Agenda

- Api Groups
- Authorization
- Image Security
- Network Policies
- DNS

API Groups

Pre-Requisite

```
curl https://kube-master:6443/version
```

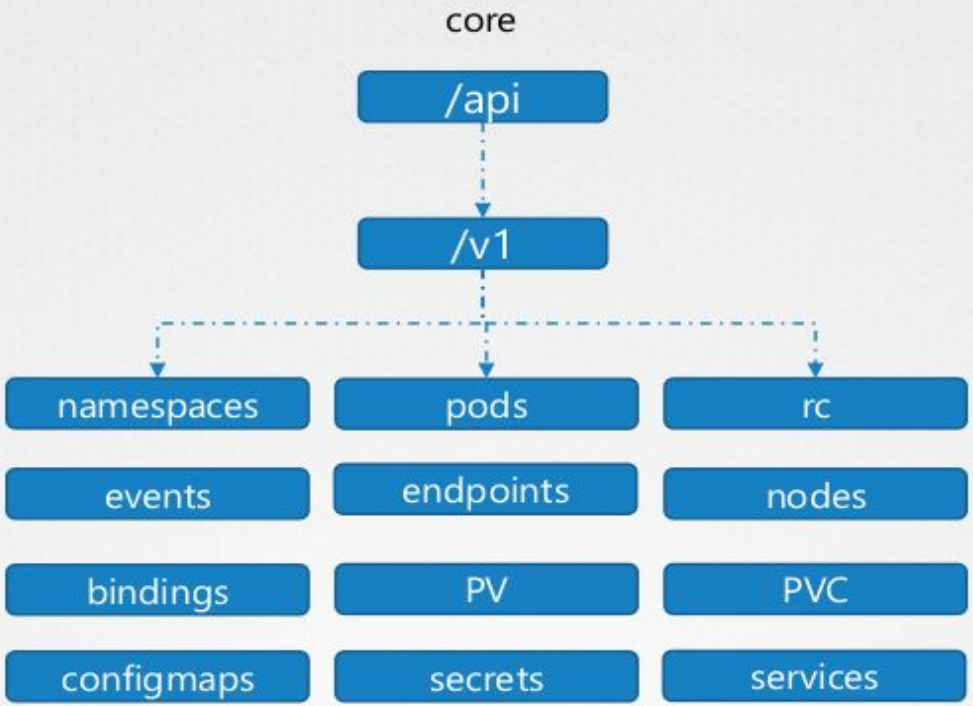

```
{
  "major": "1",
  "minor": "13",
  "gitVersion": "v1.13.0",
  "gitCommit": "ddf47ac13c1a9483ea035a79cd7c10005ff21a6d",
  "gitTreeState": "clean",
  "buildDate": "2018-12-03T20:56:12Z",
  "goVersion": "go1.11.2",
  "compiler": "gc",
  "platform": "linux/amd64"
}
```

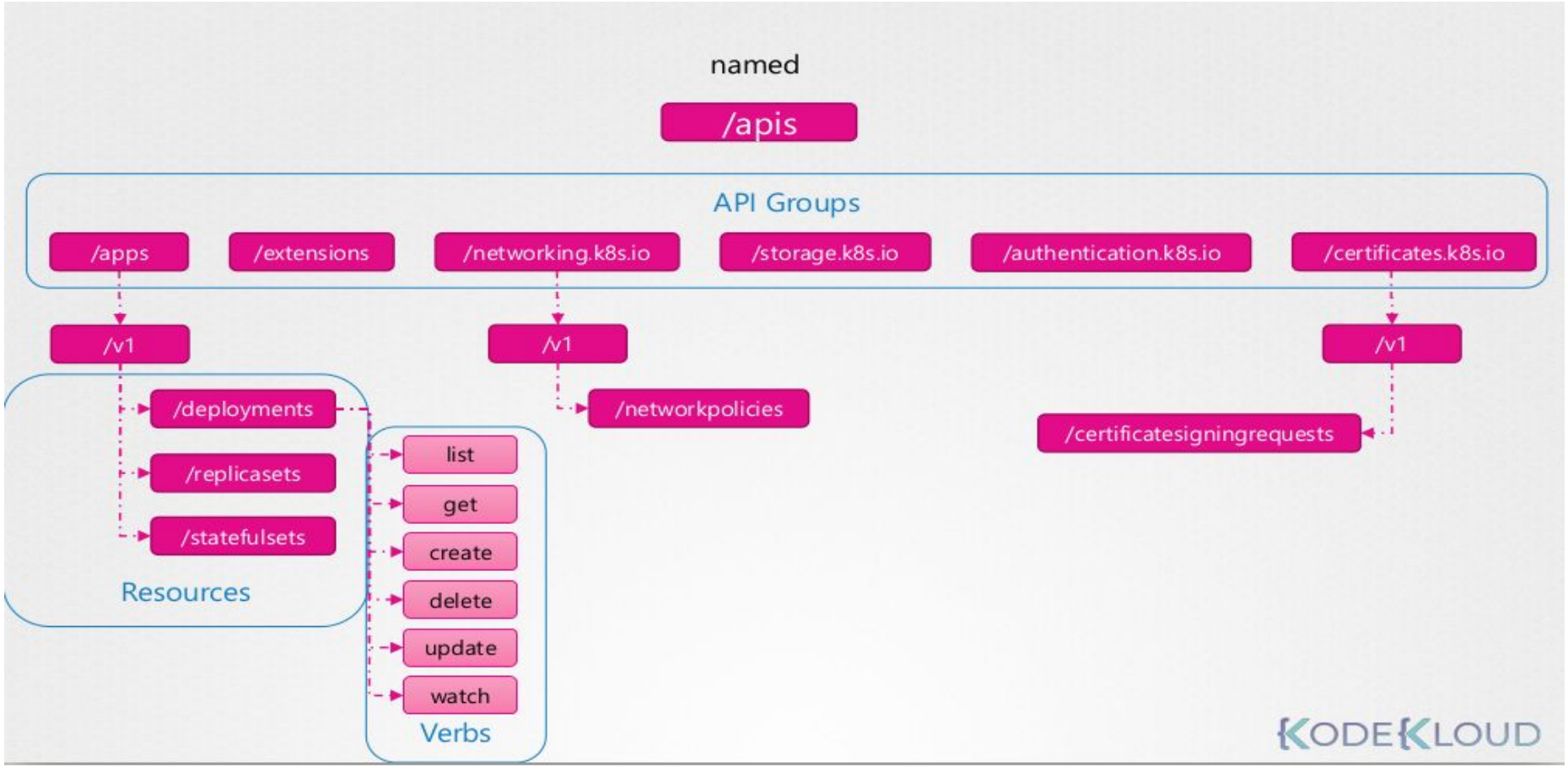

```
curl https://kube-master:6443/api/v1/pods
```


```
{
  "kind": "PodList",
  "apiVersion": "v1",
  "metadata": {
    "selfLink": "/api/v1/pods",
    "resourceVersion": "153068"
  },
  "items": [
    {
      "metadata": {
        "name": "nginx-5c7588df-ghsbd",
        "generateName": "nginx-5c7588df-",
        "namespace": "default",
        "creationTimestamp": "2019-03-20T10:57:48Z",
        "labels": {
          "app": "nginx",
          "pod-template-hash": "5c7588df"
        },
        "ownerReferences": [
          {
            "apiVersion": "apps/v1",
            "kind": "ReplicaSet",
            "name": "nginx-5c7588df",
            "uid": "398ce179-4af9-11e9-beb6-020d3114c7a7",
            "controller": true,
            "blockOwnerDeletion": true
          }
        ]
      },
```


/metrics
/healthz
/version
/api
/apis
/logs


core
named
/api
/apis


core
/api
/v1
namespaces
pods
rc
events
endpoints
nodes
bindings
PV
PVC
configmaps
secrets
services


named
/apis
API Groups
/apps
/extensions
/networking.k8s.io
/storage.k8s.io
/authentication.k8s.io
/certificates.k8s.io
/v1
/v1
/v1
/deployments
/replicasets
/statefulsets
Resources
/networkpolicies
/certificatesigningrequests
list
get
create
delete
update
watch
Verbs
KODEKLOUD

AUTHORIZATION

Authorization

What can they do?

- RBAC Authorization
- ABAC Authorization
- Node Authorization
- Webhook Mode

Why Authorization?

Admins

```
kubectl get pods
NAME       STATUS   ROLES    AGE     VERSION
worker-1   Ready    <none>   5d21h   v1.13.0
worker-2   Ready    <none>   5d21h   v1.13.0
```

```
kubectl get nodes
NAME       STATUS   ROLES    AGE     VERSION
worker-1   Ready    <none>   5d21h   v1.13.0
worker-2   Ready    <none>   5d21h   v1.13.0
```

```
kubectl delete node worker-2
Node worker-2 Deleted!
```

Developers

```
kubectl get pods
NAME       STATUS   ROLES    AGE     VERSION
worker-1   Ready    <none>   5d21h   v1.13.0
worker-2   Ready    <none>   5d21h   v1.13.0
```

```
kubectl get nodes
NAME       STATUS   ROLES    AGE     VERSION
worker-1   Ready    <none>   5d21h   v1.13.0
worker-2   Ready    <none>   5d21h   v1.13.0
```

```
kubectl delete node worker-2
Error from server (Forbidden): nodes
"worker-1" is forbidden: User "developer"
cannot delete resource "nodes"
```

Bots

```
kubectl get pods
Error from server (Forbidden): nodes
"worker-1" is forbidden: User “Bot-1”
delete resource "nodes"
```

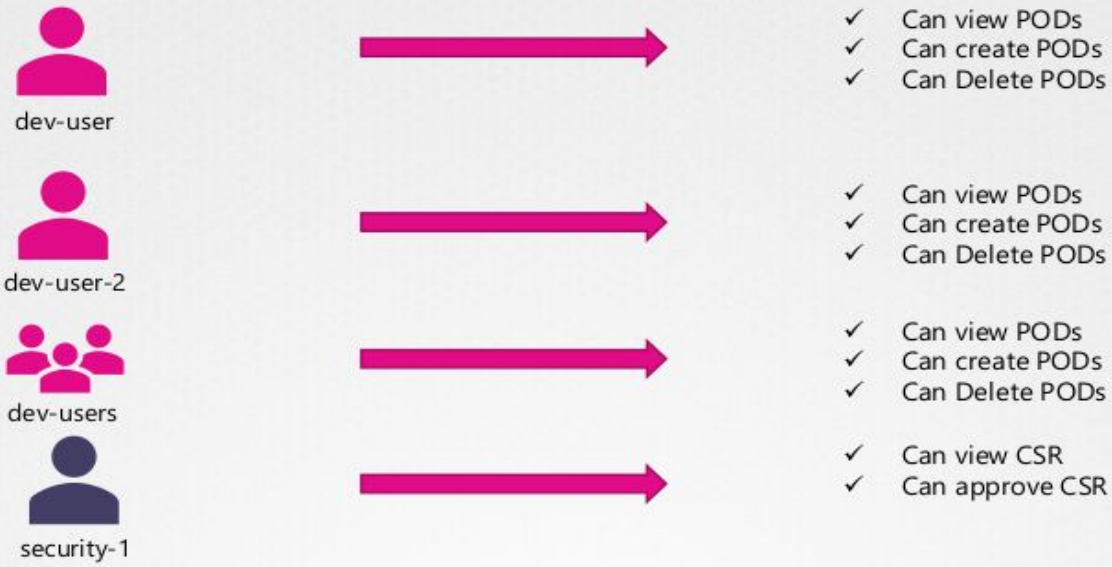
```
kubectl get nodes
Error from server (Forbidden): nodes
"worker-1" is forbidden: User “Bot-1”
delete resource "nodes"
```

```
kubectl delete node worker
Error from server (Forbidden): nodes
"worker-1" is forbidden: User “Bot-1”
delete resource "nodes"
```

