**Lesson 05 Demo 02**

**Configuring the DNS for Kubernetes Services and Pods**

**Objective:** To configure the domain name system (DNS) for Kubernetes services and pods to ensure proper network resolution and connectivity

**Tools required:** kubeadm, kubectl, kubelet, and containerd

**Prerequisites:** A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

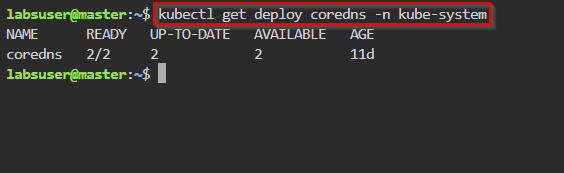
Steps to be followed:

1. Determine the default DNS in the cluster
2. Execute the DNS query
3. Configure the DNS policy
4. Create a custom DNS configuration

**Step 1:** **Determine the default DNS in the cluster**

1. To identify the core DNS deployment, execute the following command:

**kubectl get deploy coredns -n kube-system**



Kubernetes creates a default DNS in the **kube-system** namespace.

1. To identify the **coredns** pods using the selector, run the following command:

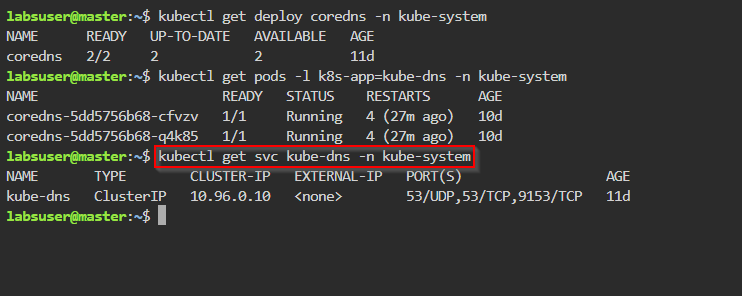
**kubectl get pods -l k8s-app=kube-dns -n kube-system**

A screen shot of a computer

Description automatically generated

1. To identify the **coredns** service, execute the following command:

**kubectl get svc kube-dns -n kube-system**



1. Use the following command to get the service endpoints:

**kubectl get endpoints kube-dns -n kube-system**

A screenshot of a computer

Description automatically generated

1. Run the following command to describe the endpoints:

**kubectl describe endpoints kube-dns -n kube-system**

A screenshot of a computer

Description automatically generated

**Step 2:** **Execute the DNS query**

1. Execute the following command to create an nginx deployment file:

**vi nginx.yaml**

**A screen shot of a computer

Description automatically generated**

1. Enter the following YAML code in the **nginx.yaml** file to define a Kubernetes deployment with two replicas, each running an nginx container on port 80:

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: my-nginx**

**spec:**

**selector:**

**matchLabels:**

**run: my-nginx**

**replicas: 2**

**template:**

**metadata:**

**labels:**

**run: my-nginx**

**spec:**

**containers:**

**- name: my-nginx**

**image: nginx**

**ports:**

**- containerPort: 80**

**A screenshot of a computer

Description automatically generated**

1. Run the following command to apply the configuration specified in the **nginx.yaml** file to create Kubernetes resources:

**kubectl apply -f nginx.yaml**

**A screenshot of a computer

Description automatically generated**

1. Run the following commands to get the status of the **my-nginx** deployment and list all pods with the **run=my-nginx** label:

**kubectl get deploy my-nginx**

**kubectl get pods -l run=my-nginx**

**A screenshot of a computer program

Description automatically generated**

1. Enter the following command to create the **my-nginx-service.yaml** file:

**vi my-nginx-service.yaml**

A screenshot of a computer

Description automatically generated

1. Add the following code to the YAML file:

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: my-nginx**

**spec:**

**type: NodePort**

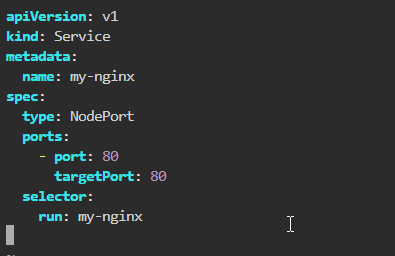
**ports:**

**- port: 80**

**targetPort: 80**

**selector:**

**run: my-nginx**



This YAML configuration defines a Kubernetes service named **my-nginx** of the type **NodePort**. It exposes port 80 and directs traffic to it on pods labeled with

