Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
Institute of Computer Technology

Name: Tushar Panchal

En.No: 21162101014

Sub: CS(Cloud Security)

Branch: CBA

Batch:71

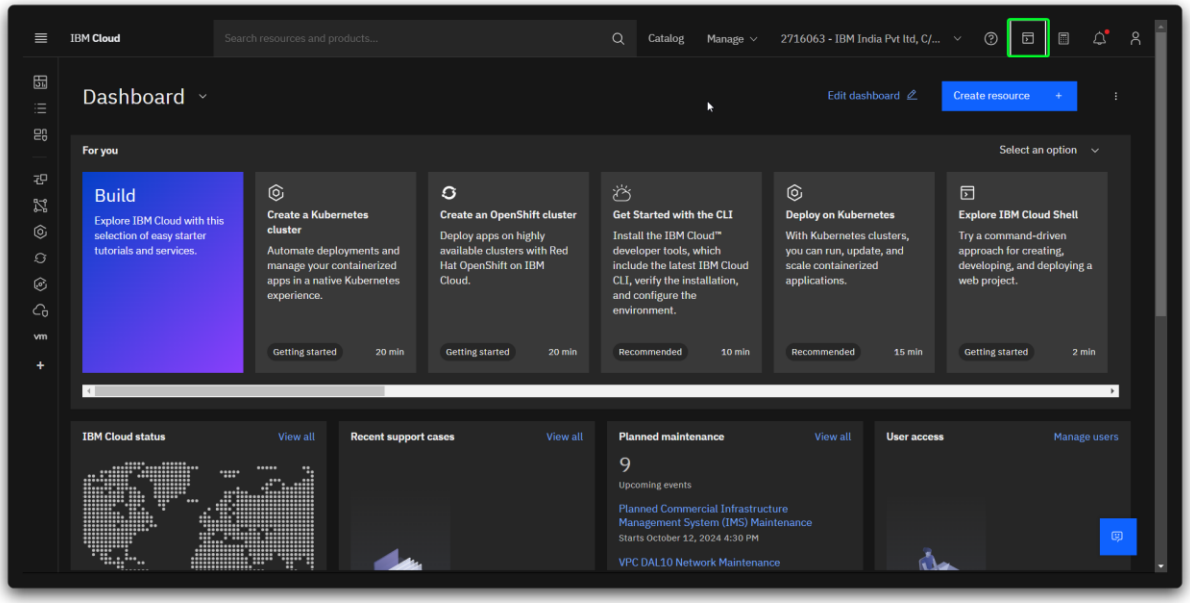
----------PRACTICAL 09----------

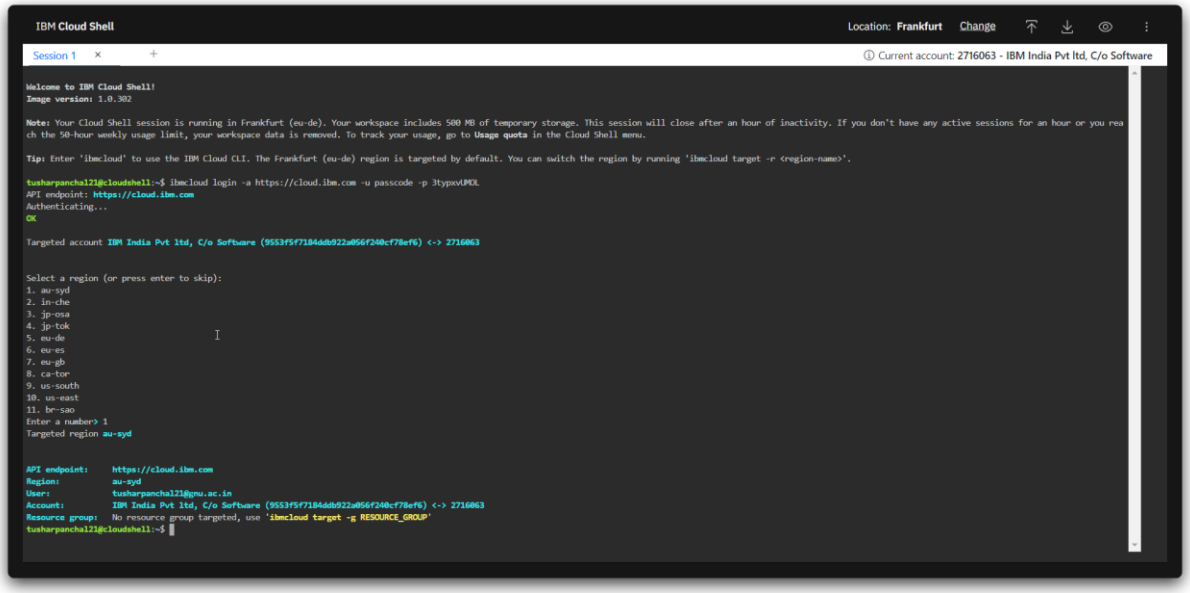
To restrict network access to an application deployed on Kubernetes, you can use Network Policies. A Network Policy resource defines how groups of pods are allowed to communicate with each other and with other network endpoints. You can configure network policies to control both ingress and egress traffic, limiting access to your application only to the necessary services or pods.

