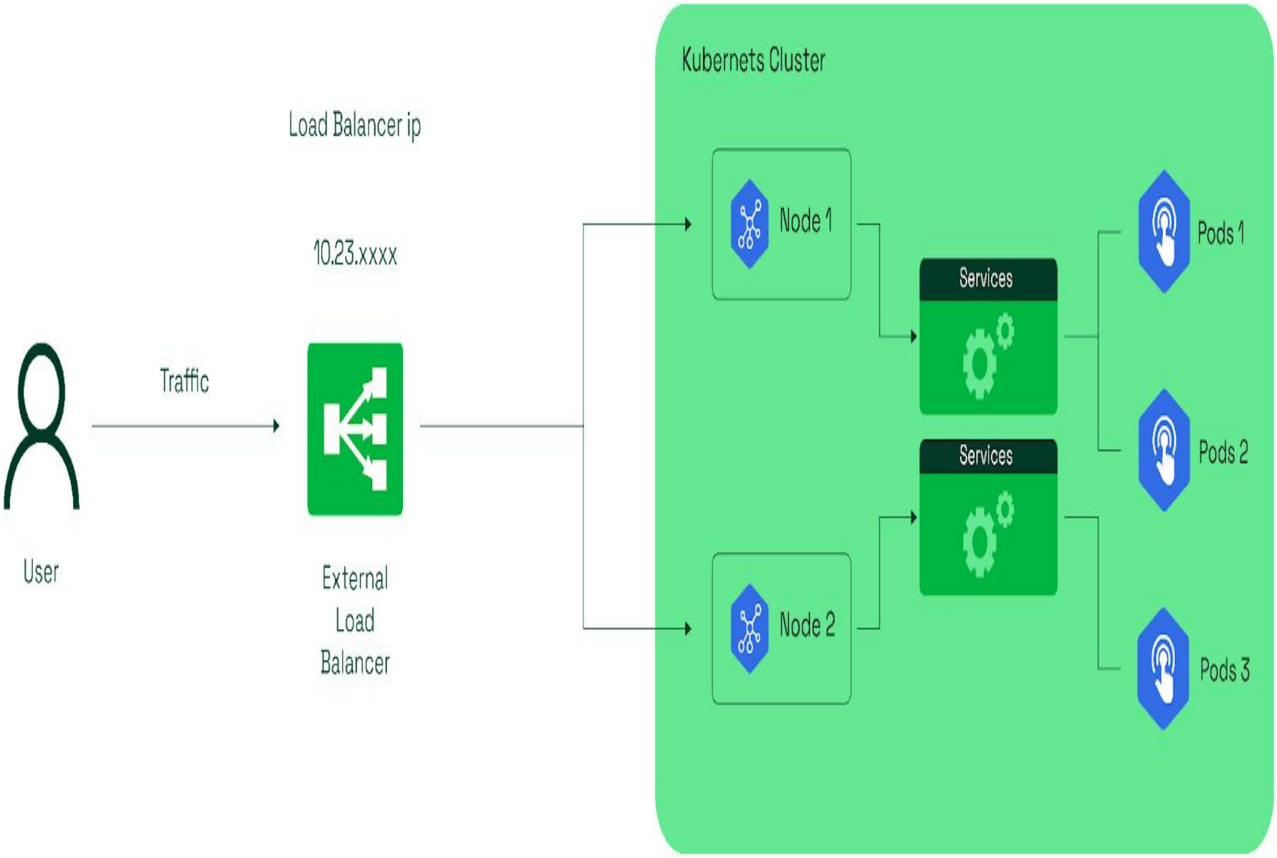
KUBERNETES PROJECT WITH LOAD BALANCER SERVICE



Part 1: EKS Cluster Creation and Setup

Step 1: AWS Login aur EKS Cluster Create Karna

1. AWS Management Console pe login karein.
2. EKS service ko search kar ke “Create Cluster” option pe click karein.
3. Cluster ka naam, region aur configuration choose karein.

Step 2: Networking Configuration

1. VPC, subnets aur security groups ko setup karein.
2. Networking settings ka dhyan rakhein jaise ki Public aur Private subnets.

Step 3: Observability Configure Karna

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

Step 4: Add-ons Configure Karna

1. EKS cluster ke liye add-ons jaise ki CoreDNS aur VPC CNI configure karein.
2. Pod Identity IAM Role for Service Account Configure karo.

Step 5: Cluster Create Karna

1. Sab settings review karein aur "Create Cluster" par click karein.
2. Cluster ke Active hone ka wait karein jo kuch minutes le sakta hai.

Step 6: Node Group Create Karna

1. "Add Node Group" option select karein.
2. Node group ka naam define karein aur IAM role assign karein.
3. Instance types ka selection karein jo workload ke liye best ho.

