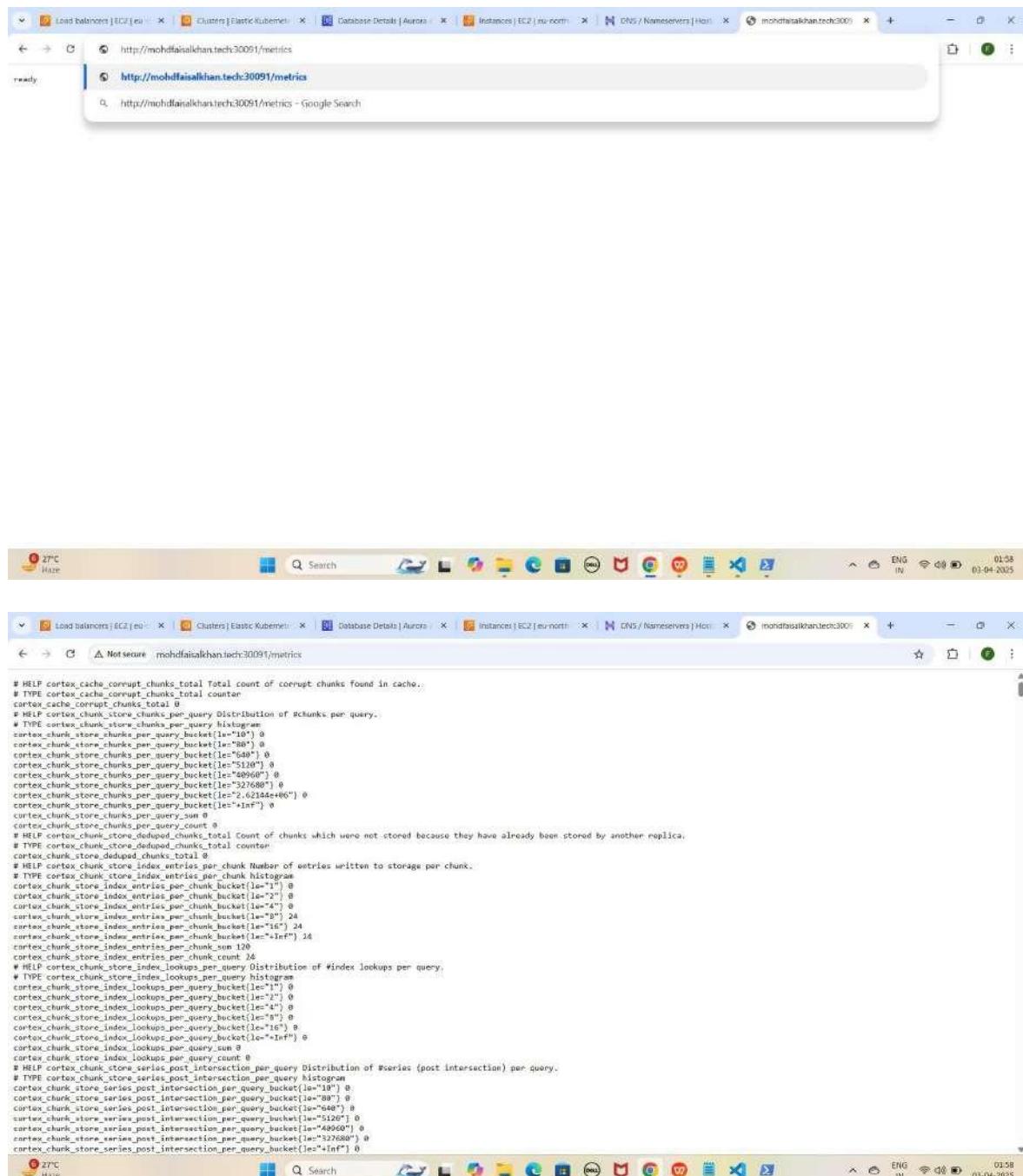
YE KUCH ISTARHA LAGEGA



**2. Agar aapko ready show karraha hai to iska matlab loki successfully run horaha hai ab loki metrics loggs collect karraha hai check karne ke liye/metrics run kariye jaise ki mere case me kuch aisa hogा.**