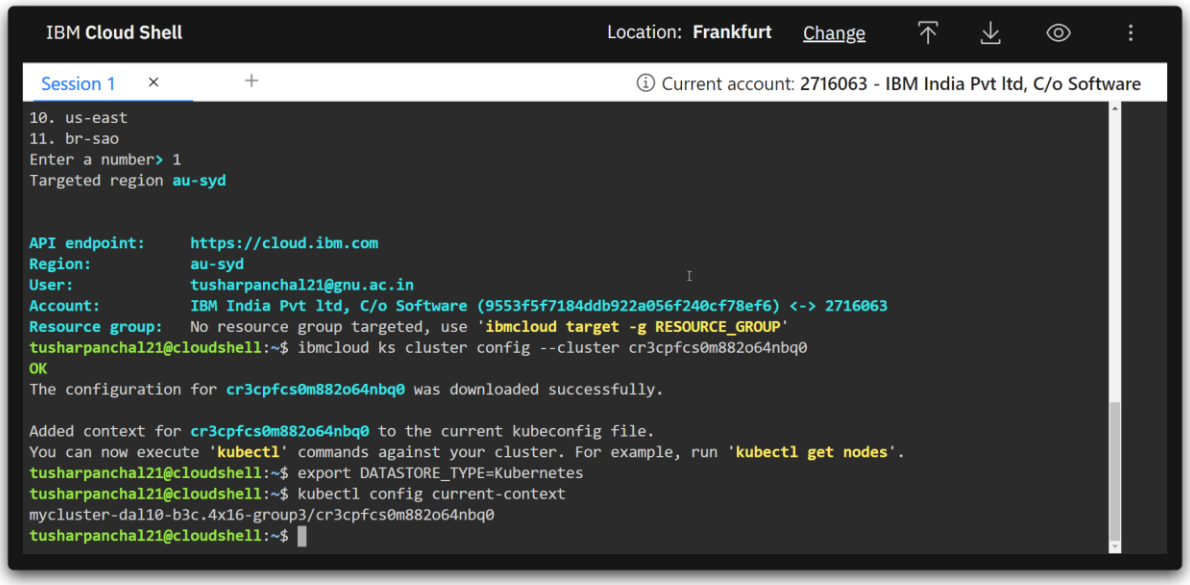
Tasks:

- **Review any default network policies applied by Calico or Kubernetes that might affect traffic in your cluster.**
- **Add new network policies to meet the security requirements outlined above.**
- **After testing the policies, clean up any temporary or redundant policies that are no longer needed.**

» Login to IBM Cloud CLI / Shell and configure Kubernetes and export datastore type env variable with value Kubernetes