Authorization Mechanisms

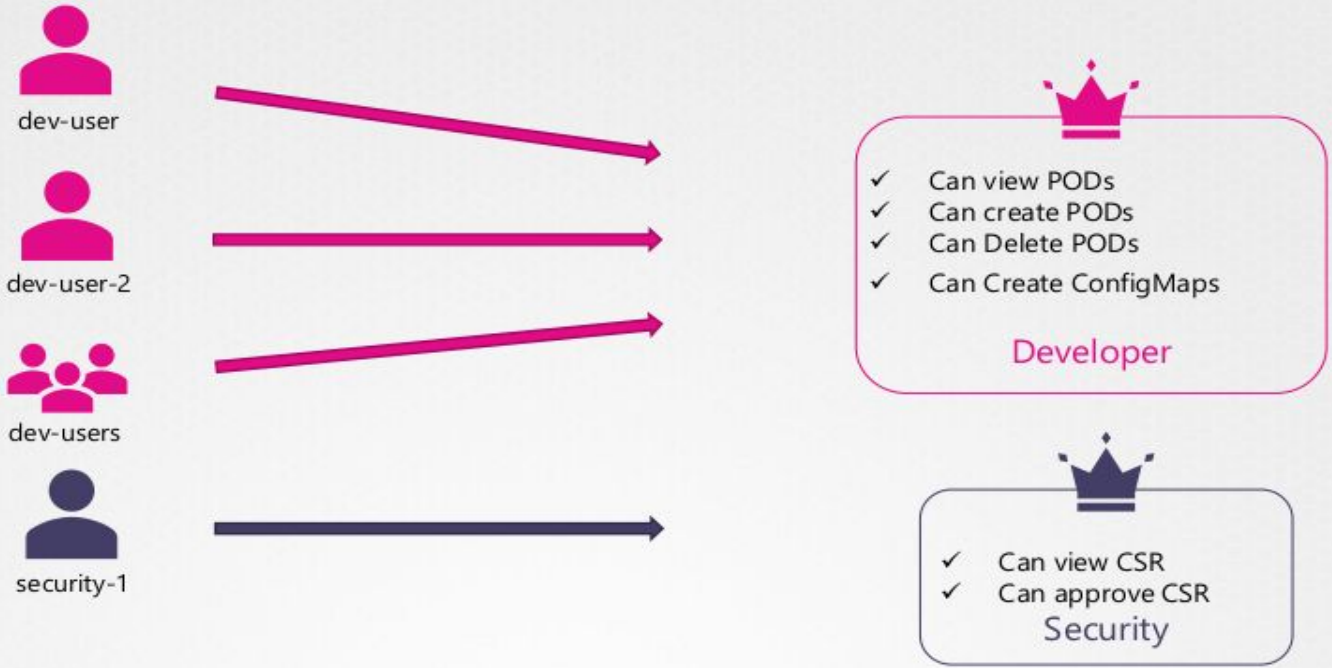
Node | ABAC | RBAC | Webhook

ABAC




```
{"kind": "Policy", "spec": {"user": "dev-user", "namespace": "*", "resource": "pods", "apiGroup": "*"}}
{"kind": "Policy", "spec": {"user": "dev-user-2", "namespace": "*", "resource": "pods", "apiGroup": "*"}}
{"kind": "Policy", "spec": {"group": "dev-users", "namespace": "*", "resource": "pods", "apiGroup": "*"}}
{"kind": "Policy", "spec": {"user": "security-1", "namespace": "*", "resource": "csr", "apiGroup": "*"}}
```

RBAC

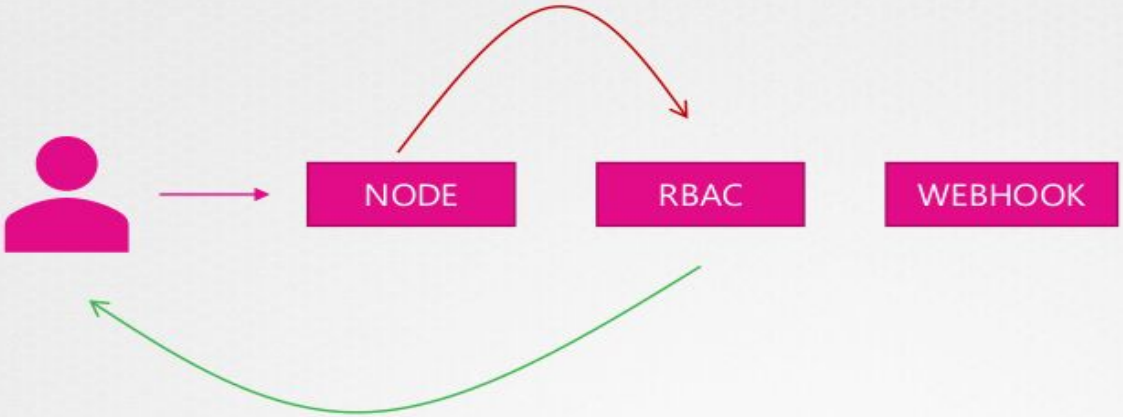

dev-user
dev-user-2
dev-users
security-1
✓ Can view PODs
✓ Can create PODs
✓ Can Delete PODs
✓ Can Create ConfigMaps
Developer
✓ Can view CSR
✓ Can approve CSR
Security

Authorization Mode

AlwaysAllow | NODE | ABAC | RBAC | WEBHOOK | AlwaysDeny

```
ExecStart=/usr/local/bin/kube-apiserver \\
  --advertise-address=${INTERNAL_IP} \\
  --allow-privileged=true \\
  --apiserver-count=3 \\
  --authorization-mode=Node,RBAC,Webhook \\
  --bind-address=0.0.0.0 \\
  --enable-swagger-ui=true \\
  --etcd-cafile=/var/lib/kubernetes/ca.pem \\
  --etcd-certfile=/var/lib/kubernetes/apiserver-etcd-client.crt \\
  --etcd-keyfile=/var/lib/kubernetes/apiserver-etcd-client.key \\
  --etcd-servers=https://127.0.0.1:2379 \\
  --event-ttl=1h \\
  --kubelet-certificate-authority=/var/lib/kubernetes/ca.pem \\
  --kubelet-client-certificate=/var/lib/kubernetes/apiserver-etcd-client.crt \\
  --kubelet-client-key=/var/lib/kubernetes/apiserver-etcd-client.key \\
  --service-node-port-range=30000-32767 \\
  --client-ca-file=/var/lib/kubernetes/ca.pem \\
  --tls-cert-file=/var/lib/kubernetes/apiserver.crt \\
```

Authorization Mode




```
ExecStart=/usr/local/bin/kube-apiserver \\
  --advertise-address=${INTERNAL_IP} \\
  --allow-privileged=true \\
  --apiserver-count=3 \\
  --authorization-mode=Node,RBAC,Webhook \\
  --bind-address=0.0.0.0 \\
```

RBAC

RBAC

- ✓ Can view PODs
- ✓ Can create PODs
- ✓ Can Delete PODs
- ✓ Can Create ConfigMaps

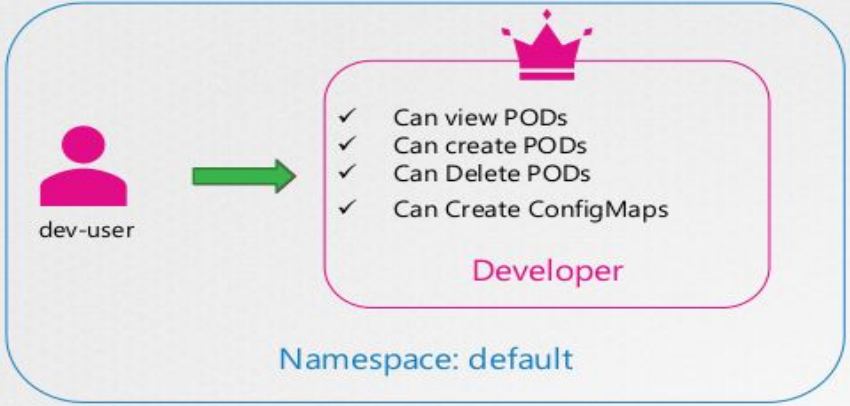
Developer

developer-role.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: developer
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["list", "get", "create", "update", "delete"]
- apiGroups: [""]
  resources: ["ConfigMap"]
  verbs: ["create"]
```

```
kubectl create -f developer-role.yaml
```

RBAC




```
kubectl create -f devuser-developer-binding.yaml
```

developer-role.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: developer
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["list", "get", "create", "update", "de
- apiGroups: [""]
  resources: ["ConfigMap"]
  verbs: ["create"]
```

devuser-developer-binding.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: devuser-developer-binding
subjects:
- kind: User
  name: dev-user
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: developer
  apiGroup: rbac.authorization.k8s.io
```

View RBAC

```
kubectl get roles
```

```
NAME        AGE
developer   4s
```

```
kubectl get rolebindings
```

```
NAME                          AGE
devuser-developer-binding     24s
```

```
kubectl describe role developer
```

```
Name:         developer
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources  Non-Resource URLs  Resource Names  Verbs
  ---------  -----------------  --------------  -----
  ConfigMap  []                 []              [create]
  pods       []                 []              [get watch list create delete]
```

View RBAC

```
kubectl describe rolebinding devuser-developer-binding
Name:         devuser-developer-binding
Labels:       <none>
Annotations:  <none>
Role:
  Kind:  Role
  Name:  developer
Subjects:
  Kind  Name      Namespace
  ----  ----      ---------
  User  dev-user
```

Check Access

```
kubectl auth can-i create deployments
yes
```

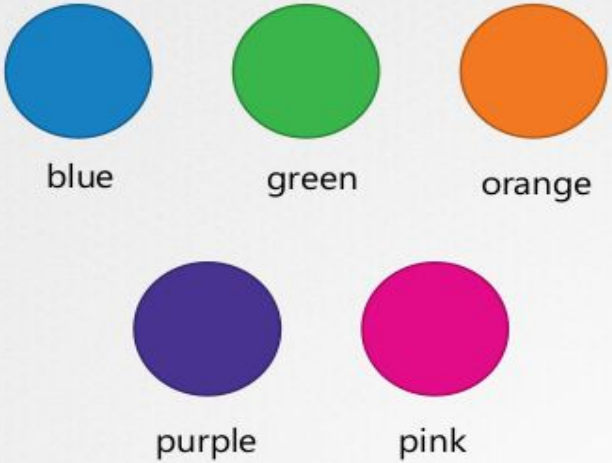
```
kubectl auth can-i delete nodes
no
```

```
kubectl auth can-i create deployments --as dev-user
no
```

```
kubectl auth can-i create pods --as dev-user
yes
```

```
kubectl auth can-i create pods --as dev-user --namespace test
no
```

Resource Names

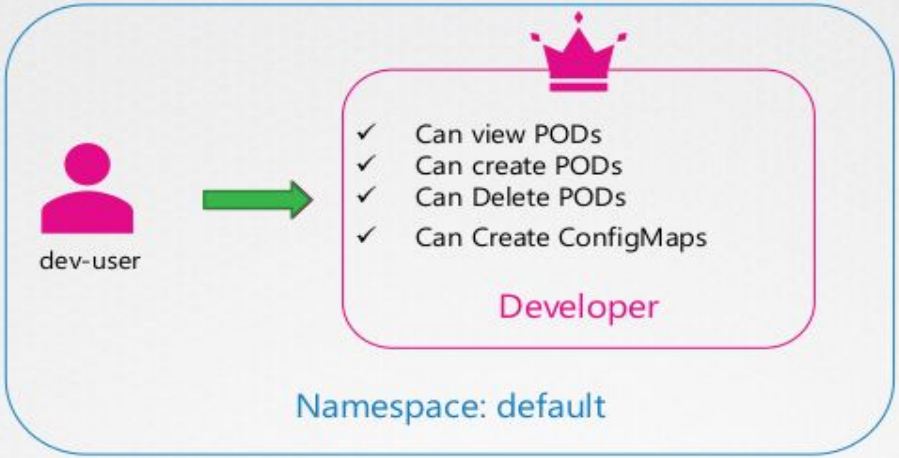


developer-role.yaml

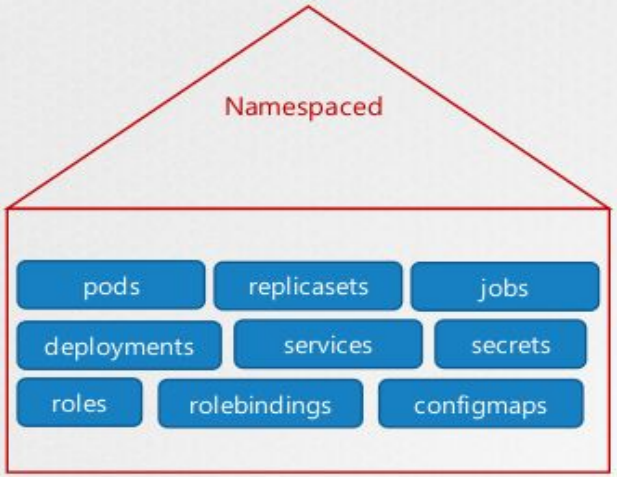
```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: developer
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "create", "update"]
  resourceNames: ["blue", "orange"]
```

Cluster Roles

Roles


dev-user
✓ Can view PODs
✓ Can create PODs
✓ Can Delete PODs
✓ Can Create ConfigMaps
Developer
Namespace: default

Namespace



Cluster Scoped

nodes
PV
PVC
clusterroles
clusterrolebindings
certificatesigningrequests
namespaces

```
kubectl api-resources --namespaced=true
```

```
kubectl api-resources --namespaced=false
```

clusterroles

- ✓ Can view Nodes
- ✓ Can create Nodes
- ✓ Can delete Nodes

Cluster Admin

- ✓ Can view PVs
- ✓ Can create PVs
- ✓ Can delete PVCs

Storage Admin

cluster-admin-role.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: cluster-administrator
rules:
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["list", "get", "create", "delete"]
```

```
kubectl create -f cluster-admin-role.yaml
```

clusterrolebinding

cluster-admin

- ✓ Can view Nodes
- ✓ Can create Nodes
- ✓ Can delete Nodes

Cluster Admin

```
kubectl create -f cluster-admin-role-binding.yaml
```

cluster-admin-role.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: cluster-administrator
rules:
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["list", "get", "create", "delete"]
```

cluster-admin-role-binding.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: cluster-admin-role-binding
subjects:
- kind: User
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: cluster-administrator
  apiGroup: rbac.authorization.k8s.io
```

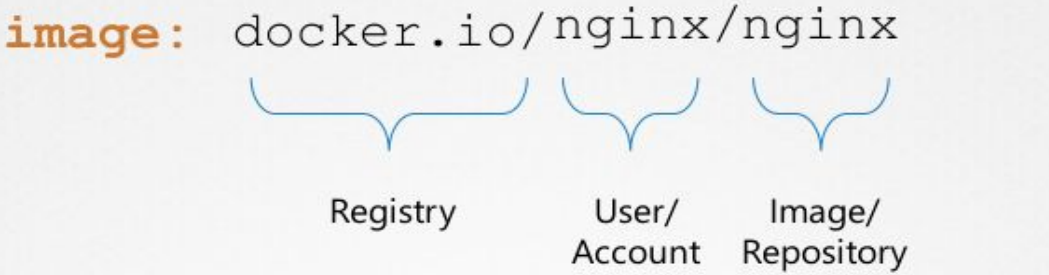
Image Security

Image

nginx-pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx
    image: nginx
```

Image



gcr.io/ kubernetes-e2e-test-images/dnsutils

Private Repository

```
docker login private-registry.io
```

```
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to
https://hub.docker.com to create one.
Username: registry-user
Password:
WARNING! Your password will be stored unencrypted in /home/vagrant/.docker/config.json.