<http://mohdfaikhan.tech:30091/metrics>

YE KUCH ISTARHA LAGEGA

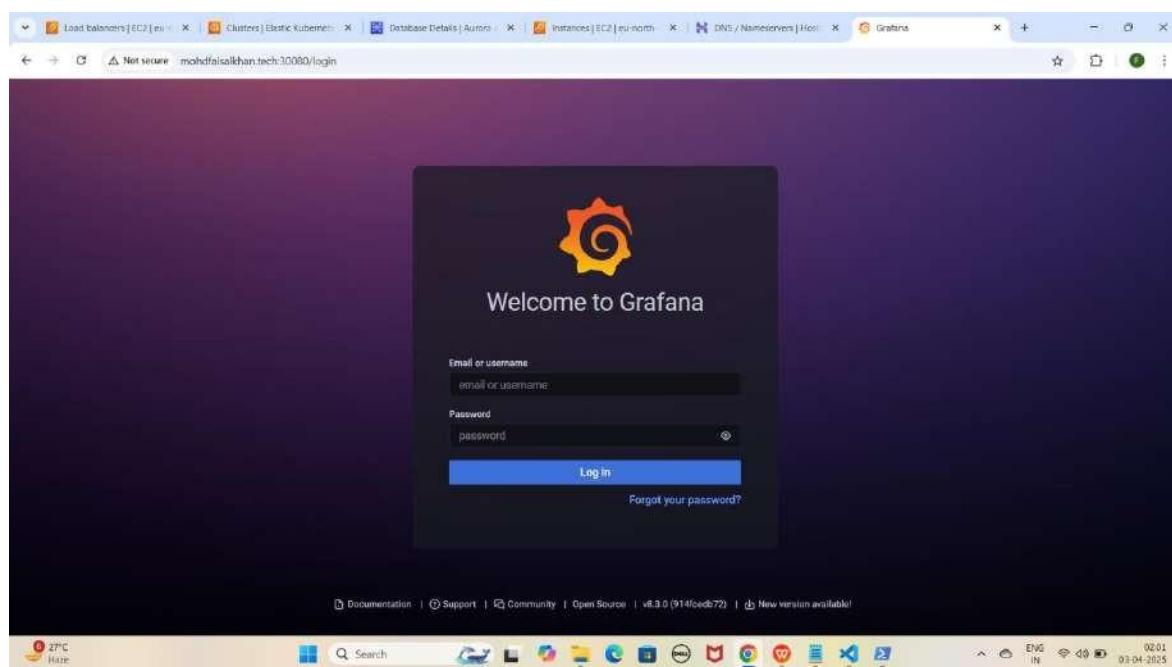


```
# HELP cortex_cache_corrupt_chunks_total Total count of corrupt chunks found in cache.
# TYPE cortex_cache_corrupt_chunks_total counter
cortex_cache_corrupt_chunks_total 0
# HELP cortex_chunk_store_chunks_per_query Distribution of #chunks per query.
# TYPE cortex_chunk_store_chunks_per_query histogram
cortex_chunk_store_chunks_per_query_bucket{le="1e-10"} 0
cortex_chunk_store_chunks_per_query_bucket{le="80"} 0
cortex_chunk_store_chunks_per_query_bucket{le="640"} 0
cortex_chunk_store_chunks_per_query_bucket{le="5120"} 0
cortex_chunk_store_chunks_per_query_bucket{le="40960"} 0
cortex_chunk_store_chunks_per_query_bucket{le="327680"} 0
cortex_chunk_store_chunks_per_query_bucket{le="2621440+640"} 0
cortex_chunk_store_chunks_per_query_bucket{le="+Inf"} 0
cortex_chunk_store_chunks_per_query_sum 0
cortex_chunk_store_deduped_chunks_total 0
# HELP cortex_chunk_store_deduped_chunks_total Count of chunks which were not stored because they have already been stored by another replica.
# TYPE cortex_chunk_store_deduped_chunks_total counter
cortex_chunk_store_deduped_chunks_total 0
# HELP cortex_chunk_store_index_entries_per_chunk Number of entries written to storage per chunk.
# TYPE cortex_chunk_store_index_entries_per_chunk histogram
cortex_chunk_store_index_entries_per_chunk_bucket{le="1"} 0
cortex_chunk_store_index_entries_per_chunk_bucket{le="2"} 0
cortex_chunk_store_index_entries_per_chunk_bucket{le="4"} 0
cortex_chunk_store_index_entries_per_chunk_bucket{le="8"} 24
cortex_chunk_store_index_entries_per_chunk_bucket{le="16"} 24
cortex_chunk_store_index_entries_per_chunk_bucket{le="+Inf"} 14
cortex_chunk_store_index_entries_per_chunk_sum 126
# HELP cortex_chunk_store_index_lockups_per_query Distribution of #index lockups per query.
# TYPE cortex_chunk_store_index_lockups_per_query histogram
cortex_chunk_store_index_lockups_per_query_bucket{le="1"} 0
cortex_chunk_store_index_lockups_per_query_bucket{le="2"} 0
cortex_chunk_store_index_lockups_per_query_bucket{le="4"} 0
cortex_chunk_store_index_lockups_per_query_bucket{le="8"} 0
cortex_chunk_store_index_lockups_per_query_bucket{le="16"} 0
cortex_chunk_store_index_lockups_per_query_bucket{le="+Inf"} 0
cortex_chunk_store_index_lockups_per_query_sum 0
# HELP cortex_chunk_store_series_post_intersection_per_query Distribution of #series (post intersection) per query.
# TYPE cortex_chunk_store_series_post_intersection_per_query histogram
cortex_chunk_store_series_post_intersection_per_query_bucket{le="10"} 0
cortex_chunk_store_series_post_intersection_per_query_bucket{le="80"} 0
cortex_chunk_store_series_post_intersection_per_query_bucket{le="640"} 0
cortex_chunk_store_series_post_intersection_per_query_bucket{le="5120"} 0
cortex_chunk_store_series_post_intersection_per_query_bucket{le="40960"} 0
cortex_chunk_store_series_post_intersection_per_query_bucket{le="327680"} 0
cortex_chunk_store_series_post_intersection_per_query_bucket{le="2621440+640"} 0
cortex_chunk_store_series_post_intersection_per_query_bucket{le="+Inf"} 0
```