IBM Cloud
Search resources and products...
Catalog
Manage
2716063 - IBM India Pvt ltd, C/...
Dashboard
Edit dashboard
Create resource
For you
Select an option
Build
Explore IBM Cloud with this selection of easy starter tutorials and services.
Create a Kubernetes cluster
Automate deployments and manage your containerized apps in a native Kubernetes experience.
Getting started
20 min
Create an OpenShift cluster
Deploy apps on highly available clusters with Red Hat OpenShift on IBM Cloud.
Getting started
20 min
Get Started with the CLI
Install the IBM Cloud™ developer tools, which include the latest IBM Cloud CLI, verify the installation, and configure the environment.
Recommended
10 min
Deploy on Kubernetes
With Kubernetes clusters, you can run, update, and scale containerized applications.
Recommended
15 min
Explore IBM Cloud Shell
Try a command-driven approach for creating, developing, and deploying a web project.
Getting started
2 min
IBM Cloud status
View all
Recent support cases
View all
Planned maintenance
View all
9
Upcoming events
Planned Commercial Infrastructure Management System (IMS) Maintenance
Starts October 12, 2024 4:30 PM
VPC DAL10 Network Maintenance
User access
Manage users


IBM Cloud Shell
Location: Frankfurt
Change
Session 1
Current account: 2716063 - IBM India Pvt ltd, C/o Software
Welcome to IBM Cloud Shell!
Image version: 1.0.302
Note: Your Cloud Shell session is running in Frankfurt (eu-de). Your workspace includes 500 MB of temporary storage. This session will close after an hour of inactivity. If you don't have any active sessions for an hour or you reach the 50-hour weekly usage limit, your workspace data is removed. To track your usage, go to Usage quota in the Cloud Shell menu.
Tip: Enter 'ibmcloud' to use the IBM Cloud CLI. The Frankfurt (eu-de) region is targeted by default. You can switch the region by running 'ibmcloud target -r <region-name>'.
tusharpanchal21@cloudshell:~$ ibmcloud login -a https://cloud.ibm.com -u passcode -p 3typxvUMOL
API endpoint: https://cloud.ibm.com
Authenticating...
OK
Targeted account IBM India Pvt ltd, C/o Software (9553f5f7184ddb922a056f240cf78ef6) <-> 2716063
Select a region (or press enter to skip):
1. au-syd
2. in-che
3. jp-osa
4. jp-tok
5. eu-de
6. eu-es
7. eu-gb
8. ca-tor
9. us-south
10. us-east
11. br-sao
Enter a number> 1
Targeted region au-syd
API endpoint: https://cloud.ibm.com
Region: au-syd
User: tusharpanchal21@gnu.ac.in
Account: IBM India Pvt ltd, C/o Software (9553f5f7184ddb922a056f240cf78ef6) <-> 2716063
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
tusharpanchal21@cloudshell:~$


IBM Cloud Shell
Location: Frankfurt
Change
Session 1
Current account: 2716063 - IBM India Pvt ltd, C/o Software
10. us-east
11. br-sao
Enter a number> 1
Targeted region au-syd
API endpoint: https://cloud.ibm.com
Region: au-syd
User: tusharpanchal21@gnu.ac.in
Account: IBM India Pvt ltd, C/o Software (9553f5f7184ddb922a056f240cf78ef6) <-> 2716063
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
tusharpanchal21@cloudshell:~$ ibmcloud ks cluster config --cluster cr3cpfcs0m882o64nbq0
OK
The configuration for cr3cpfcs0m882o64nbq0 was downloaded successfully.
Added context for cr3cpfcs0m882o64nbq0 to the current kubeconfig file.
You can now execute 'kubectl' commands against your cluster. For example, run 'kubectl get nodes'.
tusharpanchal21@cloudshell:~$ export DATASTORE_TYPE=Kubernetes
tusharpanchal21@cloudshell:~$ kubectl config current-context
mycluster-dal10-b3c.4x16-group3/cr3cpfcs0m882o64nbq0
tusharpanchal21@cloudshell:~$

- Configure k8s with admin access to network part

```
tusharpanchal21@cloudshell:~$ ibmcloud ks cluster config --cluster cr3cpfcs0m882o64nbq0 --admin --network
OK
The configuration for cr3cpfcs0m882o64nbq0 was downloaded successfully.

Network Config:

/tmp/ic/-1/.bluemix/plugins/container-service/clusters/mycluster-dal10-b3c.4x16-group3-cr3cpfcs0m882o64nbq0-admin/calicoctl.cfg


Added context for cr3cpfcs0m882o64nbq0 to the current kubeconfig file.
You can now execute 'kubectl' commands against your cluster. For example, run 'kubectl get nodes'.
tusharpanchal21@cloudshell:~$
```

```
tusharpanchal21@cloudshell:~$ alias calicoctl='calicoctl --allow-version-mismatch'
tusharpanchal21@cloudshell:~$ calicoctl get nodes
NAME
10.210.8.231
10.210.8.252
```

- If in IBM shell, copy the calico config to /etc directory to configure calicoctl

```
tusharpanchal21@cloudshell:~$ cp /tmp/ic/-1/.bluemix/plugins/container-service/clusters/mycluster-dal10-b3c.4x16-group
3-cr3cpfcs0m882o64nbq0-admin/calicoctl.cfg /etc/calico
```