Step 7: Compute aur Scaling Configuration

1. Desired, Minimum, aur Maximum node count set karein taake autoscaling sahi kaam kare.
2. Spot Instances ya On-Demand Instances ka selection karein based on cost aur availability.

Step 8: Node Group ki Networking Configure Karna

1. Node Group ke liye subnets, security groups, aur VPC configuration set karein.

Step 9: Node Group Review and Create Karna

1. Node group ke configuration ko review karein aur create kar dein.
2. "Create Node Group" par click karein aur nodes ka active hone ka wait karein.

Step 10: Security Groups Configure Karna EKS Cluster Ke liye

1. Cluster ke security group mein required inbound aur outbound rules configure karein.
2. API server aur node communication ke liye proper ports open karein.

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

1. Nodes ke security group mein required inbound aur outbound rules configure karein.
2. SSH access agar required ho toh specific IPs ke liye allow karein.

Step 12: Project Ki GitHub Repository Clone Karein

1. GitHub repository ko clone karein.
2. Code aur configurations ko local machine pe setup karein.

Step 13: AWS CLI Configure Karo

1. AWS CLI ko configure karein with access keys.
2. aws configure command se credentials aur region set karein.

Part 2: Nginx Pods and Service Deployment

Step 1: nginx-deployment.yaml File Ka Kaam

1. nginx-deployment.yaml file ko configure karein taake pods create ho sakein.
2. Nginx application ki specifications set karein.

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

1. Load Balancer service ko configure karein jo Nginx pods ko expose kare.
2. Load Balancer ko configure karke service ko deploy karein.

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

Step 1: Domain Name Provider Choose Karo

1. Hostinger ko domain provider ke liye choose karein.

Step 2: Hostinger Pe Domain Search Karo

1. Hostinger ki website pe jaake domain name search karein.

Step 3: Domain Select Karo

1. Domain name ko select karein jo aapko pasand aaye.

Step 4: Account Create Karo

1. Hostinger pe account create karein.

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

1. Payment complete karein aur domain ko apne account mein register karein.

Step 6: Hostinger Me Domains Section Me Navigate Karo

1. Hostinger ke dashboard pe domains section me navigate karein.

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

1. DNS/Nameservers section me jaake configurations set karein.

Part 4: Configuring DNS on Hostinger for Load Balancer

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

1. Hostinger DNS settings mein CNAME record add karein jo Load Balancer se connect ho.

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

1. WWW subdomain ke liye bhi CNAME record add karein.

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

1. DNS records add karne ke baad website ko domain name se access karein.

Part 5: Enabling HTTPS on Load Balancer Using ACM

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

1. ACM ke liye CAA records ko add karein.

Step 2: ACM se SSL Certificate Request Karo

1. AWS ACM (AWS Certificate Manager) se SSL certificate request karein.

Step 3: ACM CNAME Records Hostinger Me Add Karo

1. Hostinger mein ACM ke CNAME records ko add karein.

Step 4: ingress.yaml File Ka Kaam

1. Ingress YAML file ko configure karein taake HTTPS enabled ho.

Part 6: Load Balancer Settings and Updates

Step 1: Load Balancer ki Security Settings Update Karo

1. Load Balancer ki security settings ko update karein.

Step 2: Load Balancer me Listeners Update Karo

1. HTTP aur HTTPS listeners ko update karein.

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

1. Website ko HTTPS domain se access karne ke baad verify karein.

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

Step 1: Subnet Group Create Karna

1. RDS ke liye subnet group create karein.

Step 2: MySQL Database Create Karna

1. MySQL database ko create karein.

Step 3: Oracle Database Create Karna

1. Oracle database ko create karein.

Step 4: PostgreSQL Database Create Karna

1. PostgreSQL database ko create karein.

Step 5: Database Status Check Karna

1. RDS database ka status check karein taake sab kuch sahi ho.

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

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

Step 1: Dependencies Install Karo

1. Required Dependencies install karo Database Jaise MySQL, Oracle, PostgreSQL npm ka use karke

Step 2: Application Image ki Files aur Tools

1. Dockerfile, config files, and application code ready rakho.
2. Tools: Docker, Git, VS Code, AWS CLI.

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

1. RDS MySQL endpoint ko use karke CLI ya application se connect karo.
2. mysql -h <endpoint> -u <username> -p

Step 4: Amazon RDS (Oracle) Database Access Karo - Method 1

1. Oracle SQL Developer ya CLI se connect karo.
2. sqlplus <username>/<password>@<endpoint>:<port>/<service_name>

Step 5: Amazon RDS (PostgreSQL) Database Access Karo - Method 1

1. psql tool ya app se RDS PostgreSQL connect karo.
2. psql -h <endpoint> -U <username> -d <database_name>

