Using Kubernetes Services with DNS

Your company, BeeBox, is working on evolving their Kubernetes application infrastructure. Currently, they have two application components in two different namespaces: a user database and a web frontend.

Your developers aren't quite sure how Kubernetes DNS works, and they are having trouble reaching some of their Services as a result. They have asked you to perform some tests that will help confirm and clarify for them how Kubernetes DNS behaves.

A Pod called busybox already exists in the web namespace. You can use this Pod to perform your tests. Save the output of your tests to some files, so that the developers can see the results.

In this lab we are going to look up DNS records for a service in the same namespace as our busybox pod. So we will be looking up DNS records for the web-frontend service and outputting that to a file. We then will look up records for a service in another namespace. Which would be the user-db service and we'll output those records to a file

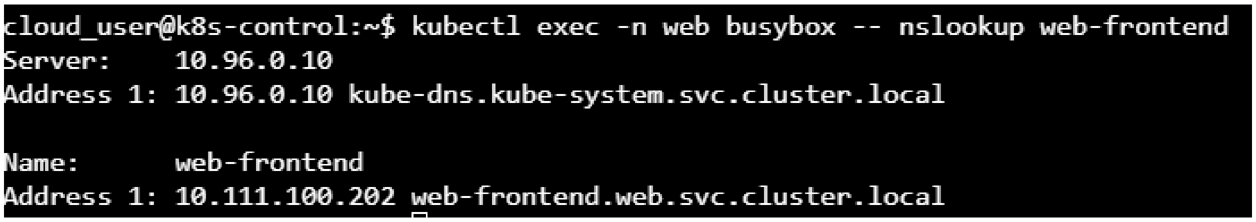
Perform an Nslookup for a Service in the Same Namespace

We will perform the DNS nslookups using both short names of the services, as well as the FQDN names.

1. Start using the busybox Pod in the web namespace to perform an nslookup on the web-frontend

kubectl exec -n web busybox -- nslookup web-frontend

Figure 1-1



```
cloud_user@k8s-control:~$ kubectl exec -n web busybox -- nslookup web-frontend
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      web-frontend
Address 1: 10.111.100.202 web-frontend.web.svc.cluster.local
```

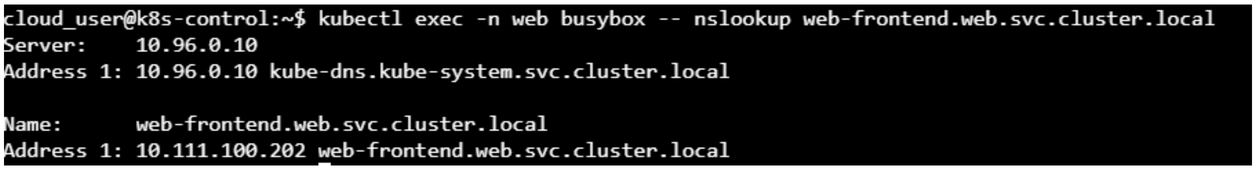
2. We need to save the results in a file, so rerun the previous command. But redirect the output (In the instructions above)

kubectl exec -n web busybox -- nslookup web-frontend >> ~/dns_same_namespace_results.txt

3. Now we need to lookup the same service this time using the FQDN

kubectl exec -n web busybox -- nslookup web-frontend.web.svc.cluster.local

Figure 1-2



```
cloud_user@k8s-control:~$ kubectl exec -n web busybox -- nslookup web-frontend.web.svc.cluster.local
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      web-frontend.web.svc.cluster.local
Address 1: 10.111.100.202 web-frontend.web.svc.cluster.local
```

4. No repeat step 2, same command to the same file

kubectl exec -n web busybox -- nslookup web-frontend.web.svc.cluster.local >> ~/dns_same_namespace_results.txt

5. So we done a lot so far, so we need to verify if everything looks okay before continuing on

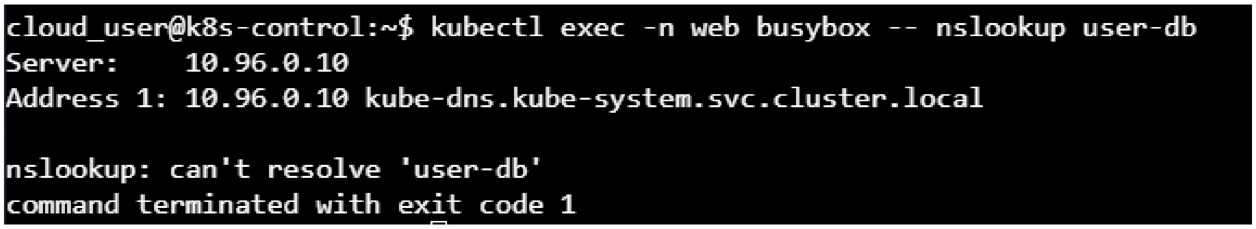
cat ~/dns_same_namespace_results.txt

Perform an Nslookup For a Service in a Different Namespace

6. Using the busybox pod in the web namespace to perform an nslookup on the user-db service in the data namespace

kubectl exec -n web busybox -- nslookup user-db

Figure 1-3



```
cloud_user@k8s-control:~$ kubectl exec -n web busybox -- nslookup user-db
Server:     10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

nslookup: can't resolve 'user-db'
command terminated with exit code 1
```