### 3. Apni Grafana application ko Domain Name aur NodePort ke saath access karo.

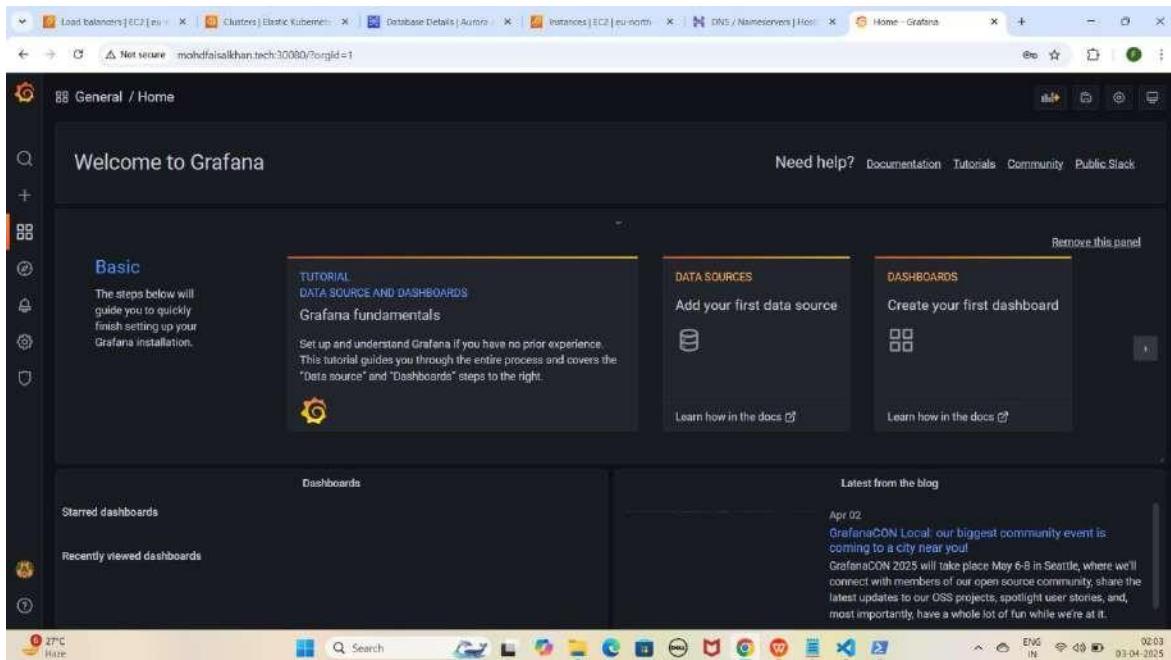
- **Grafana:** <http://mohdfaiskhan.tech:30080>

YE KUCH ISTARHA LAGEGA





**NOTE : Ye wahi password hai jo humne first time Grafana ki Public IP ko NodePort ke saath access karte waqt use kiya tha.**