Login Succeeded
```

```
docker run private-registry.io/apps/internal-app
```

Private Repository

```
docker login private-registry.io
```

```
docker run private-registry.io/apps/internal-app
```

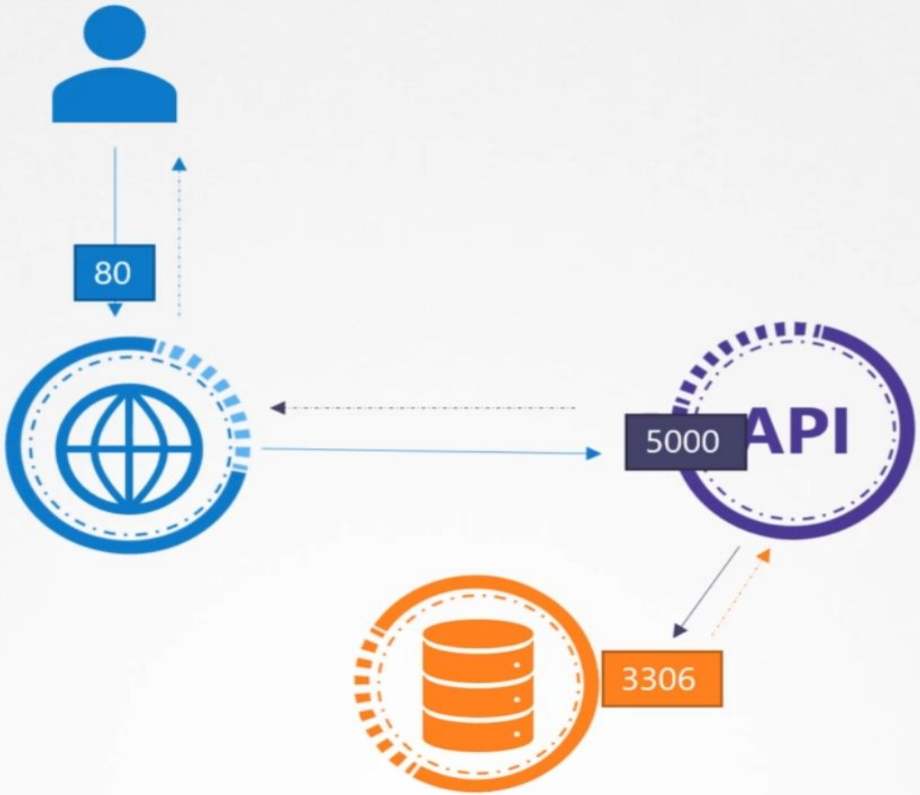
```
kubectl create secret docker-registry regcred \
    --docker-server= private-registry.io     \
    --docker-username= registry-user  \
    --docker-password= registry-password  \
    --docker-email= registry-user@org.com
```

nginx-pod.yaml

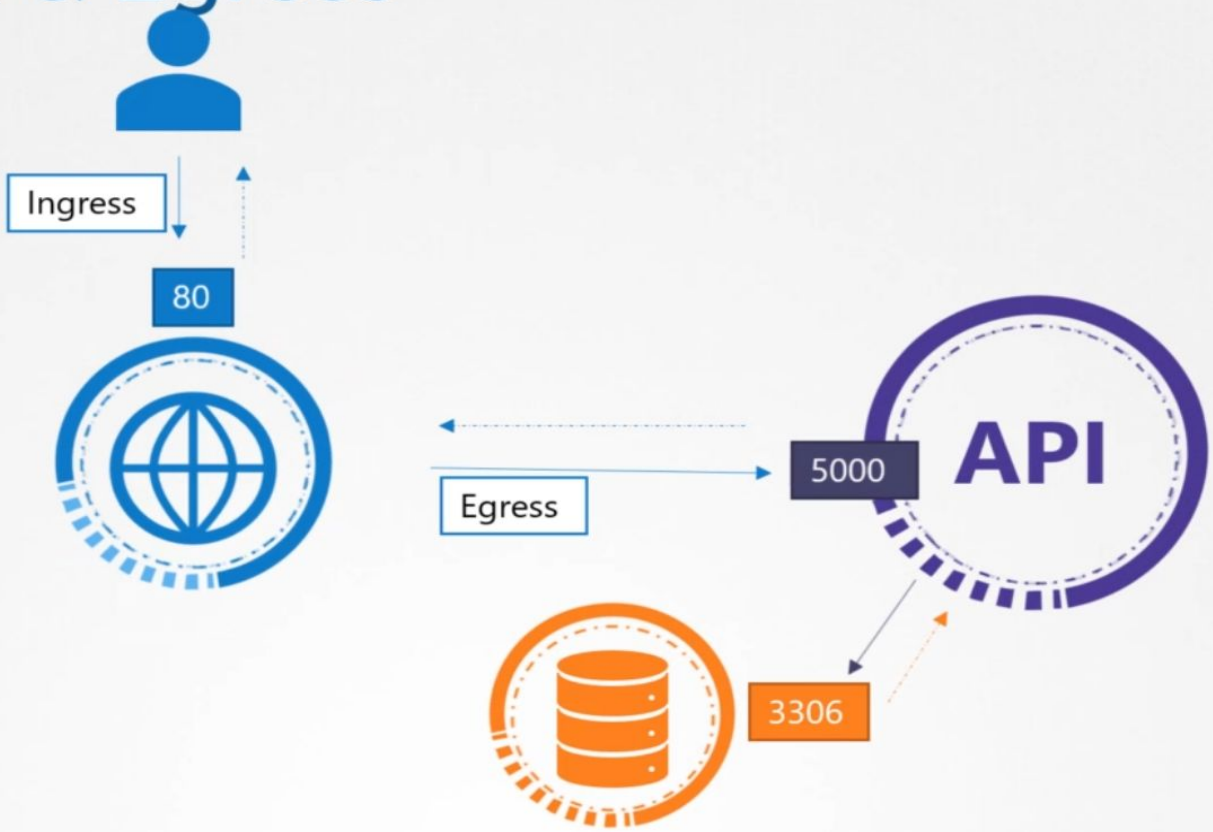
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx
    image: private-registry.io/apps/internal-app
  imagePullSecrets:
  - name: regcred
```

Network Policies

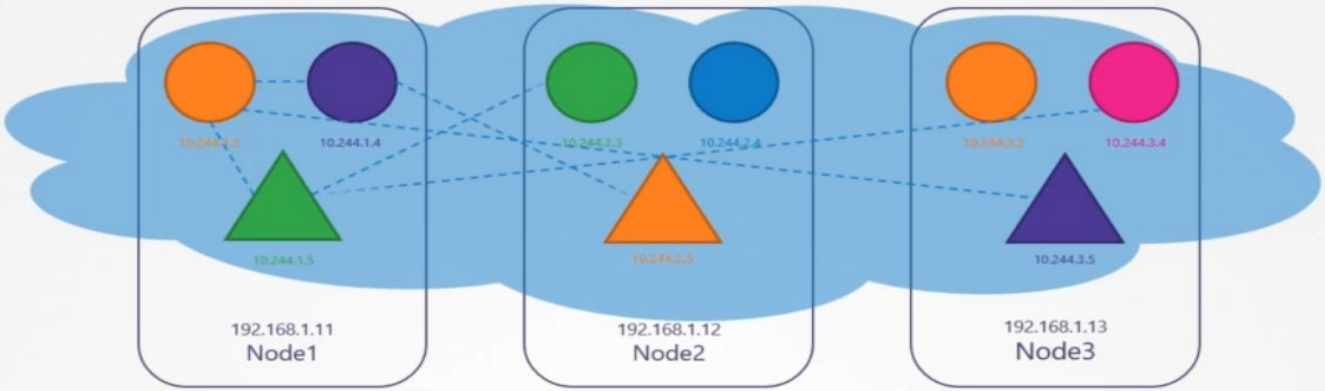
Traffic



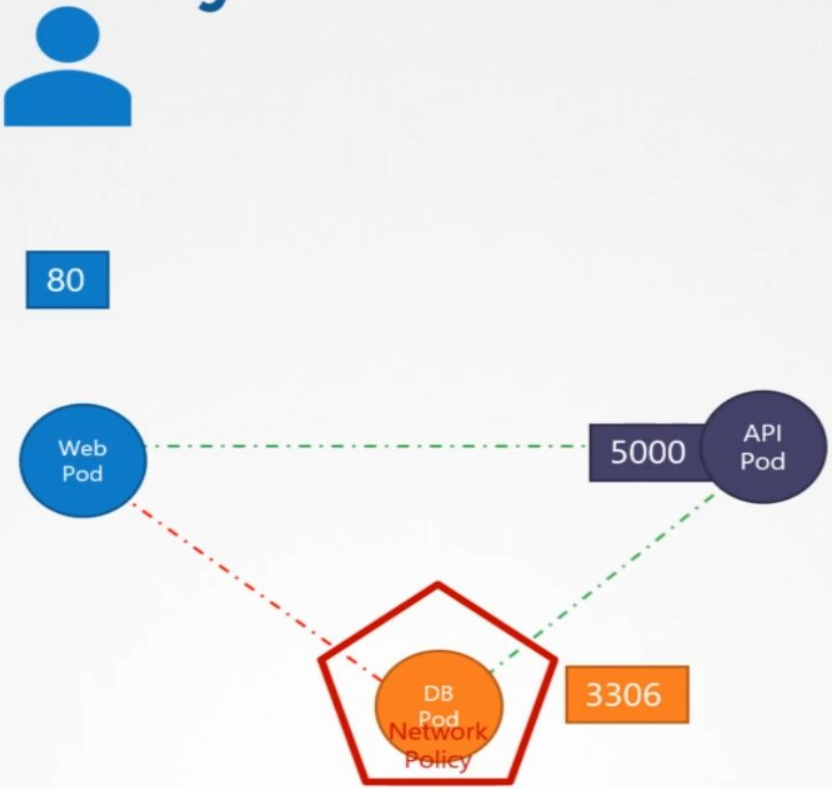
Ingress & Egress



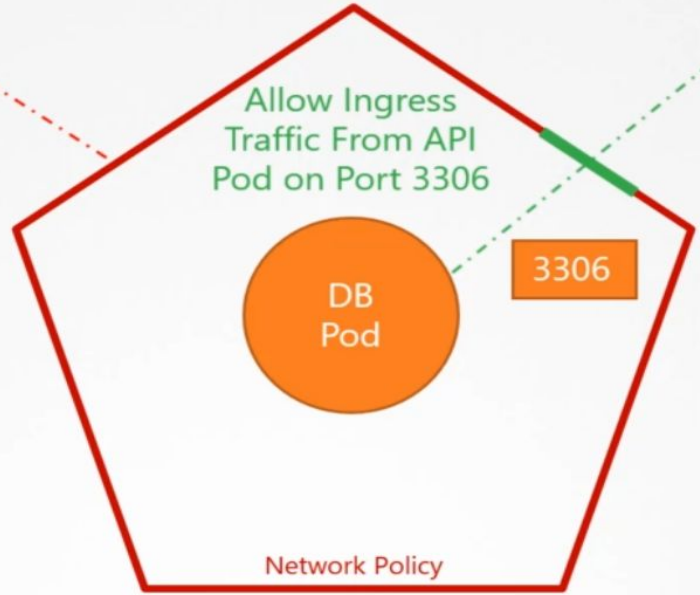
Network Security



Network Policy

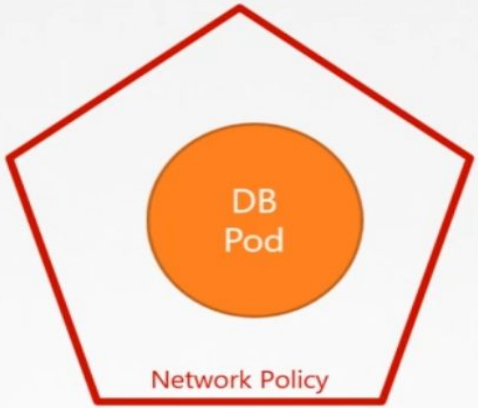


Network Policy



Network Policy - Selectors

Allow Ingress
Traffic From API
Pod on Port 3306




```
podSelector:
  matchLabels:
    role: db
```

```
labels:
  role: db
```