- Try getting hostendpoints, Global Network Policies and other Network Policies using calicoctl

```
tusharpanchal21@cloudshell:~$ calicoctl get hostendpoint -o yaml
apiVersion: projectcalico.org/v3
items:
- apiVersion: projectcalico.org/v3
  kind: HostEndpoint
  metadata:
    annotations:
      kubectl.kubernetes.io/last-applied-configuration: |
        {"apiVersion":"crd.projectcalico.org/v1","kind":"HostEndpoint","metadata":{"annotations":{},"creationTimestamp":null,"labels":{"add
onmanager.kubernetes.io/mode":"Reconcile","arch":"amd64","beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/instance-type":"u3c.2x4.encr
ypted","beta.kubernetes.io/os":"linux","failure-domain.beta.kubernetes.io/region":"au-syd","failure-domain.beta.kubernetes.io/zone":"syd01"
,"ibm-cloud.kubernetes.io/encrypted-docker-data":"true","ibm-cloud.kubernetes.io/external-ip":"159.23.67.205","ibm-cloud.kubernetes.io/ha-w
orker":"true","ibm-cloud.kubernetes.io/iaas-provider":"softlayer","ibm-cloud.kubernetes.io/interface-name":"eth0","ibm-cloud.kubernetes.io/
internal-ip":"10.210.8.252","ibm-cloud.kubernetes.io/machine-type":"u3c.2x4.encrypted","ibm-cloud.kubernetes.io/os":"UBUNTU_20_64","ibm-clo
ud.kubernetes.io/region":"au-syd","ibm-cloud.kubernetes.io/sgx-enabled":"false","ibm-cloud.kubernetes.io/worker-id":"kube-cr3cpfcs0m882o64n
bq0-myclusterda-default-00000170","ibm-cloud.kubernetes.io/worker-pool-id":"cr3cpfcs0m882o64nbq0-4d27b23","ibm-cloud.kubernetes.io/worker-p
ool-name":"default","ibm-cloud.kubernetes.io/worker-version":"1.30.3_1533","ibm-cloud.kubernetes.io/zone":"syd01","ibm.role":"worker_privat
e","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"10.210.8.252","kubernetes.io/os":"linux","node.kubernetes.io/instance-type":"u3c.
2x4.encrypted","privateVLAN":"3438703","publicVLAN":"3438701","topology.kubernetes.io/region":"au-syd","topology.kubernetes.io/zone":"syd01
"},"name":"kube-cr3cpfcs0m882o64nbq0-myclusterda-default-00000170-worker-private"},"spec":{"expectedIPs":["10.210.8.252"],"interfaceName":"
eth0","node":"10.210.8.252"}}
    creationTimestamp: "2024-08-22T05:45:11Z"
    labels:
      addonmanager.kubernetes.io/mode: Reconcile
      arch: amd64
```

```
      ibm-cloud.kubernetes.io/worker-id: kube-cr3cpfcs0m882o64nbq0-myclusterda-default-00000269
      ibm-cloud.kubernetes.io/worker-pool-id: cr3cpfcs0m882o64nbq0-4d27b23
      ibm-cloud.kubernetes.io/worker-pool-name: default
      ibm-cloud.kubernetes.io/worker-version: 1.30.5_1538
      ibm-cloud.kubernetes.io/zone: syd01
      ibm.role: worker_public
      kubernetes.io/arch: amd64
      kubernetes.io/hostname: 10.210.8.231
      kubernetes.io/os: linux
      node.kubernetes.io/instance-type: u3c.2x4.encrypted
      privateVLAN: "3438703"
      publicVLAN: "3438701"
      topology.kubernetes.io/region: au-syd
      topology.kubernetes.io/zone: syd01
    name: kube-cr3cpfcs0m882o64nbq0-myclusterda-default-00000269-worker-public
    resourceVersion: "3623825"
    uid: 84de3006-d998-4cb0-a67a-3a29f5a1beba
  spec:
    expectedIPs:
    - 159.23.67.202
    interfaceName: eth1
    node: 10.210.8.231
kind: HostEndpointList
metadata:
  resourceVersion: "4838094"
tusharpanchal21@cloudshell:~$
```

```
tusharpanchal21@cloudshell:~$ calicoctl get hostendpoint -o wide
NAME                                                                NODE          INTERFACE   IPS             PROFILES
kube-cr3cpfcs0m882o64nbq0-myclusterda-default-00000170-worker-private   10.210.8.252   eth0        10.210.8.252
kube-cr3cpfcs0m882o64nbq0-myclusterda-default-00000170-worker-public    10.210.8.252   eth1        159.23.67.205
kube-cr3cpfcs0m882o64nbq0-myclusterda-default-00000269-worker-private   10.210.8.231   eth0        10.210.8.231
kube-cr3cpfcs0m882o64nbq0-myclusterda-default-00000269-worker-public    10.210.8.231   eth1        159.23.67.202
```

```
tusharpanchal21@cloudshell:~$ calicoctl get GlobalNetworkPolicy -o wide
NAME                        ORDER    SELECTOR
allow-all-outbound          1900     ibm.role == 'worker_public'
allow-all-private-default   100000   ibm.role == 'worker_private'
allow-icmp                  1500     ibm.role == 'worker_public'
allow-node-port-dnat        1500     ibm.role == 'worker_public'
allow-sys-mgmt              1950     ibm.role == 'worker_public'
allow-vrrp                  1500     ibm.role == 'worker_public'
```

```
tusharpanchal21@cloudshell:~$ calicoctl get GlobalNetworkPolicy -o yaml
apiVersion: projectcalico.org/v3
items:
- apiVersion: projectcalico.org/v3
  kind: GlobalNetworkPolicy
  metadata:
    creationTimestamp: "2024-08-22T05:39:03Z"
    name: allow-all-outbound
    resourceVersion: "661059"
    uid: 3ecd8d2a-a962-44ba-8e6d-418fd2848a9a
  spec:
    egress:
    - action: Allow
      destination: {}
      source: {}
    order: 1900
    selector: ibm.role == 'worker_public'
    types:
    - Egress
- apiVersion: projectcalico.org/v3
  kind: GlobalNetworkPolicy
  metadata:
    creationTimestamp: "2024-08-22T05:39:04Z"
    name: allow-all-private-default
    resourceVersion: "661071"
    uid: 8df9606e-d93e-4f67-a101-472ff67c7c4c
```

```
  kind: GlobalNetworkPolicy
  metadata:
    creationTimestamp: "2024-08-22T05:39:04Z"
    name: allow-vrrp
    resourceVersion: "661069"
    uid: b2351d70-5fb2-41db-8ce0-176ac2c71d2e
  spec:
    egress:
    - action: Allow
      destination: {}
      protocol: 112
      source: {}
    ingress:
    - action: Allow
      destination: {}
      protocol: 112
      source: {}
    order: 1500
    selector: ibm.role == 'worker_public'
    types:
    - Ingress
    - Egress
kind: GlobalNetworkPolicyList
metadata:
  resourceVersion: "4838574"
tusharpanchal21@cloudshell:~$
```

