Hands-on Exercises

Exercise 1 – ClusterIP service

# Objective

The purpose of this exercise is for the student to create a functioning Kubernetes ClusterIP service. This service will then be used in a future exercise with a Kubernetes deployment as part of a “full-stack” application.

# Task 1 – Service creation

The ClusterIP service can be considered the main-stay of a Kubernetes cluster. Meaning, it is the service by which one or more pods can be located by other pods without having to track IP addresses. The service acts as a access-gateway or fixed-point of reference to an application (or piece of an application) which can be one or more pods

Kubernetes keeps track of the IP address assigned to the service. The service keeps track of the IP address(es) of the associated pods. In this manner, an application can reference, say, the database service. Kubernetes “looks up” database-service, then directs the traffic to the appropriate service which in turn directs the request to the appropriate pod.

1. Log into your master node.
2. All work should be placed in a subdirectory of your home-directory. So, create a subdirectory named: ~/exercises
3. Change directory to ~/exercises
4. Using the ClusterIP diagram in the slide presentation as a guide, create a mysql-service.yaml file (mysql’s default listening port is 3306). (Hint: Watch your yaml formatting file!)
5. Using kubectl, create the service
6. Get a list of services to ensure the service has been created
7. Describe the service via kubectl. Write down the various parameters for use in a later exercise.

When you have completed this task, you should have something that looks like this ( kubectl aliased to ‘kc’):

**Kubernetes output**

|  |
| --- |
| ubuntu@master2004:~/exercises$ **kc get svc**  NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 2d17h  mysql-service ClusterIP 10.100.6.51 <none> 3306/TCP 34h  ubuntu@master2004:~/exercises$ **kc describe service mysql-service**  Name: mysql-service  Namespace: default  Labels: tier=backend  Annotations: <none>  Selector: tier=backend  Type: ClusterIP  IP Family Policy: SingleStack  IP Families: IPv4  IP: 10.100.6.51  IPs: 10.100.6.51  Port: <unset> 3306/TCP  TargetPort: 3306/TCP  Endpoints: <none>  Session Affinity: None  Events: <none> |

**mysql-service.yaml**

|  |
| --- |
| apiVersion: v1  kind: Service  metadata:  name: gowebapp-mysql  labels:  tier: backend  spec:  type: ClusterIP  ports:  - port: 3306  targetPort: 3306  selector:  tier: backend |

## Questions

1. Can you assign an IP address to the service? Y or N
2. Does the targetPort have to be 3306?
3. Can a different port number be used for the targetPort?
4. What else might be affected if the targetPort is different that that shown?
5. What is the purpose of the label?
6. Why is this a ClusterIP service but the service type was not specified in the yaml file?

Exercise 2 – NodePort service

# Objective

The purpose of this exercise is for the student to create a functioning Kubernetes NodePort service. This service will then be used in a future exercise with a Kubernetes deployment as part of a “full-stack” application.

# Task 1 – Service creation

The NodePort service can be considered front door of an application running in a Kubernetes cluster.. It is the service which allows traffic from outside of the cluster to access objects (applications) inside the cluster. One or more pods “fronted” without having to track IP addresses as the port number specified is opened on all the Kubernetes nodes to outside traffic.

When the NodePort service is created, Kubernetes automatically creates the companion ClusterIP service based on the port-assignment in the NodePort specification. The ClusterIP service is what keeps track of the IP address(es) of the associated pods as well as the targetPort assignment. This allows an application to be reachable from an outside network (a.k.a., the internet) via any node in the cluster.

1. Log into your master node.
2. All work should be placed in a subdirectory of your home-directory. So, create a subdirectory named: ~/exercises
3. Change directory to ~/exercises
4. Using the NodePort diagram in the slide presentation as a guide, create a frontend-service.yaml file. Keep in mind that the targetPort specified is the port that the actual application will be expecting traffic to arrive on. (Hint: Watch your yaml formatting file!)
5. Using kubectl, create the service
6. Get a list of services to ensure the service has been created
7. Describe the service via kubectl. Write down the various parameters for use in a later exercise.

When you have completed this task, you should have something that looks like this ( kubectl aliased to ‘kc’):

**Kubernetes output**

|  |
| --- |
| ubuntu@master2004:~/exercises$ **kc get svc**  NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  frontend-service NodePort 10.104.28.68 <none> 8080:31000/TCP 22m  kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 2d17h  mysql-service ClusterIP 10.100.6.51 <none> 3306/TCP 34h  ubuntu@master2004:~/exercises$ kc describe service frontend-service  Name: frontend-service  Namespace: default  Labels: tier=frontend  Annotations: <none>  Selector: tier=frontend  Type: NodePort  IP Family Policy: SingleStack  IP Families: IPv4  IP: 10.104.28.68  IPs: 10.104.28.68  Port: <unset> 8080/TCP  TargetPort: 8080/TCP  NodePort: <unset> 31000/TCP  Endpoints: <none>  Session Affinity: None  External Traffic Policy: Cluster  Events: <none> |

**frontend-NodePort.yaml**

|  |
| --- |
| apiVersion: v1  kind: Service  metadata:  name: gowebapp-mysql  labels:  tier: backend  spec:  type: ClusterIP  ports:  - port: 3306  targetPort: 3306  selector:  tier: backend |

# Questions

1. What value can be used for the specification, “port”?
2. Why are the labels important in this configuration?
3. Can any label be used?
4. What happens if you get the targetPort value wrong?
5. If you have all the ports specified correctly, are you guaranteed that your deployment will work correctly?