Network Policy - Rules

```
policyTypes:
- Ingress
ingress:
- from:
  - podSelector:
      matchLabels:
        name: api-pod
  ports:
  - protocol: TCP
    port: 3306
```

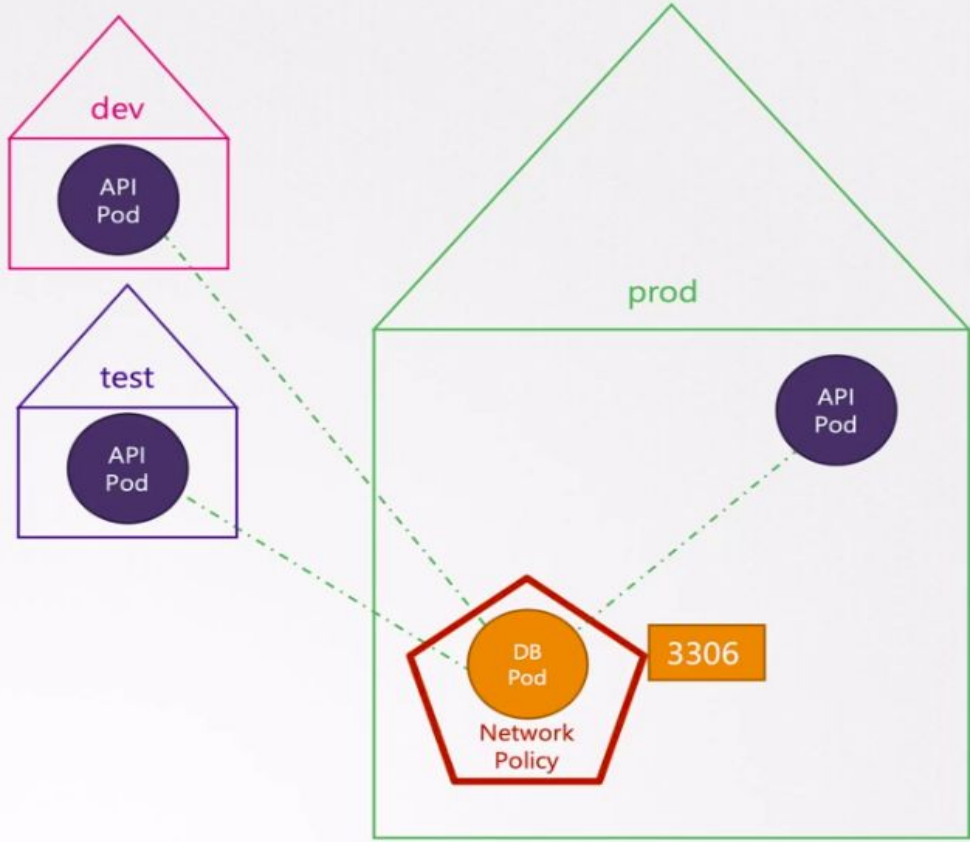
Allow Ingress Traffic From API Pod on Port 3306

Network Policy

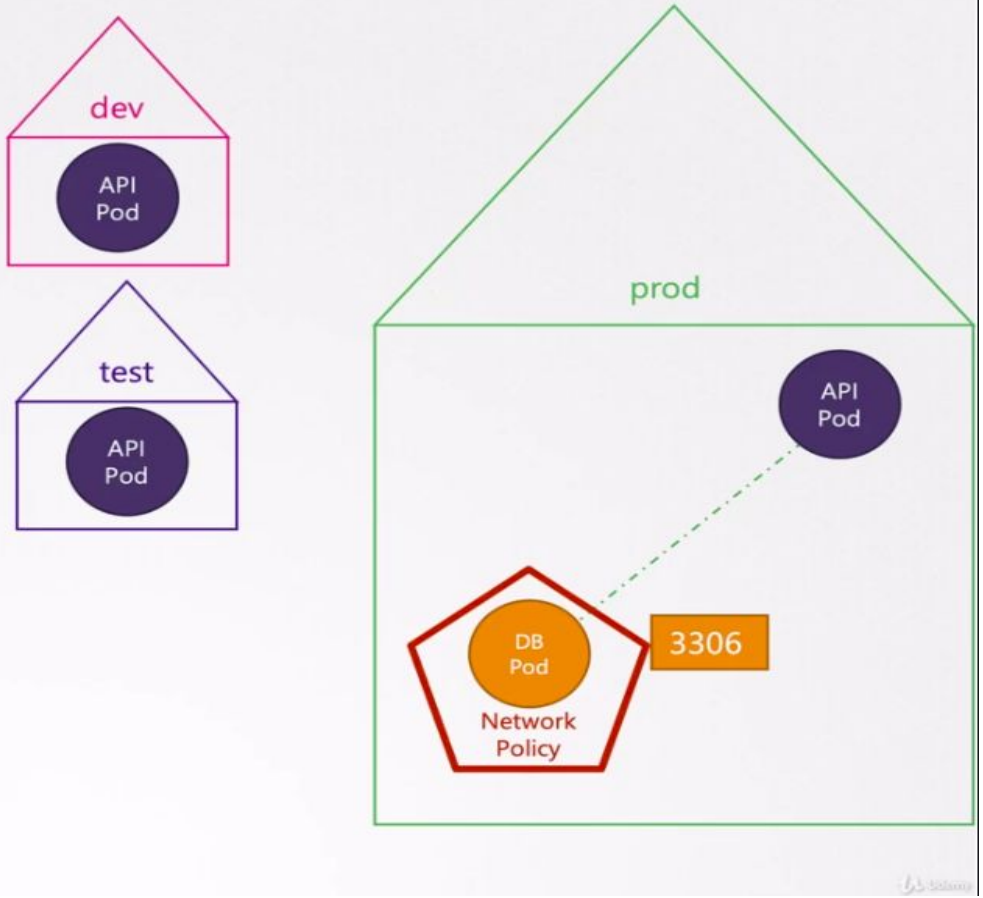
```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
   podSelector:
     matchLabels:
       role: db
   policyTypes:
   - Ingress
   ingress:
   - from:
     - podSelector:
         matchLabels:
           name: api-pod
     ports:
     - protocol: TCP
       port: 3306
```

```
kubectl create -f policy-definition.yaml
```

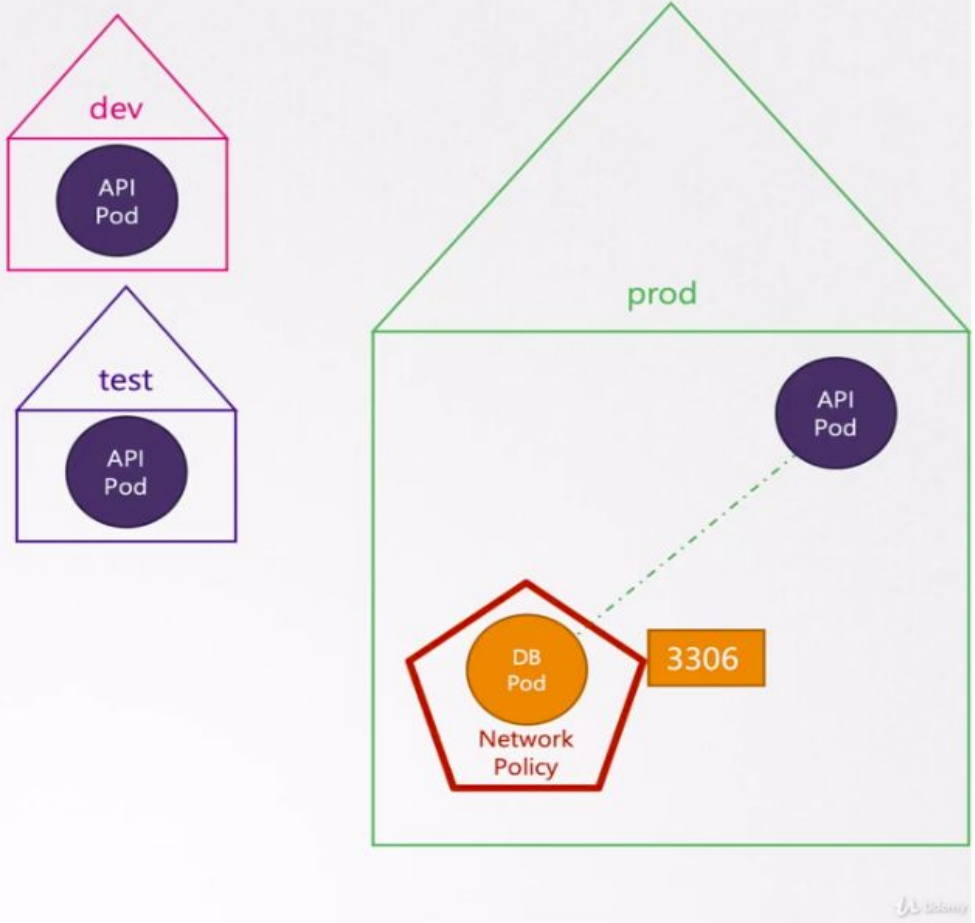
```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db

  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
          matchLabels:
            name: api-pod
    ports:
    - protocol: TCP
      port: 3306
```



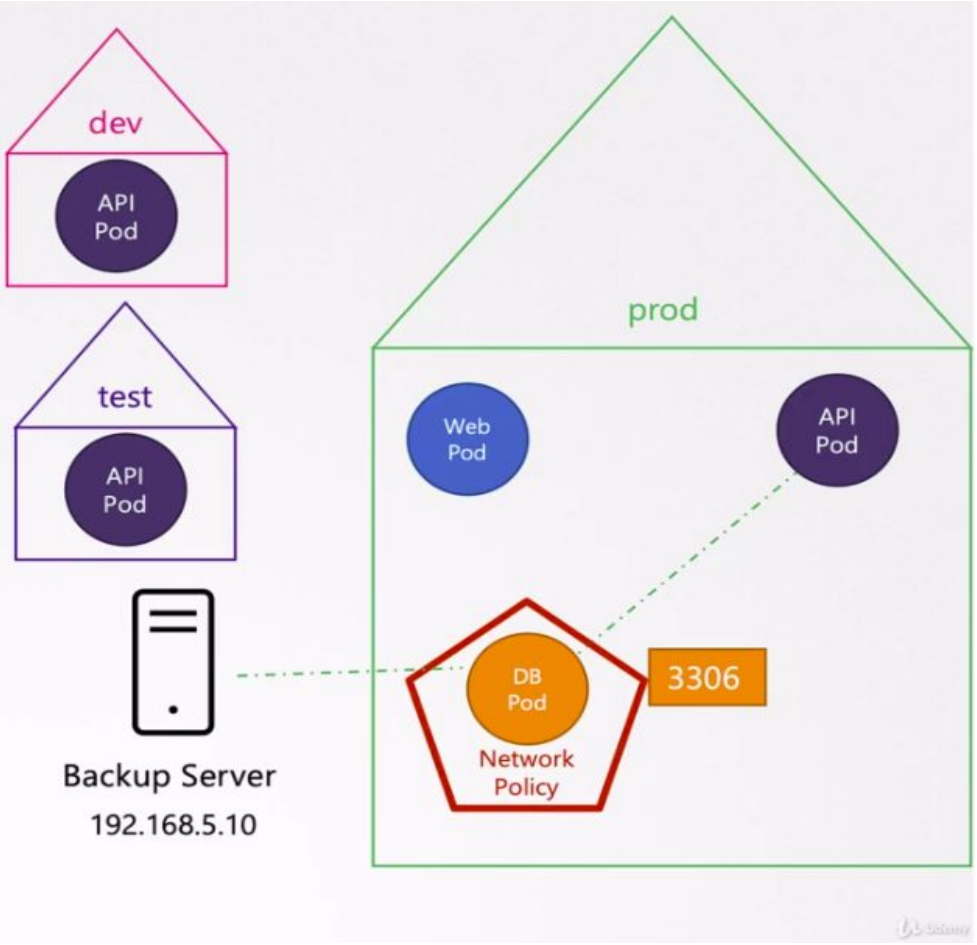
```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
      namespaceSelector:
        matchLabels:
          name: prod
    ports:
    - protocol: TCP
      port: 3306
```