Method 2: Client Pods ke through Database Access Karna (With Endpoint)

Step 1: Client Pods ki YAML Files ki Location par Navigate Karo

1. Terminal mein us directory mein jao jahan YAML files stored hain.
2. Example: cd ~/Multi-DB-CLI-Client-Method/

Step 2: Client Pods Deploy Karo

1. Kubernetes mein pods deploy karo
2. kubectl apply -f mysql-client.yaml
3. kubectl apply -f oracle-client.yaml
4. kubectl apply -f postgresql-client.yaml

Step 3: Amazon RDS (MySQL) Database Access Karo - Method 2

1. MySQL client pod ke andar jao
2. kubectl exec -it mysql-client -- bash
3. RDS MySQL ko connect karo
4. mysql -h <endpoint> -u <username> -p

Step 4: Amazon RDS (Oracle) Database Access Karo - Method 2

1. Oracle client pod ke andar jao
2. kubectl exec -it oracle-client -- bash
3. RDS Oracle ko connect karo
4. sqlplus <username>/<password>@<endpoint>:<port>/<service_name>

Step 5: Amazon RDS (PostgreSQL) Database Access Karo - Method 2

1. PostgreSQL client pod ke andar jao
2. kubectl exec -it postgres-client -- bash
3. RDS PostgreSQL ko connect karo
4. psql -h <endpoint> -U <username> -d <database_name>

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

1. Is file se Kubernetes mein ek ExternalName Service create hoti hai.
2. Ye service DNS ke through AWS RDS endpoint ko forward karti hai.

Step 2: Amazon RDS (MySQL) Database Access Karo - Method 1

1. Application pod se MySQL access karo via ExternalName
2. mysql -h mysql-db -u <username> -p

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

1. Application pod se Oracle access karo via ExternalName
2. sqlplus <username>/<password>@oracle-db:<port>/<service_name>

Step 4: Amazon RDS (PostgreSQL) Database Access Karo - Method 1

1. Application pod se PostgreSQL access karo via ExternalName
2. psql -h postgres-db -U <username> -d <database_name>

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 ke andar jao
2. kubectl exec -it mysql-client -- bash
3. MySQL Client Pod se MySQL access karo via ExternalName
4. mysql -h mysql-db -u <username> -p

Step 2: Amazon RDS (Oracle) Database Access Karo-Method 2

1. Oracle client pod ke andar jao
2. kubectl exec -it oracle-client -- bash
3. Oracle Client Pod se Oracle access karo via ExternalName
4. sqlplus <username>/<password>@oracle-db:<port>/<service_name>

Step 3: Amazon RDS (PostgreSQL) Database Access Karo - Method 2

1. PostgreSQL client pod ke andar jao
2. kubectl exec -it postgres-client -- bash
3. PostgreSQL Client Pod se PostgreSQL access karo via ExternalName
4. psql -h postgres-db -U <username> -d <database_name>

Part 10: Monitoring Using Prometheus and Loki With Grafana

Step 1: prometheus-daemonset.yaml File Ka Kaam

1. Prometheus ko deploy karne ke liye daemonset.yaml file configure karein.

Step 2: prometheus-rbac.yaml File Ka Kaam

1. Prometheus ke liye RBAC roles configure karein.

Step 3: prometheus-nodeport-service.yaml File Ka Kaam

1. Prometheus ke liye service ko configure karein.

Step 4: promtail-daemonset.yaml File Ka Kaam

1. Logs ko collect karne ke liye promtail daemonset file ko configure karein.

Step 5: loki-daemonset.yaml File Ka Kaam

1. Loki daemonset file ko configure karein.

Step 6: loki-nodeport-service.yaml File Ka Kaam

1. Loki ke liye service ko configure karein.

Step 7: grafana-deployment.yaml File Ka Kaam

1. Grafana ko deploy karne ke liye grafana-deployment.yaml file configure karein.

Step 8: Grafana Mein Prometheus Data Source Add Karna, Queries Add Karna Aur Dashboard Create Karna

1. Grafana mein Prometheus data source add karein, queries configure karein aur dashboard create karein.

Step 9: Grafana Mein Loki Data Source Add Karna Aur Dashboard Create Karna

1. Grafana mein Loki data source add karein aur dashboard configure karein.

Part 11: Accessing Prometheus, Loki, and Grafana Using a Domain Name (Hostinger Setup)

Step 1: Hostinger me A Record Add Karo

1. Hostinger mein A Record add karein jo Prometheus, Loki, aur Grafana ko point kare.

Step 2: Records Add Karne Ke Baad Websites Ko Domain Name Se Access Karna

1. DNS records update karne ke baad websites ko Domain Name se access karein.