**Project Video Demo: <https://youtu.be/tcYUQcHCkV4>**

## **\*\*\*\*\*Commands Used in This Project\*\*\*\*\***

### **Part 1: EKS Cluster Creation and Setup**

#### **1. Configure AWS CLI**

```
aws configure
```

#### **2. Enter the following details**

AWS Access Key ID: XXXXXXXXXXXXXXXXXX

AWS Secret Access Key: XXXXXXXXXXXXXXXXXXXXXXXXX

Default Region: eu-north-1

Output Format: text

#### **3. Update kubeconfig for your EKS cluster**

```
aws eks update-kubeconfig --name Faisal --region eu-north-1
```

#### **4. Verify the EKS node status**

```
kubectl get nodes
```

## **Part 2: Nginx pods and Service Deployment**

### **1. Deploy NGINX with Replicas**

```
kubectl apply -f nginx-deployment.yaml
```

### **2. Expose NGINX using Load Balancer Service**

```
kubectl apply -f nginx-loadbalancer-service.yaml
```

### **3. Verify Deployments & Services**

```
kubectl get pods
```

```
kubectl get services
```

## **Part 5: Enabling HTTPS on Load Balancer Using ACM**

### **1. Apply Ingress for Routing**

```
kubectl apply -f ingress.yaml
```

### **2. Verify Ingress Rules**

```
kubectl get ingress
```

## **Part 8: Accessing Amazon RDS (MySQL, Oracle, PostgreSQL) from an EKS Cluster (With Endpoint)**

### **Method 1: Application ke through Database Access Karna (With Endpoint)**

#### **1. Initialize Node.js Application & Install Dependencies**

```
sudo npm init -y
```

```
sudo npm install express mysql pg oracledb
```

#### **2. Build & Push Docker Image**

```
sudo docker build -t faisalkhan35/backend-app .
```

```
sudo docker images
```

```
sudo docker login
```

```
sudo docker push faisalkhan35/backend-app
```

#### **3. Generate Base64 Encodings for Secrets**

```
echo -n FaisalKhan | base64
```

```
echo -n Faisalkhan35$ | base64
```

```
echo -n 'RmFpc2FsS2hhbg==' | base64 --decode
```

```
echo -n 'RmFpc2Fsa2hhbjM1JA==' | base64 --decode
```

#### **4. Apply Secrets, ConfigMaps & RBAC**

```
kubectl apply -f rds-secrets.yaml
```

```
kubectl apply -f rds-configmap.yaml
```

```
kubectl apply -f service-account.yaml
```

```
kubectl apply -f role.yaml
```

```
kubectl apply -f rolebinding.yaml
```

#### **5. Deploy Backend Application Pod**

```
kubectl apply -f backend-app-pod.yaml
```

#### **6. Verify Pod Deployment**

```
kubectl get pods
```

#### **7. Access Backend Application Pod**

```
kubectl exec -it backend-app-pod -- /bin/bash
```

## **MySQL Database Access**

### **1. Login to MySQL**

```
mysql -h faisal-mysql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -  
u FaisalKhan -p
```

### **2. Create & Verify Database**

```
CREATE DATABASE faisal_db;  
  
SHOW DATABASES;
```

## **Oracle Database Access**

### **1. Login to Oracle**

```
sqlplus FaisaLKhan@//faisal-oracle-db.clis8ywwg5wd.eu-north-  
1.rds.amazonaws.com:1521/ORCL
```

### **2. Create & Verify User**

```
CREATE USER "faisal_db" IDENTIFIED BY Faisalkhan35$;  
  
SELECT username FROM all_users WHERE username = 'faisal_db';
```

## **PostgreSQL Database Access**

### **1. Login to PostgreSQL**

```
psql -h faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -U FaisalKhan -d postgres
```

### **2. Create & Verify Database**

```
CREATE DATABASE faisal_db;
```

```
SELECT datname FROM pg_database;
```

## **Method 2: Client Pods ke through Database Access Karna (With Endpoint)**

### **1. Deploy Client Pods for MySQL, Oracle, and PostgreSQL**

```
kubectl apply -f mysql-client.yaml
```

```
kubectl apply -f oracle-client.yaml
```

```
kubectl apply -f postgresql-client.yaml
```

### **2. Verify Pods**

```
kubectl get pods
```

## **MySQL Database Access**

### **1. Access MySQL Client Pod**

```
kubectl exec -it mysql-client -- /bin/bash
```

### **2. Login to MySQL**

```
mysql -h faisal-mysql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -  
u FaisalKhan -p
```

### **3. Verify Databases**

```
SHOW DATABASES;
```

## **Oracle Database Access**

### **1. Access Oracle Client Pod**

```
kubectl exec -it oracle-client -- /bin/bash
```

### **2. Login to Oracle**

```
sqlplus FaisalKhan@//faisal-oracle-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com:1521/ORCL
```