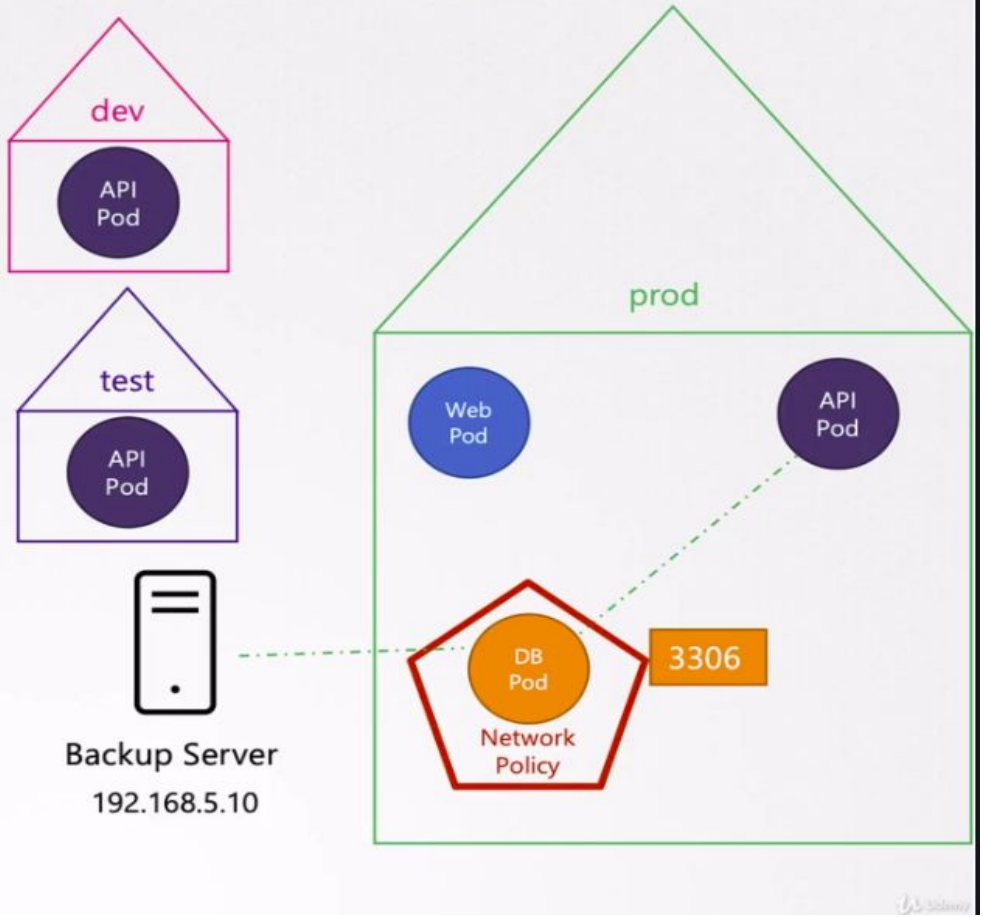
```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-policy
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
      namespaceSelector:
        matchLabels:
          name: prod
    ports:
    - protocol: TCP
      port: 3306
```



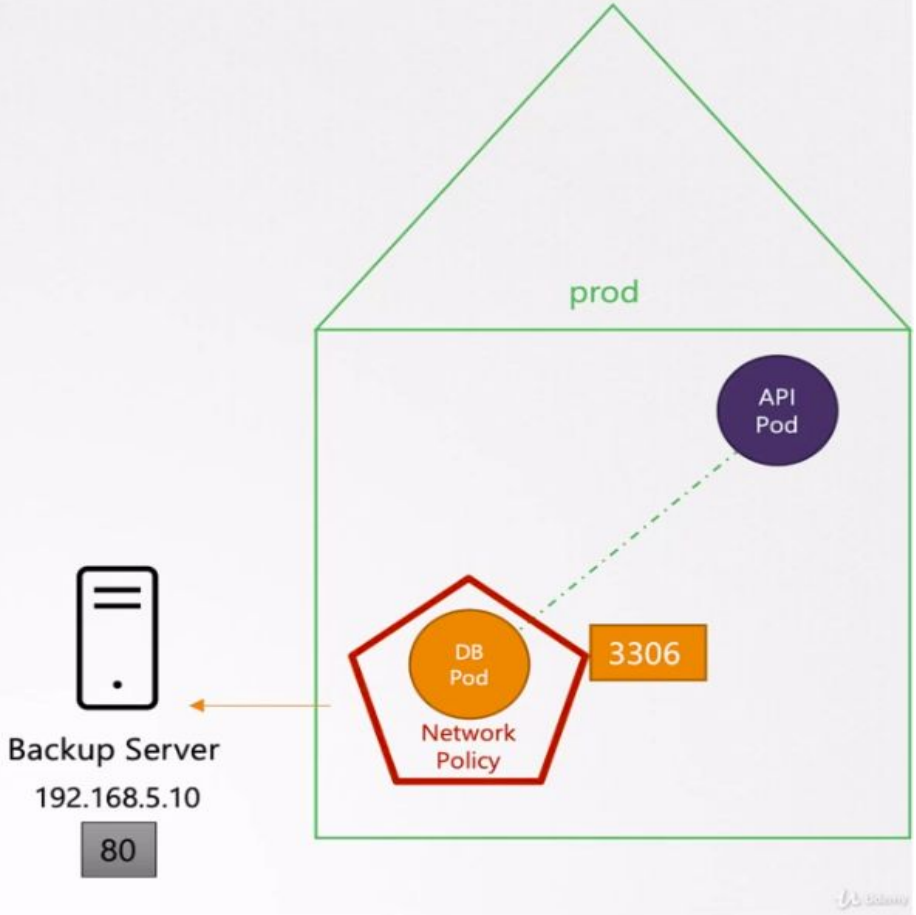
```yaml
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
      namespaceSelector:
        matchLabels:
          name: prod
    - ipBlock:
        cidr: 192.168.5.10/32
    ports:
    - protocol: TCP
      port: 3306
```



```yaml
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
    - namespaceSelector:
        matchLabels:
          name: prod
    - ipBlock:
        cidr: 192.168.5.10/32
    ports:
    - protocol: TCP
      port: 3306
```



```yaml
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          name: api-pod
    ports:
    - protocol: TCP
      port: 3306
  egress:
  - to:
    - ipBlock:
        cidr: 192.168.5.10/32
    ports:
    - protocol: TCP
      port: 80
```



Note

Solutions that Support Network Policies:

- Kube-router
- Calico
- Romana
- Weave-net

Solutions that DO NOT Support Network Policies:

- Flannel

DNS

For the Absolute Beginners


```
ping 192.168.1.11
```

```
Reply from 192.168.1.11: bytes=32 time=4ms TTL=117
Reply from 192.168.1.11: bytes=32 time=4ms TTL=117
```

```
ping db
```

```
ping: unknown host db
```


```
ping db
ping: unknown host db
```

```
hostname
host-2
```

```
cat >> /etc/hosts
192.168.1.11        db
```

```
ping db
PING db (192.168.1.11) 56(84) bytes of data.
64 bytes from db (192.168.1.11): icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from db (192.168.1.11): icmp_seq=2 ttl=64 time=0.079 ms
```

KODEKLOUD


```
cat >> /etc/hosts
192.168.1.11        db
192.168.1.11        www.google.com
```

```
hostname
host-2
```

```
ping db
PING db (192.168.1.11) 56(84) bytes of data.
64 bytes from db (192.168.1.11): icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from db (192.168.1.11): icmp_seq=2 ttl=64 time=0.079 ms
```

```
ping www.google.com
PING www.google.com (192.168.1.11) 56(84) bytes of data.
64 bytes from www.google.com (192.168.1.11): icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from www.google.com (192.168.1.11): icmp_seq=2 ttl=64 time=0.079 ms
```

Name Resolution


```
cat >> /etc/hosts
192.168.1.11        db
192.168.1.11        www.google.com
```

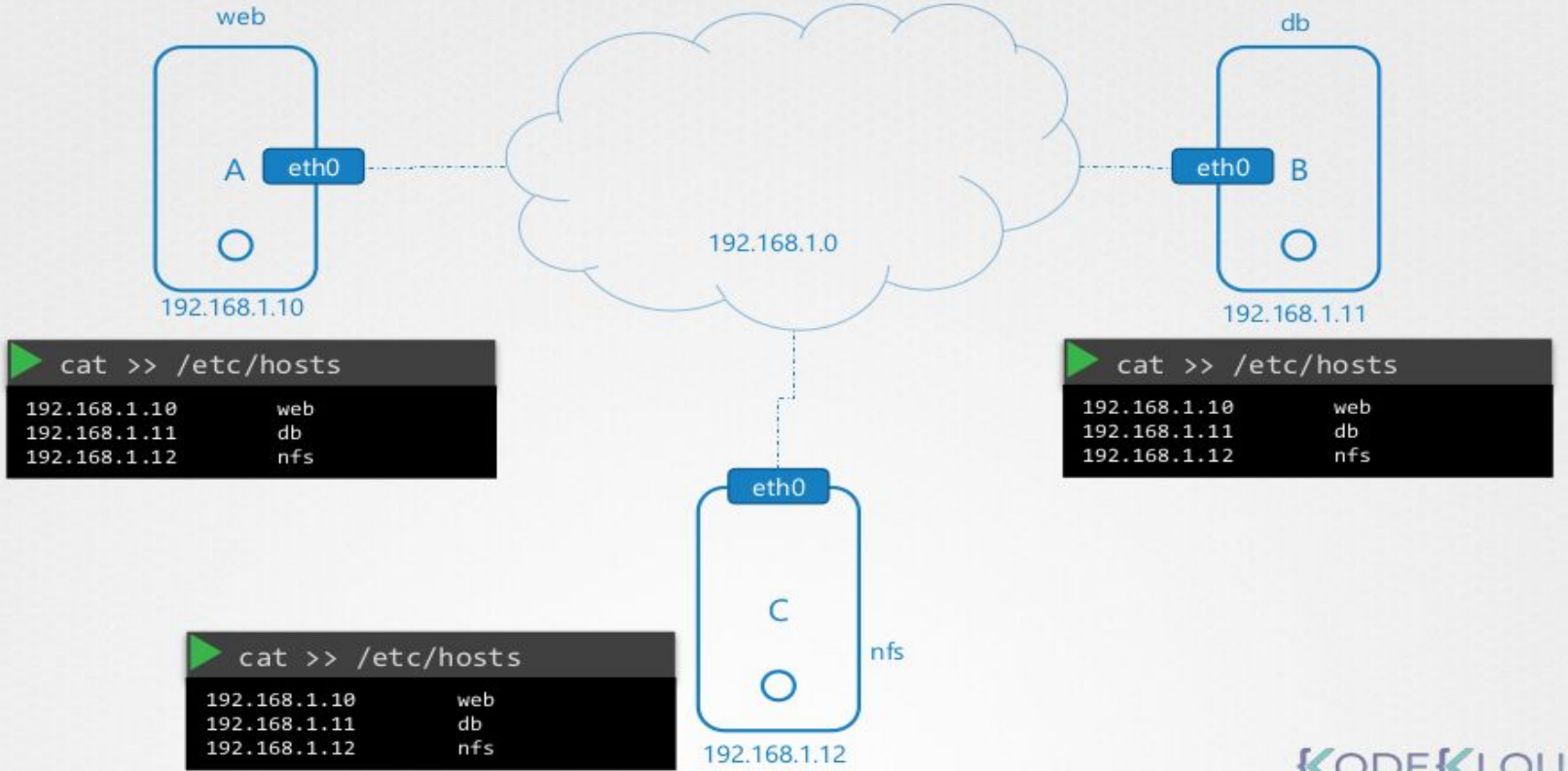
```
ping db
```

```
ssh db
```

```
curl http://www.google.com
```