We got an error with an exit code 1

7. We now need to capture this into a file

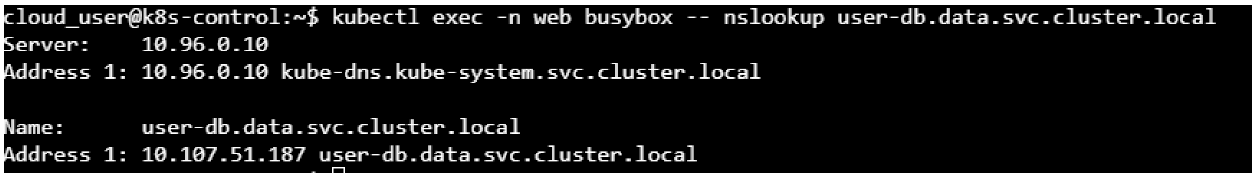
kubectl exec -n web busybox -- nslookup user-db >> ~/dns_different_namespace_results.txt

This command still gave us errors. But looking at Figure 1-3 looks like we may have to try the FQDN one.

8. Search the same lookup on the FQDN

kubectl exec -n web busybox -- nslookup user-db.data.svc.cluster.local

Figure 1-4



```
cloud_user@k8s-control:~$ kubectl exec -n web busybox -- nslookup user-db.data.svc.cluster.local
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      user-db.data.svc.cluster.local
Address 1: 10.107.51.187 user-db.data.svc.cluster.local
```

This was successful even though we were looking up a nervice in a different namespace. So now we have to rerun the command from #7 but of course change up the naming for the txt file

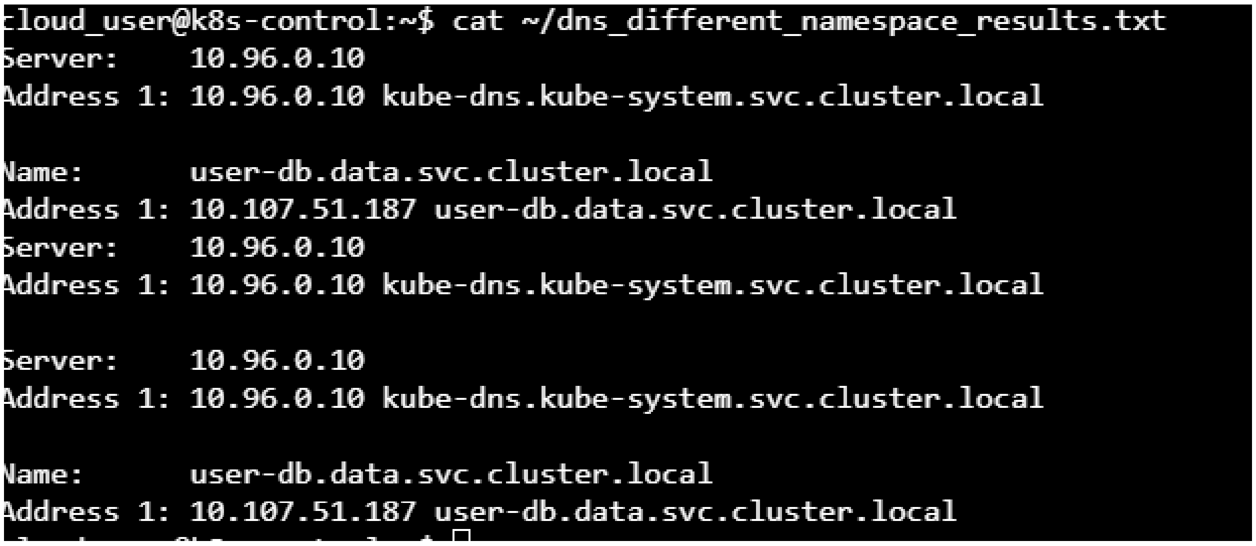
9. Save the results in a text file

kubectl exec -n web busybox -- nslookup user-db.data.svc.cluster.local >> ~/dns_different_namespace_results.txt

10. Now we check the output of the file

cat ~/dns_different_namespace_results.txt

Figure 1-5



```
loud_user@k8s-control:~$ cat ~/dns_different_namespace_results.txt
erver:    10.96.0.10
ddress 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

ame:      user-db.data.svc.cluster.local
ddress 1: 10.107.51.187 user-db.data.svc.cluster.local
erver:    10.96.0.10
ddress 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

erver:    10.96.0.10
ddress 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

ame:      user-db.data.svc.cluster.local
ddress 1: 10.107.51.187 user-db.data.svc.cluster.local
```

Completed