**run: my-nginx.**

1. Run the following command to apply the configurations of the **my-nginx-service.yaml** file:

**kubectl apply -f my-nginx-service.yaml**

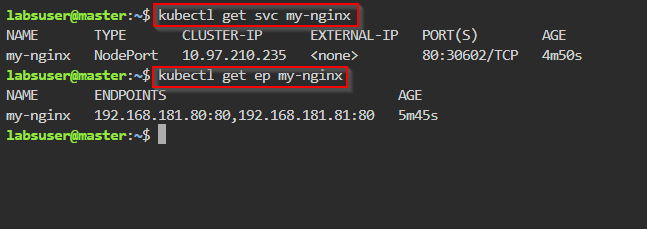
**A screenshot of a computer

Description automatically generated**

1. Run the following commands to retrieve the status and details of the **my-nginx** service and its associated endpoints:

**kubectl get svc my-nginx**

**kubectl get ep my-nginx**



1. To create a curl pod to perform a DNS query, run the following commands:

**kubectl run curl --image=radial/busyboxplus:curl -i –tty**

**nslookup google.com**

**nslookup my-nginx**

A screenshot of a computer program

Description automatically generated

**Note:** Create a pod using the **radial/busyboxplus:curl** image. This image has network tools pre-installed, which helps perform DNS queries.

`

1. Run the following command to create a local cluster:

**nslookup my-nginx.default.svc.cluster.local**

A screenshot of a computer program

Description automatically generated

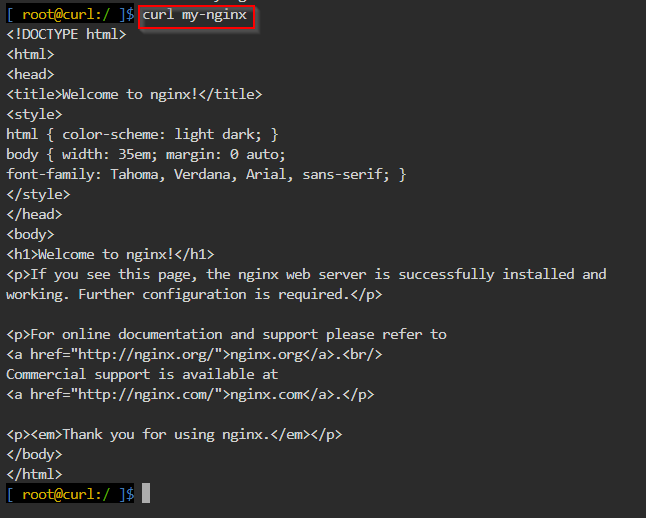
From this curl pod, you can access the **my-nginx** service.

**Note:** Use this format to run the local cluster:

**<service-name>.<namespace>.svc.cluster.local**.

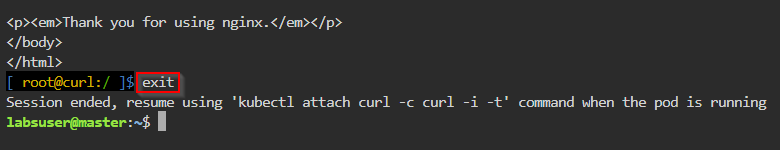
1. Run the following command to access the **my-nginx** file:

**curl my-nginx**



1. Enter the following command to exit the root directory:

**exit**



**Step 3:** **Configure the DNS policy**

1. In the master node, create a configuration file that defines the DNS policy for a Kubernetes pod using the following command:

**vi dnspolicy.yaml**

**A black rectangle with white text

Description automatically generated**

The YAML file will be created and opened in the **vi** editor.