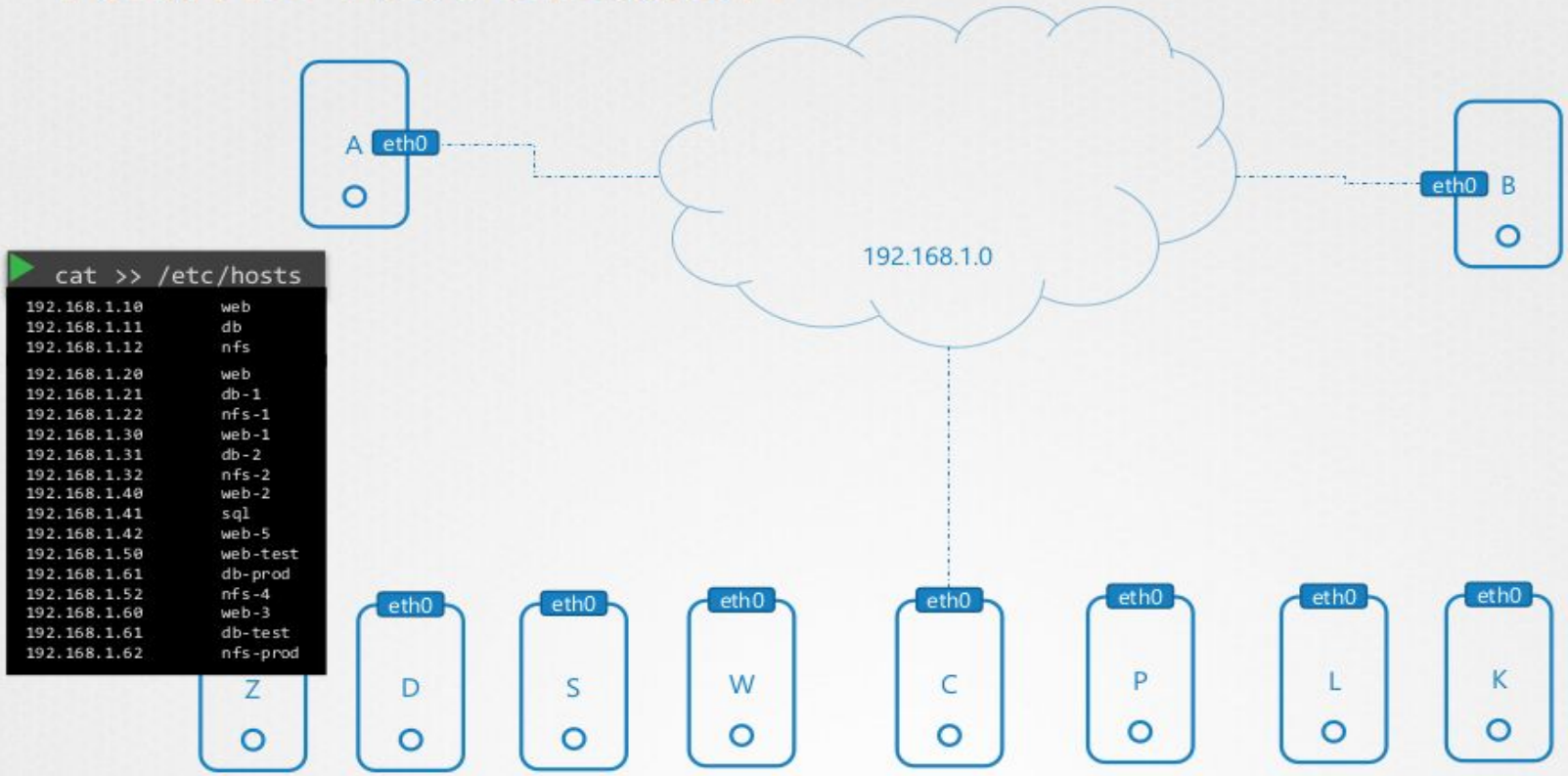
```
hostname
host-2
```

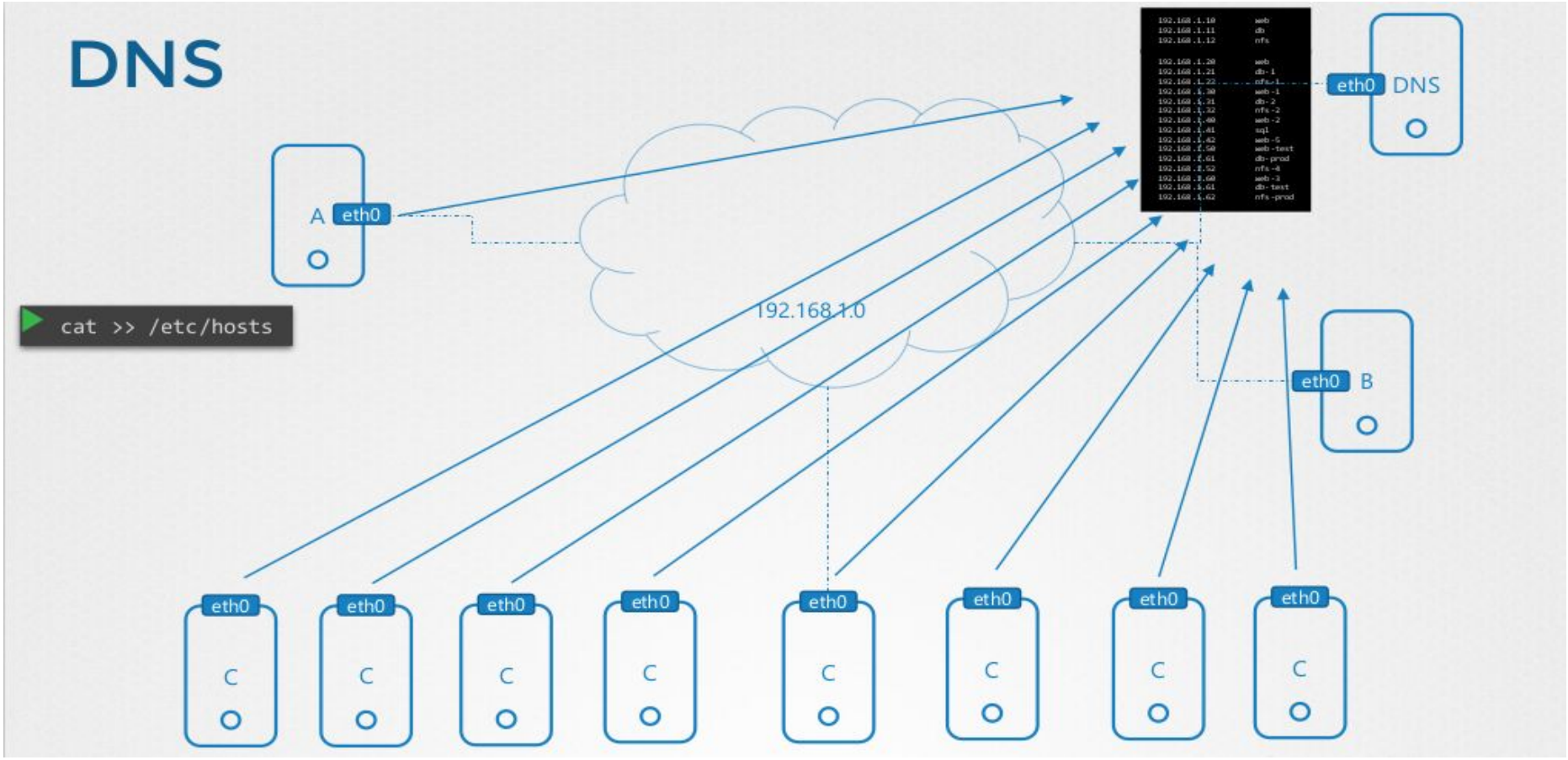
Name Resolution



KODEKLOUD

Name Resolution




DNS
192.168.1.10 web
192.168.1.11 db
192.168.1.12 nfs
192.168.1.20 web
192.168.1.21 db-1
192.168.1.22 nfs-1
192.168.1.30 web-1
192.168.1.31 db-2
192.168.1.32 nfs-2
192.168.1.40 web-2
192.168.1.41 sql
192.168.1.42 web-5
192.168.1.50 web-test
192.168.1.51 db-prod
192.168.1.52 nfs-4
192.168.1.60 web-3
192.168.1.61 db-test
192.168.1.62 nfs-prod
eth0 DNS
A eth0
cat >> /etc/hosts
192.168.1.0
eth0 B
eth0 C
eth0 C
eth0 C
eth0 C
eth0 C
eth0 C
eth0 C
eth0 C

DNS


```
cat /etc/resolv.conf
nameserver      192.168.1.100
```

```
ping db
PING db (192.168.1.11) 56(84) bytes of data.
64 bytes from db (192.168.1.11): icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from db (192.168.1.11): icmp_seq=2 ttl=64 time=0.079 ms
```

```
cat >> /etc/hosts
192.168.1.115       test
```

```
ping test
PING test (192.168.1.115) 56(84) bytes of data.
64 bytes from test (192.168.1.115): icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from test (192.168.1.115): icmp_seq=2 ttl=64 time=0.079 ms
```

KODEKLOUD

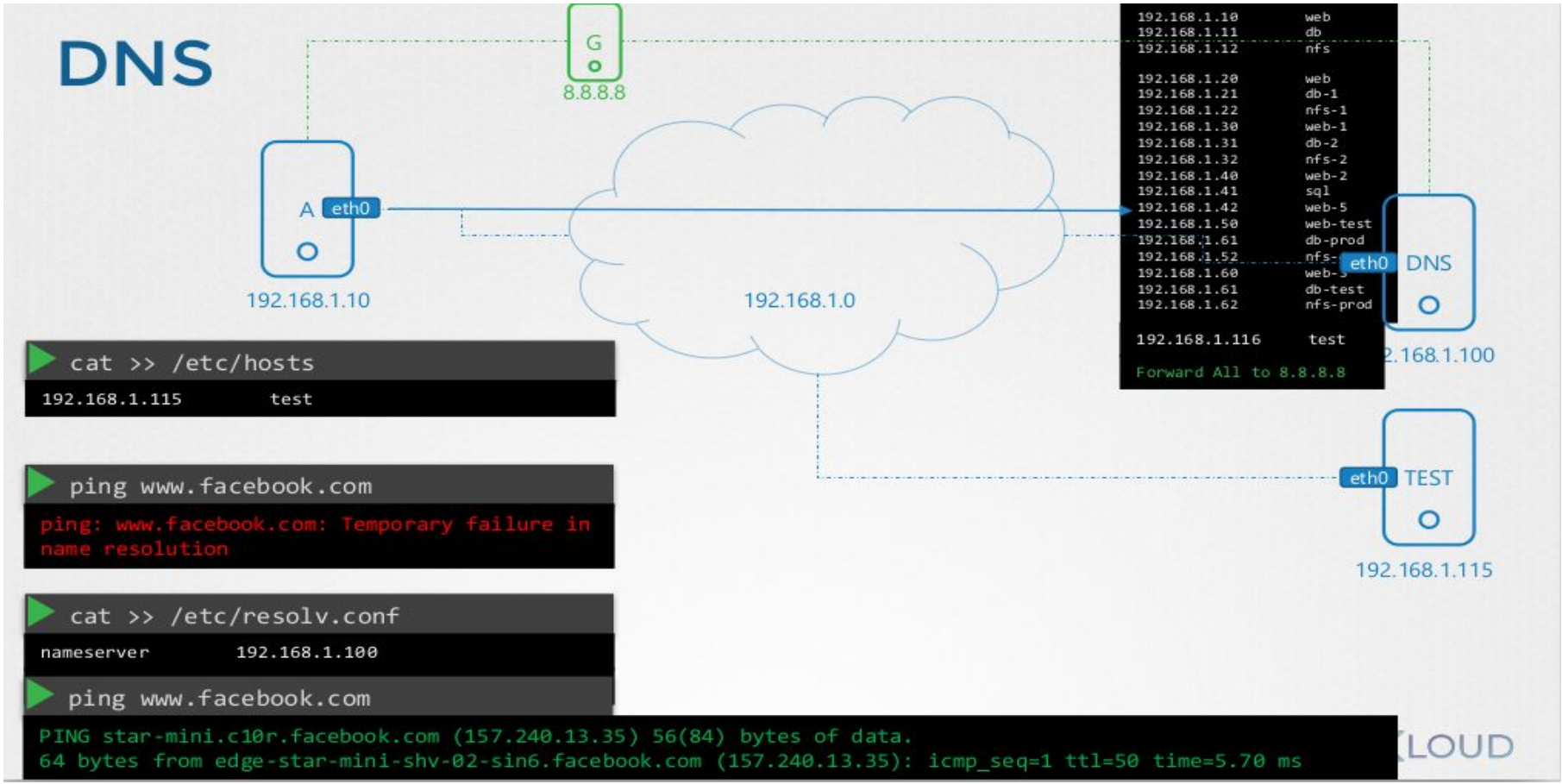
DNS


```
192.168.1.10        web
192.168.1.11        db
192.168.1.12        nfs

192.168.1.20        web
192.168.1.21        db-1
192.168.1.22        nfs-1
192.168.1.30        web-1
192.168.1.31        db-2
192.168.1.32        nfs-2
192.168.1.40        web-2
192.168.1.41        sql
192.168.1.42        web-5
192.168.1.50        web-test
192.168.1.61        db-prod
192.168.1.52        nfs-
192.168.1.60        web-
192.168.1.61        db-test
192.168.1.62        nfs-prod

192.168.1.116       test
```


```
cat >> /etc/hosts
192.168.1.115       test
```

```
cat /etc/nsswitch.conf
…
hosts:          files dns
…
```

