Setup Application on AWS Elastic Kuberentes Service

Prerequisites:

- An AWS account with appropriate permissions.
- AWS CLI installed and configured with your credentials.
- kubectl (Kubernetes CLI) installed.
- Docker or a containerization tool to build your application's container image.

Create an EKS Cluster:

Use the AWS Management Console, AWS CLI, or CloudFormation to create an EKS cluster.

```
aws eks create-cluster --name my-cluster --role-arn arn:aws:iam::123456789012:role/eks-cluster-role --resources-vpc-config subnetIds=subnet-0abcd1234,subnet-0efgh5678
```

Configure kubectl:

Configure kubectl to interact with the EKS cluster you created.

```
aws eks update-kubeconfig --name my-cluster
```

Create Kubernetes Deployment Manifest:

Write a Kubernetes Deployment manifest (deployment.yaml) for your application. This defines the desired state of your application, including the container image, replicas, ports, etc.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 3
  selector:
    matchLabels:
```

```
    app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
        - name: my-app-container
          image: your-container-image:tag
          ports:
            - containerPort: 80
```

Apply the Deployment Manifest:

Apply the deployment manifest to your EKS cluster.

```
kubectl apply -f deployment.yaml
```

Create a Service:

Create a Kubernetes Service to expose your application to the internet or other services within the cluster.

```
apiVersion: v1
kind: Service
metadata:
  name: my-app-service
spec:
  selector:
    app: my-app
  ports:
    - protocol: TCP
```

```
    port: 80
    targetPort: 80
  type: LoadBalancer
```

Apply the Service Manifest:

Apply the service manifest to your EKS cluster.

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

Access Your Application:

Once the LoadBalancer service is provisioned (this might take a few minutes), you can access your application using the LoadBalancer's DNS endpoint.