Exposing Applications with Ingress

Relevant Documentation

- Ingress

Exam Tips

- An Ingress manages external access to Kubernetes applications.
- An Ingress routes to 1 or more Kubernetes Services.
- You need an Ingress controller to implement the Ingress functionality. Which controller you use determines the specifics of how the Ingress will work.

Lesson Reference

Log in to the **control plane node**.

Create a Pod.

```
vi ingress-test-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: ingress-test-pod
  labels:
    app: ingress-test
spec:
  containers:
  - name: nginx
    image: nginx:stable
    ports:
    - containerPort: 80
```

```
kubectl apply -f ingress-test-pod.yml
```

Create a ClusterIP Service for the Pod.

```
vi ingress-test-service.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: ingress-test-service
spec:
  type: ClusterIP
  selector:
    app: ingress-test
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
kubectl apply -f ingress-test-service.yml
```

Get the Service's cluster IP address.

```
kubectl get service ingress-test-service
```

Use the cluster IP address to test the Service.

```
curl <Service cluster IP address>
```

Create an Ingress.

```
vi ingress-test-ingress.yml
```

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-test-ingress
spec:
  ingressClassName: nginx
  rules:
  - host: ingresstest.acloud.guru
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: ingress-test-service
            port:
              number: 80
```

```
kubectl apply -f ingress-test-ingress.yml
```

View a list of ingress objects.

```
kubectl get ingress
```

Get more details about the ingress. You should notice `ingress-test-service` listed as one of the backends.

```
kubectl describe ingress ingress-test-ingress
```

If you want to try accessing your Service through an Ingress controller, you will need to do some additional configuration.

First, use Helm to install the nginx Ingress controller.

```
helm repo add nginx-stable https://helm.nginx.com/stable

helm repo update

kubectl create namespace nginx-ingress

helm install nginx-ingress nginx-stable/nginx-ingress -n nginx-ingress
```

Get the cluster IP of the Ingress controller's Service.

```
kubectl get svc nginx-ingress-nginx-ingress -n nginx-ingress
```

Edit your `hosts` file.

```
sudo vi /etc/hosts
```

Use the cluster IP to add an entry to the hosts file.

```
<Nginx ingress controller Service cluster IP> ingresstest.acloud.guru
```

Now, test your setup.

```
curl ingresstest.acloud.guru
```