DNS




```
cat >> /etc/hosts
192.168.1.115        test
```

```
ping www.facebook.com
ping: www.facebook.com: Temporary failure in
name resolution
```

```
cat >> /etc/resolv.conf
nameserver        192.168.1.100
```

```
ping www.facebook.com
PING star-mini.c10r.facebook.com (157.240.13.35) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-02-sin6.facebook.com (157.240.13.35): icmp_seq=1 ttl=50 time=5.70 ms
```

Domain Names

wwwkubernetes.io

www.codepen.io

www.facebook.com

www.un.org

www.mit.edu

www.google.com

www.behance.net

www.speedtest.net

www.stanford.edu

www.care.org

Domain Names

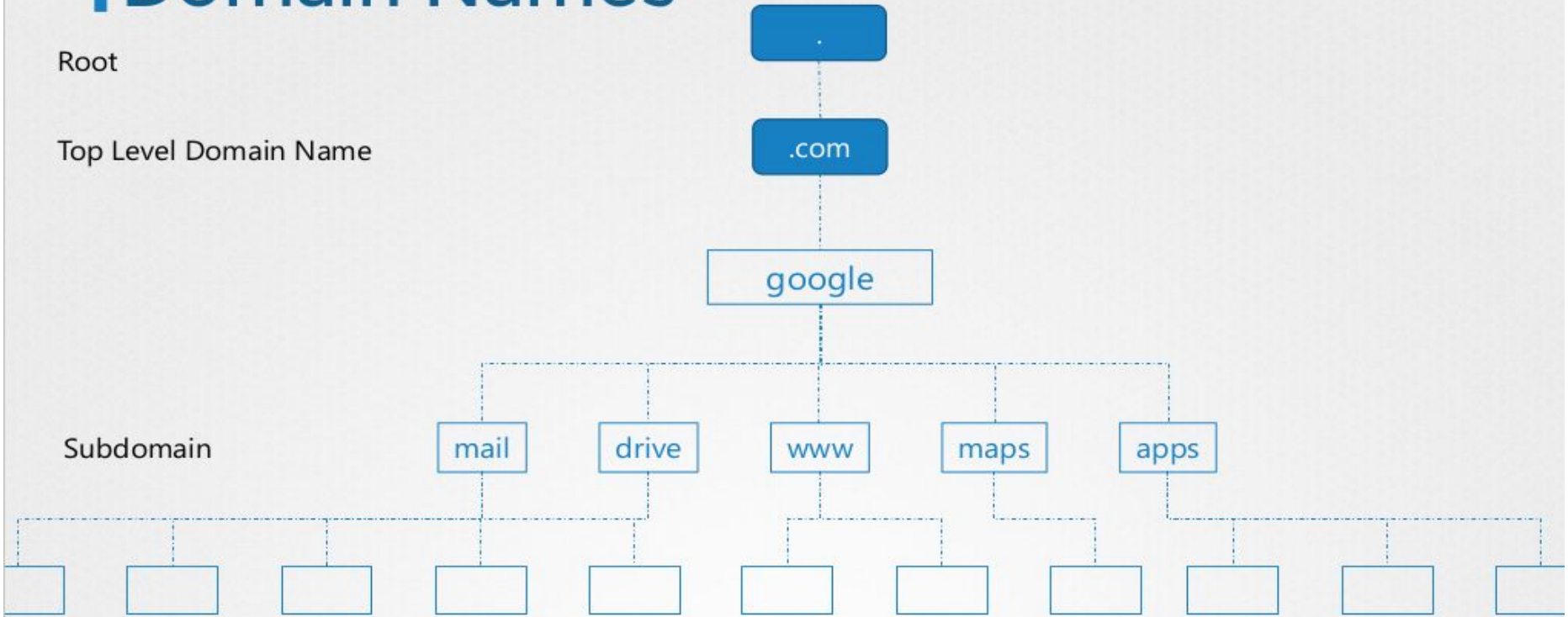
.com	.net	.edu	.org	.io
www.google	www.behance	www.stanford	www.care	www.kubernetes
www.facebook	www.speedtest	www.mit	www.un	www.codepen

Domain Names

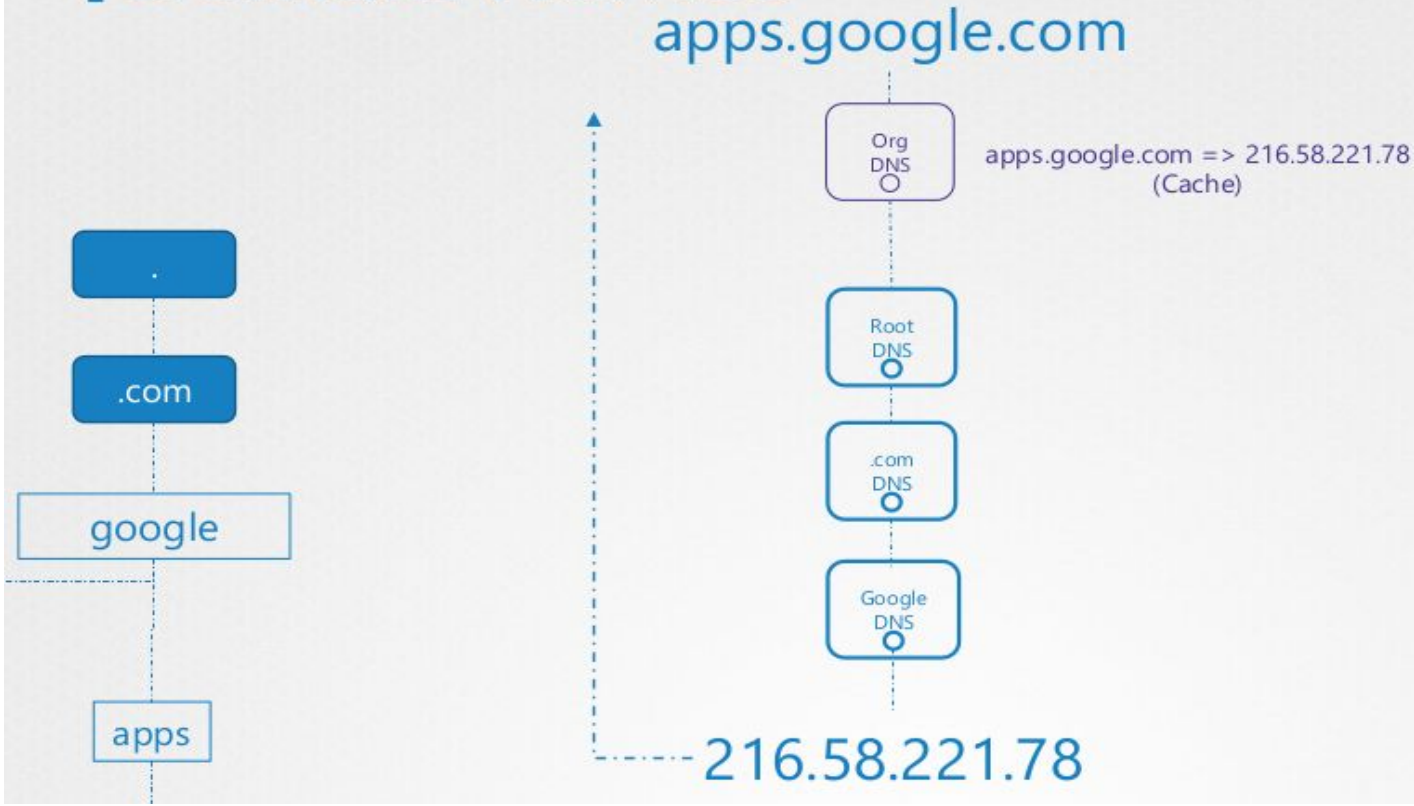
Root

Top Level Domain Name

Subdomain



Domain Names



Domain Names

Search Domain


```
192.168.1.10        web.mycompany.com
192.168.1.11        db.mycompany.com
192.168.1.12        nfs.mycompany.com
192.168.1.13        web-1.mycompany.com
192.168.1.14        sql.mycompany.com
```

mycompany.com

nfs | web | mail | drive | www | pay | hr | sql

```
cat >> /etc/resolv.conf
nameserver      192.168.1.100
search          mycompany.com  prod.mycompany.com
```

```
ping web
PING web (192.168.1.10) 56(84) bytes of data.
64 bytes from web (192.168.1.10): icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from web (192.168.1.10): icmp_seq=2 ttl=64 time=0.079 ms
```

```
ping web
PING web.mycompany.com (192.168.1.10) 56(84) bytes of data.
64 bytes from web.mycompany.com (192.168.1.10): … time=0.052 ms
64 bytes from web.mycompany.com (192.168.1.10): … time=0.079 ms
```

```
ping web
: web: Temporary failure in name resolution
```

```
ping web.mycompany.com
PING web.mycompany.com (192.168.1.10) 56(84) bytes of data.
64 bytes from web.mycompany.com (192.168.1.10): ttl=64 time=0.052 ms
```

```
ping web.mycompany.com
PING web.mycompany.com (192.168.1.10) 56(84) bytes of data.
om web.mycompany.com (192.168.1.10): ttl=64 time=0.052 ms
```

Record Types

A	web-server	192.168.1.1
AAAA	web-server	2001:0db8:85a3:0000:0000:8a2e:0370:7334
CNAME	food.web-server	eat.web-server, hungry.web-server