1. Add the following YAML code inside the **dnspolicy.yaml** file to configure the DNS policy:

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: busybox**

**namespace: default**

**spec:**

**containers:**

**- image: busybox:1.28**

**command:**

**- sleep**

**- "3600"**

**imagePullPolicy: IfNotPresent**

**name: busybox**

**restartPolicy: Always**

**hostNetwork: true**

**dnsPolicy: ClusterFirstWithHostNet**

**A screenshot of a computer

Description automatically generated**

1. View the content of the **dnspolicy.yaml** file using the following command:

**cat dnspolicy.yaml**

**A screenshot of a computer

Description automatically generated**

1. Create a Kubernetes pod using the following command:

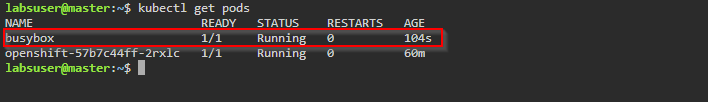
**kubectl apply -f dnspolicy.yaml**

**A black screen with white text

Description automatically generated**

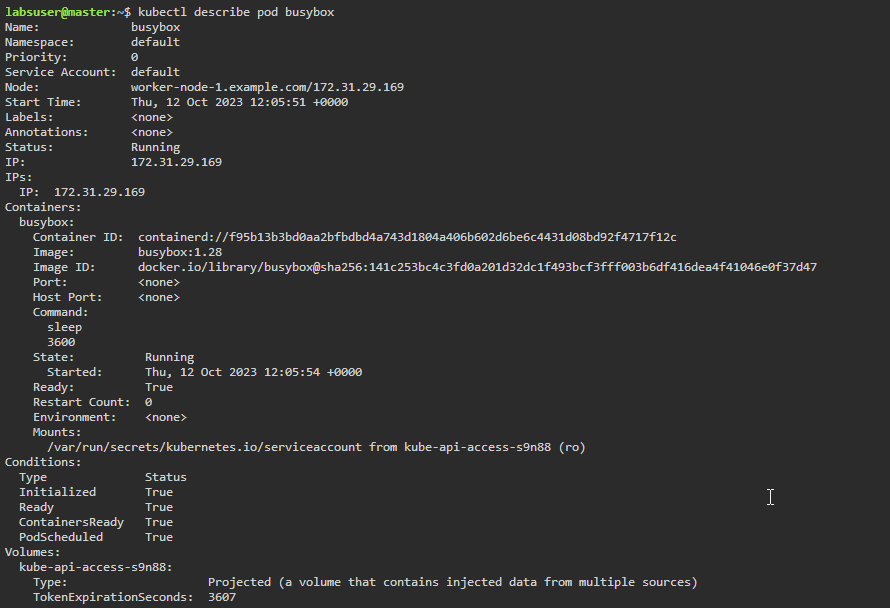
1. Execute the following command to list the newly created pod:

**kubectl get pods**

****

1. Execute the following command to list all the details regardingthe **busybox** pod:

**kubectl describe pod busybox**

****

**A screen shot of a computer

Description automatically generated**

**Step 4: Create a custom DNS configuration**

1. Create a DNS configuration YAML file using the following command:

**vi dnsconfig.yaml**

**A black rectangle with white text

Description automatically generated**

1. Add the following YAML code in the **dnsconfig.yaml** file:

**apiVersion: v1**

**kind: Pod**

**metadata:**

**namespace: default**

**name: dnscustomconfig**

**spec:**

**containers:**

**- name: test**

**image: nginx**

**dnsPolicy: "None"**

**dnsConfig:**

**nameservers:**

**- 1.2.3.4**

**searches:**

**- ns1.svc.cluster-domain.example**

**- my.dns.search.suffix**

**options:**

**- name: ndots**

**value: "2"**

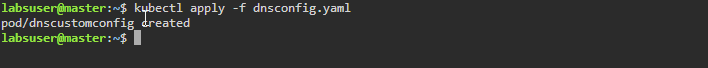
**- name: edns0**

**A screen shot of a computer

Description automatically generated**

1. Create another pod using the following command:

**kubectl apply -f dnsconfig.yaml**

****

1. Set up the IPv6 for the DNS connectivity using the following command:

**kubectl exec -it dnscustomconfig -- cat /etc/resolv.conf**

**A computer screen with white text

Description automatically generated**

By following these steps, you have successfully configured the DNS for Kubernetes services and pods, ensuring efficient network resolution and seamless connectivity.