### **3. Verify User**

```
SELECT username FROM all_users WHERE username = 'faisal_db';
```

## **PostgreSQL Database Access**

### **1. Access PostgreSQL Client Pod**

```
kubectl exec -it postgresql-client -- /bin/bash
```

### **2. Login to PostgreSQL**

```
psql -h faisal-postgresql-db.clis8ywwg5wd.eu-north-1.rds.amazonaws.com -U FaisalKhan -d postgres
```

### **3. Verify Databases**

```
SELECT datname FROM pg_database;
```

## **Part 9: Accessing Amazon RDS (MySQL, Oracle, PostgreSQL) from an EKS Cluster (Without Hardcoded Endpoint)**

### **Method 1: Application ke through Database Access Karna (Without Hardcoded Endpoint)**

#### **1. Apply External Name Service**

```
kubectl apply -f external-name.yaml
```

#### **2. Verify Service**

```
kubectl get services
```

#### **3. Access Backend Application Pod**

```
kubectl exec -it backend-app-pod -- /bin/bash
```

### **MySQL Database Access Without Hardcoded Endpoint**

#### **1. Login to MySQL**

```
mysql -h faisal-mysql -u FaisalKhan -p
```

#### **2. Verify Databases**

```
SHOW DATABASES;
```

## **Oracle Database Access Without Hardcoded Endpoint**

### **1. Login to Oracle**

```
sqlplus FaisaLKhan@//faisal-oracle:1521/ORCL
```

### **2. Verify User**

```
SELECT username FROM all_users WHERE username = 'faisal_db';
```

## **PostgreSQL Database Access Without Hardcoded Endpoint**

### **1. Login to PostgreSQL**

```
psql -h faisal-postgresql -U FaisaLKhan -d postgres
```

### **2. Verify Databases**

```
SELECT datname FROM pg_database;
```

## **Method 2: Client Pods ke through Database Access Karna (Without Hardcoded Endpoint)**

### **MySQL Database Access Without Hardcoded Endpoint**

#### **1. Access MySQL Client Pod**

```
kubectl exec -it mysql-client -- /bin/bash
```

#### **2. Login to MySQL**

```
mysql -h faisal-mysql -u FaisalKhan -p
```

#### **3. Verify Databases**

```
SHOW DATABASES;
```

### **Oracle Database Access Without Hardcoded Endpoint**

#### **1. Access Oracle Client Pod**

```
kubectl exec -it oracle-client -- /bin/bash
```

#### **2. Login to Oracle**

```
sqlplus FaisaLKhan@//faisal-oracle:1521/ORCL
```

#### **3. Verify User**

```
SELECT username FROM all_users WHERE username = 'faisal_db';
```

## **PostgreSQL Database Access Without Hardcoded Endpoint**

### **1. Access PostgreSQL Client Pod**

```
kubectl exec -it postgresql-client -- /bin/bash
```

### **2. Login to PostgreSQL**

```
psql -h faisal-postgresql -U FaisalKhan -d postgres
```

### **3. Verify Databases**

```
SELECT datname FROM pg_database;
```

## **Part 10: Monitoring Using Prometheus and Loki With Grafana**

### **1. Create Monitoring Namespace**

```
kubectl create namespace monitoring
```

### **2. Deploy Prometheus Components**

```
kubectl apply -f prometheus-daemonset.yaml
```

```
kubectl apply -f prometheus-rbac.yaml
```

```
kubectl apply -f prometheus-nodeport-service.yaml
```

### **3. Verify Prometheus Pods & Services**

```
kubectl get pods -n monitoring
```

```
kubectl get services -n monitoring
```

### **4. Important Prometheus Metrics**

```
rate(process_cpu_seconds_total[30s])
```

```
process_resident_memory_bytes
```

```
process_network_transmit_bytes_total
```

```
process_network_receive_bytes_total
```

## **5. Deploy Loki & Promtail**

```
kubectl apply -f promtail-daemonset.yaml
```

```
kubectl apply -f loki-daemonset.yaml
```

```
kubectl apply -f loki-nodeport-service.yaml
```

## **6. Verify Loki & Promtail Pods & Services**

```
kubectl get pods -n monitoring
```

```
kubectl get services -n monitoring
```

## **7. Deploy Grafana**

```
kubectl apply -f grafana-deployment.yaml
```

## **8. Verify Grafana Pods & Services**

```
kubectl get pods -n monitoring
```

```
kubectl get services -n monitoring
```