nslookup

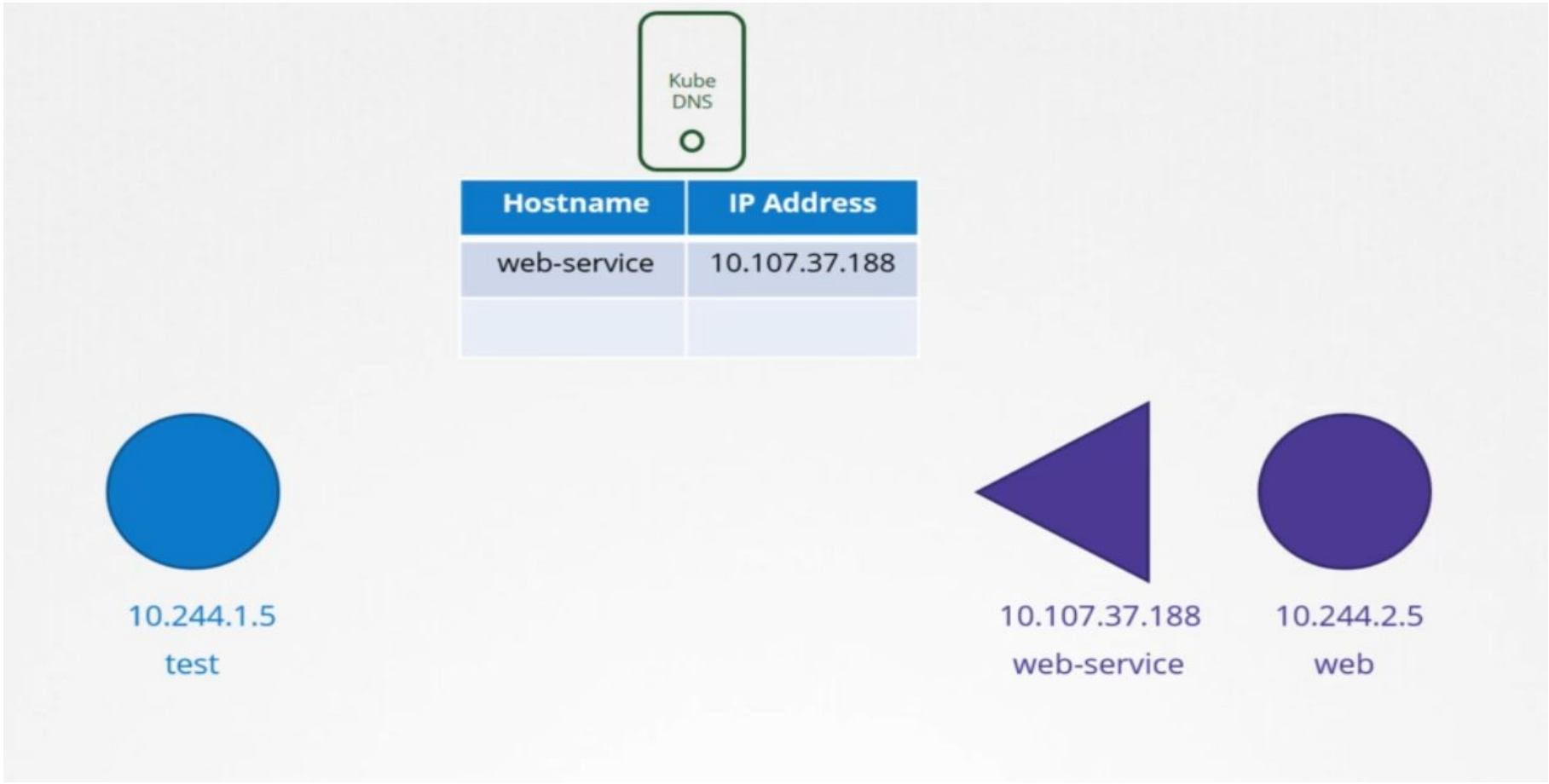
```
nslookup www.google.com
```

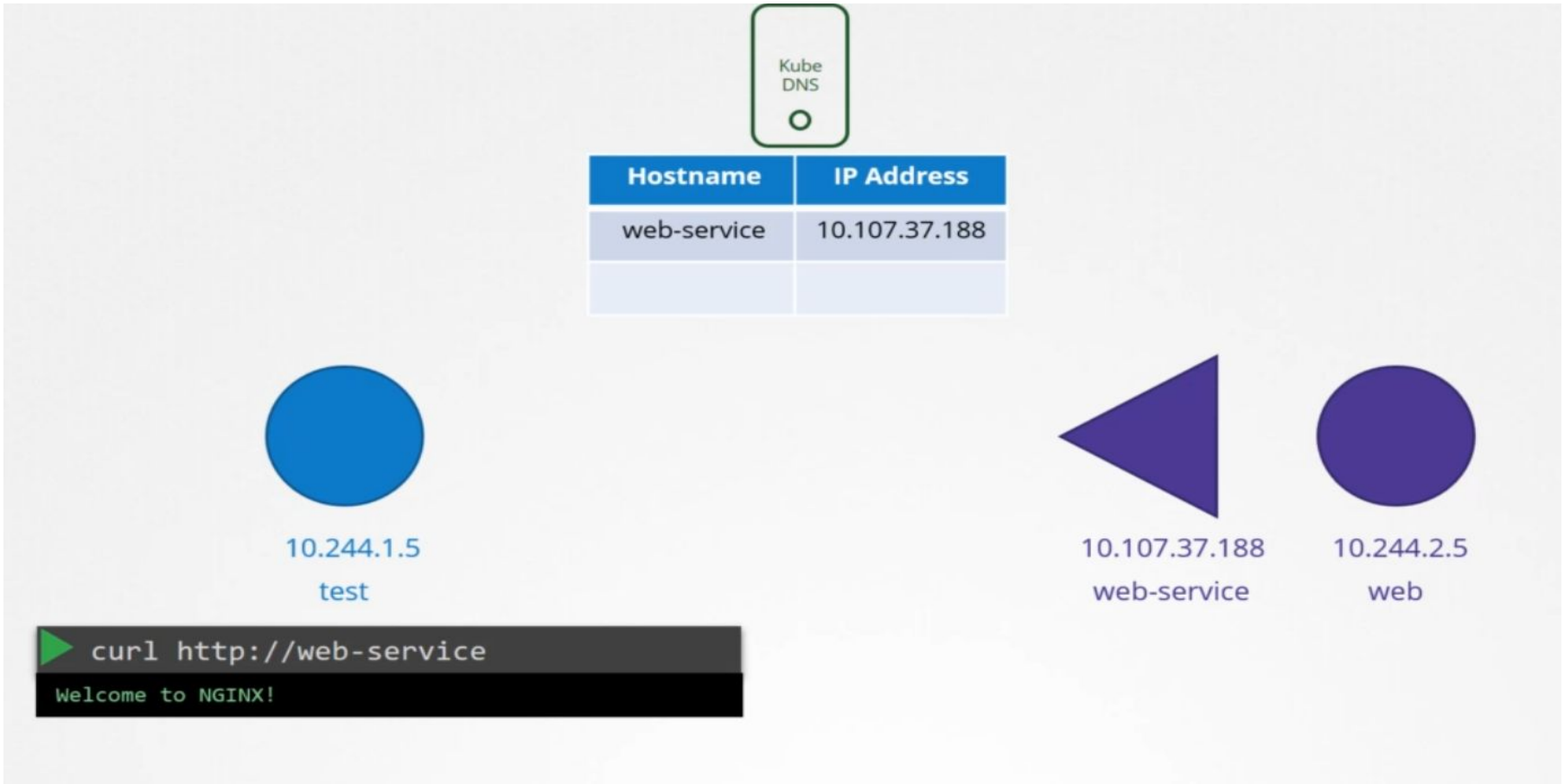
```
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.132
```

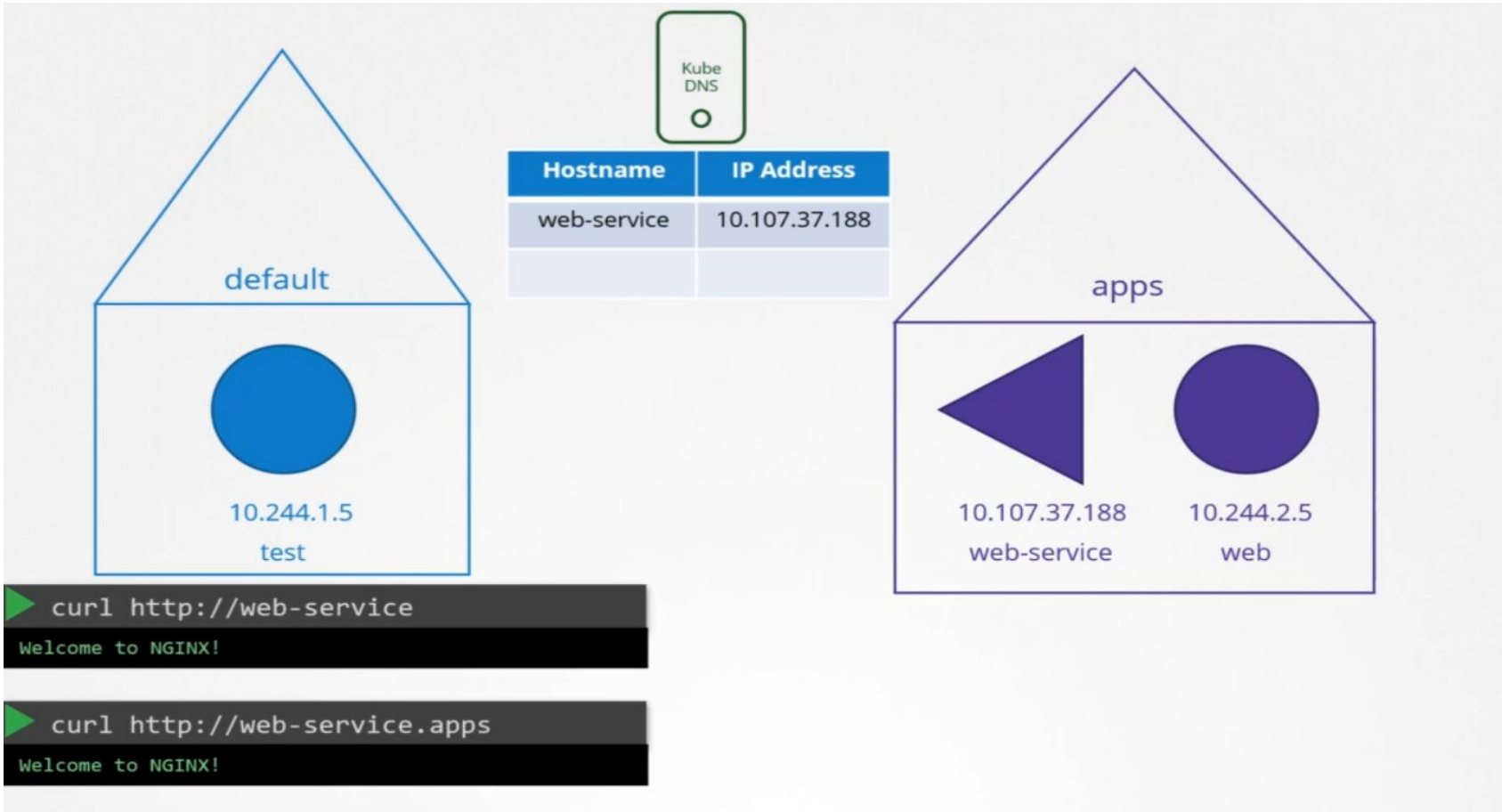
dig

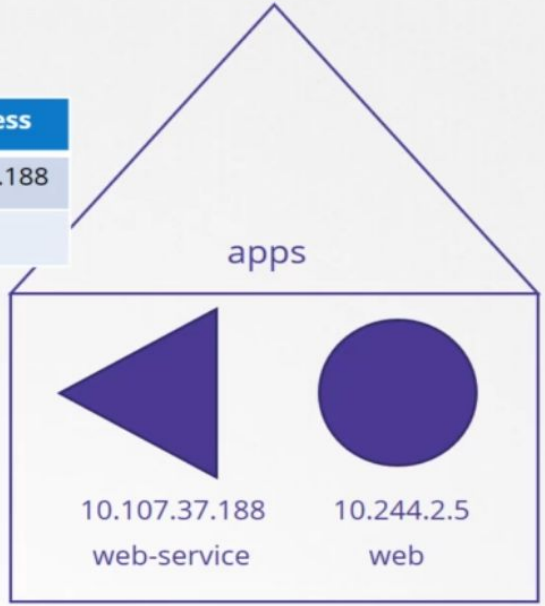
```
dig www.google.com
```

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28065
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         245     IN      A       64.233.177.103
www.google.com.         245     IN      A       64.233.177.105
www.google.com.         245     IN      A       64.233.177.147
www.google.com.         245     IN      A       64.233.177.106
www.google.com.         245     IN      A       64.233.177.104
www.google.com.         245     IN      A       64.233.177.99

;; Query time: 5 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sun Mar 24 04:34:33 UTC 2019
;; MSG SIZE  rcvd: 139
```


Kube
DNS
Hostname
IP Address
web-service
10.107.37.188
10.244.1.5
test
10.107.37.188
web-service
10.244.2.5
web


Kube
DNS
Hostname
IP Address
web-service
10.107.37.188
10.244.1.5
test
10.107.37.188
web-service
10.244.2.5
web
curl http://web-service
Welcome to NGINX!


Kube
DNS
Hostname
IP Address
web-service
10.107.37.188
default
10.244.1.5
test
apps
10.107.37.188
web-service
10.244.2.5
web
curl http://web-service
Welcome to NGINX!
curl http://web-service.apps
Welcome to NGINX!



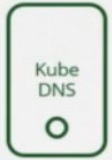
Kube DNS

Hostname	Namespace	Type	Root	IP Address
web-service	apps	svc	cluster.local	10.107.37.188

```
curl http://web-service.apps
Welcome to NGINX!
```

```
curl http://web-service.apps.svc
Welcome to NGINX!
```

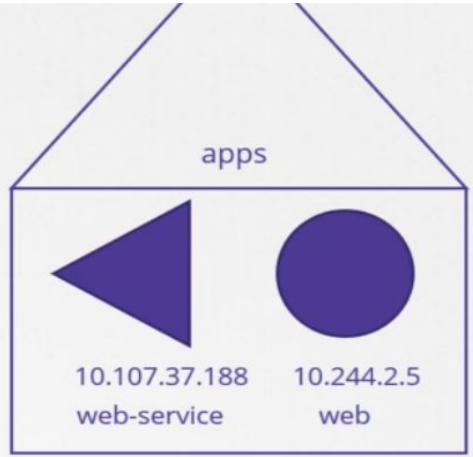
```
curl http://web-service.apps.svc.cluster.local
Welcome to NGINX!
```

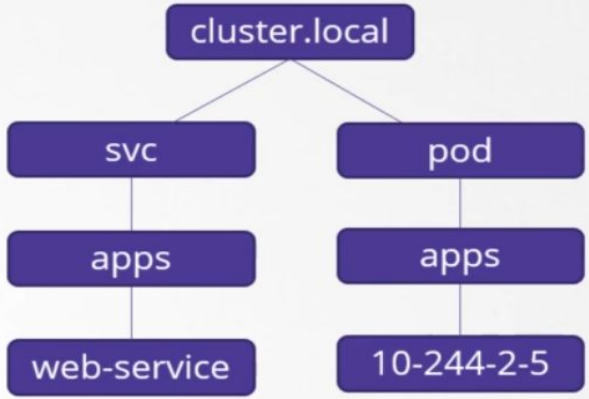


Hostname	Namespace	Type	Root	IP Address
web-service	apps	svc	cluster.local	10.107.37.188
10-244-2-5	apps	pod	cluster.local	10.244.2.5

10-244-2-5 ⟵ 10.244.2.5

```
curl http://10-244-2-5.apps.pod.cluster.local
Welcome to NGINX!
```





References:

- https://www.udemy.com/course/certified-kubernetes-administrator-with-practice-tests
- https://www.udemy.com/course/certified-kubernetes-application-developer
- https://kubernetes.io/docs