- Create namespace in k8s cluster and create and expose any pod like nginx

```
tusharpanchal21@cloudshell:~$ kubectl delete validatingwebhookconfiguration gatekeeper-validating-webhook-configuration
validatingwebhookconfiguration.admissionregistration.k8s.io "gatekeeper-validating-webhook-configuration" deleted
tusharpanchal21@cloudshell:~$ kubectl create ns advanced-policy-demo
namespace/advanced-policy-demo created
```

```
tusharpanchal21@cloudshell:~$ kubectl create ns advanced-policy-demo7
namespace/advanced-policy-demo7 created
```

```
tusharpanchal21@cloudshell:~$ kubectl create deployment --namespace=advanced-policy-demo7 nginx --image=nginx
deployment.apps/nginx created
```

```
tusharpanchal21@cloudshell:~$ kubectl expose --namespace=advanced-policy-demo7 deployment nginx --port=80
service/nginx exposed
```

```
tusharpanchal21@cloudshell:~$ kubectl get pods -n advanced-policy-demo7
NAME                     READY   STATUS    RESTARTS   AGE
nginx-bf5d5cf98-27prq    1/1     Running   0          3m37s
```

» Run pod container and check the egress and ingress traffic

```
tusharpanchal21@cloudshell:~$ kubectl run --namespace=advanced-policy-demo7 access --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
/ #
```

```
/ # wget --timeout=5 nginx -O tk
Connecting to nginx (172.21.47.105:80)
saving to 'tk'
tk                   100% |********************************************************************************|   615  0:00:00 ETA
'tk' saved
/ # cat tk
```

```
/ # cat tk
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ #
```

» Now create global network policy to deny all traffic

```
calicoctl create -f - <<EOF
apiVersion: projectcalico.org/v3
kind: GlobalNetworkPolicy
metadata:
  name: default-deny
spec:
  selector: projectcalico.org/namespace != "kube-system"
  types:
    - Ingress
    - Egress
EOF
```

```
tusharpanchal21@cloudshell:~$ calicoctl create -f -<<EOF
> apiVersion: projectcalico.org/v3
> kind: GlobalNetworkPolicy
> metadata:
>       name: default-deny
> spec:
>       selector: projectcalico.org/namespace != "kube-system"
>       types:
>       - Ingress
>       - Egress
> EOF
Successfully created 1 'GlobalNetworkPolicy' resource(s)
tusharpanchal21@cloudshell:~$
```

```
tusharpanchal21@cloudshell:~$ kubectl run --namespace=advanced-policy-demo7 access --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ # wget --timeout=2 google.com
wget: bad address 'google.com'
/ #
```

