Creating a Service and exposing it with NodePort

Section 8, Lecture 66

Exposing Application with a Service

Types of Services:

* ClusterIP
* NodePort
* LoadBalancer
* ExternalName

1. kubectl get pods
2. kubectl get svc

Sample Output:

1. NAME READY STATUS RESTARTS AGE
2. voting-appp-j52x 1/1 Running 0 2m
3. voting-appp-pr2xz 1/1 Running 0 m
4. voting-appp-qpxbm 1/1 Running 0 5m

**Setting up monitoring**

If you are not running a monitoring screen, start it in a new terminal with the following command.

1. watch -n 1 kubectl get pod,deploy,rs,svc

**Writing Service Spec**

Lets start writing the meta information for service.

Filename: vote-svc.yaml

1. ---
2. apiVersion: v1
3. kind: Service
4. metadata:
5. name: vote
6. labels:
7. role: vote
8. spec:

And then add the spec to it. Refer to Service (v1 core) api at this page https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.10/

1. ---
2. apiVersion: v1
3. kind: Service
4. metadata:
5. name: vote
6. labels:
7. role: vote
8. spec:
9. selector:
10. role: vote
11. ports:
12. - port: 80
13. targetPort: 80
14. nodePort: 30000
15. type: NodePort

Save the file.

Now to create a service:

1. kubectl apply -f vote-svc.yaml --dry-run
2. kubectl apply -f vote-svc.yaml
3. kubectl get svc

Now to check which port the pod is connected

1. kubectl describe service vote

**Check for the Nodeport here**

Sample Output

1. Name: vote
2. Namespace: instavote
3. Labels: role=svc
4. tier=front
5. Annotations: kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"role":"svc","tier":"front"},"name":"vote","namespace":"instavote"},"spec":{...
6. Selector: app=vote
7. Type: NodePort
8. IP: 10.108.108.157
9. Port: <unset> 80/TCP
10. TargetPort: 80/TCP
11. NodePort: <unset> 31429/TCP
12. Endpoints: 10.38.0.4:80,10.38.0.5:80,10.38.0.6:80 + 2 more...
13. Session Affinity: None
14. External Traffic Policy: Cluster
15. Events: <none>

Go to browser and check hostip:NodePort

Here the node port is 31429.

Sample output will be:

