Kubernetes Project using EKS:

Pre-Requirements:

```
Install the Aws-Cli, Eksctl, Kubectl in the os.
```

Aws-cli Installation:

```
Install AWS CLI on Linux

* Step 1: Download the AWC CLI installation files using curl.

          curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
"awscliv2.zip"
* Step 2:
          unzip awscliv2.zip
 Step 3:
        sudo ./aws/install
```

or use below command:

```
 curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
    unzip awscli-bundle.zip
   ./awscli-bundle/install -b ~/bin/aws
```

Installation of eks:

```
   * sudo apt update
   * curl --silent --location
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -
s)_amd64.tar.gz" | tar xz -C /tmp
   * sudo mv /tmp/eksctl /usr/local/bin
   * eksctl version
```

Installation of Kubectl on Ubuntu:

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

* sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

* chmod +x kubectl

* mkdir -p ~/.local/bin

* mv ./kubectl ~/.local/bin/kubectl
```

Create a Cluster in Aws

```
* go to aws console and create a cluster before that create a role anyone in the
below mentioned.
* 1. AWS Service: eks2.
```

* 2. eks-cluster-role .

Connect eks to kubectl:

* aws eks update-kubeconfig --name safexpay-new --region ap-south-1
* aws configure:
* AWS Access Key ID [None]: AKIAWRMAGMDWCO3JX3FK
* AWS Secret Access Key [None]: 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
* Default region name [None]: ap-south-1
* Default output format [None]:

Creating the ingress controller-nlb:

 * kubectl create serviceaccount nginx-ingress-controller
 * kubectl create clusterrolebinding nginx-ingress-controller --clusterrole=cluster-admin --serviceaccount=default:nginx-ingress-controller
 * kubectl create namespace ingress-nginx
 * mkdir nginx-ingress-controller
 * helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
 * helm repo update
 * helm install nginx-ingress ingress-nginx/ingress-nginx --namespace ingress-nginx --set controller.service.annotations."service\.beta\.kubernetes\.io/aws-load-balancer-type"=classic

Nginx-ingress-controller.yaml file

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: backend-ingress
  namespace: ingress-nginx
  annotations:
    nginx.ingress.kubernetes.io/ssl-redirect: "false"
    kubernetes.io/ingress.class: 'nginx'
    nginx.ingress.kubernetes.io/use-regex: "true"

spec:
  rules:
    - http:
        paths:
          - path: /ms-auth-server-1-0
            pathType: Prefix
            backend:
              service:
                name: auth-service
                port:
                  number: 8201
          - path: /ms-otp-service-1-0
            pathType: Prefix
            backend:
              service:
                name: otp-service
                port:
                  number: 8102
          - path: /core-service-1-0
```

```
      pathType: Prefix
      backend:
        service:
          name: core-service
          port:
            number: 9099
    - path: /ms-portal-1-0
      pathType: Prefix
    backend:
        service:
          name: portal-service
          port:
            number: 8008
    - path: /dao-service-1-0
      pathType: Prefix
      backend:
        service:
          name: dao-service
          port:
            number: 8100
    - path: /cache-service-1-0
      pathType: Prefix
      backend:
        service:
          name: cache-service
          port:
            number: 9098
    - path: /kafka-service-1-0
      pathType: Prefix
      backend:
        service:
          name: kafka-service
          port:
            number: 8105
    - path: /pg-cron-1-0
      pathType: Prefix
      backend:
        service:
          name: pg-cron
          port:
            number: 8104
     - path: /kafka-service(/|$)(.*)
      pathType: ImplementationSpecific
      backend:
        service:
          name: kafka-service
          port:
             number: 9092
```

Then use backend and frontend deployment files where you can see in the name of

Backend_Deployment_files_yamls

Frontend_deployment_files