» Now create a policy to allow our pod's egress traffic

```
calicoctl create -f - <<EOF
apiVersion: projectcalico.org/v3
kind: NetworkPolicy
metadata:
  name: allow-busybox-egress
  namespace: advanced-policy-demo7
spec:
  selector: run == 'access'
  types:
    - Egress
  egress:
```

```
    - action: Allow
EOF
```

```
tusharpanchal21@cloudshell:~$ calicoctl create -f - <<EOF
> apiVersion: projectcalico.org/v3
> kind: NetworkPolicy
> metadata:
>   name: allow-busybox-egress
>   namespace: advanced-policy-demo7
> spec:
>   selector: run == 'access'
>   types:
>     - Egress
>   egress:
>     - action: Allow
> EOF
Failed to create 'NetworkPolicy' resource: [resource already exists: NetworkPolicy(advanced-policy-demo7/default.allow-busybox-egress)]
```

```
tusharpanchal21@cloudshell:~$ kubectl run --namespace=advanced-policy-demo7 access --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ # wget --timeout=2 google.com \
>wget --timeout=2 google.com
Connecting to google.com (172.217.167.110:80)
Connecting to www.google.com (142.251.221.68:80)
saving to 'index.html'
index.html           100% |********************************************************************************| 21425  0:00:00 ETA
'index.html' saved
wget: bad address 'wget'
/ # ls
bin         etc         index.html  lib64       root        tmp         var
dev         home        lib         proc        sys         usr
/ # cat index.html
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"><head><meta content="Search the world's information, incl
uding webpages, images, videos and more. Google has many special features to help you find exactly what you're looking for." name="descrip
tion"><meta content="noodp, " name="robots"><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/bran
ding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="d32scS-Dg1Kx2IDOcGR-ZA">(function()
{var _g={kEI:'F9MRZ4GdMoTShbIPssXWyA0',kEXPI:'0,3700286,1098,538661,2872,2891,73050,16105,18161,162095,342,21266,1758,6699,124314,2006,815
5,23351,8702,13733,9779,62657,36747,3801,2412,33249,15816,1804,7734,18098,21250,1635,9708,3785,15783,12989,14094,5203209,9466,1021,395,883
2117,1212,64,4,23936043,4043710,16672,43887,3,1603,3,2124363,23029351,8163,4636,16436,2728,81317,11733,10889,885,6668,7612,8181,28703,757,
19970,14119,4892,2663,3431,3319,155,1,1,1,2481,13503,7736,9140,761,3838,328,3217,4,1235,1769,7974,1982,1,1,5,10638,4863,8159,687,3511,3780
,561,24,2938,5452,3613,25,9241,710,1134,207,9607,4046,54,361,1844,7,12,8547,726,4134,2380,2465,2970,5,319,1,3996,2389,1380,1910,1821,5030,
1539,4175,797,8683,1,1344,6,6836,7112,2224,1330,4145,836,4564,8,2988,251,594,1805,912,121,280,7829,223,2604,1,1066,1,1,2,3,273,2,1176,41,1
244,684,1,1,2,3,815,2851,428,2486,217,2321,828,1159,199,400,1,665,951,138,1498,9,1097,1,1179,365,1220,986,1,28,1347,1361,305,254,1,3,205,5
19,470,441,1,87,807,955,269,12,1432,4,1,6,1743,262,1826,1528,123,1,30,162,409,799,2,7,1,104,111,1335,180,545,921,2202,472,289,478,2,239,52
```

```
/ # wget -q --timeout=5 nginx -O -
wget: download timed out
```

» Ingress traffic is still blocked so create a new policy to allow communication between these pods

```
calicoctl create -f - <<EOF
apiVersion: projectcalico.org/v3
kind: NetworkPolicy
metadata:
  name: allow-nginx-ingress
  namespace: advanced-policy-demo7
spec:
  selector: app == 'nginx'
  types:
    - Ingress
  ingress:
    - action: Allow
      source:
        selector: run == 'access'
EOF
```

```
tusharpanchal21@cloudshell:~$ calicoctl create -f - <<EOF
> apiVersion: projectcalico.org/v3
> kind: NetworkPolicy
> metadata:
>   name: allow-nginx-ingress
>   namespace: advanced-policy-demo7
> spec:
>   selector: app == 'nginx'
>   types:
>     - Ingress
>   ingress:
>     - action: Allow
>       source:
>         selector: run == 'access'
> EOF
Failed to create 'NetworkPolicy' resource: [resource already exists: NetworkPolicy(advanced-policy-demo7/default.allow-nginx-ingress)]
```

```
tusharpanchal21@cloudshell:~$ kubectl run --namespace=advanced-policy-demo7 access --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ # wget -q --timeout=5 nginx -O -
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

» Now delete everything created in this practical:

```
# Delete Calico policies
calicoctl delete policy allow-busybox-egress -n advanced-policy-demo7
calicoctl delete policy allow-nginx-ingress -n advanced-policy-demo7
calicoctl delete gnp default-deny7

# Delete Kubernetes namespace
kubectl delete ns advanced-policy-demo7
```

```
tusharpanchal21@cloudshell:~$ calicoctl delete policy allow-busybox-egress -n advanced-policy-demo7
Successfully deleted 1 'NetworkPolicy' resource(s)
tusharpanchal21@cloudshell:~$ calicoctl delete policy allow-nginx-ingress -n advanced-policy-demo7
Successfully deleted 1 'NetworkPolicy' resource(s)
tusharpanchal21@cloudshell:~$ calicoctl delete gnp default-deny
Successfully deleted 1 'GlobalNetworkPolicy' resource(s)
tusharpanchal21@cloudshell:~$ kubectl delete ns advanced-policy-demo7
namespace "advanced-policy-demo7" deleted
```