**Applying NetworkPolicy in Kubernetes**

**Step 1: Install Cilium**

**Add the Cilium Helm Repository**

Once Helm is installed, you can add the official Cilium Helm repository:

helm repo add cilium https://helm.cilium.io/

helm repo update

A screen shot of a computer

AI-generated content may be incorrect.

This will add the Cilium repository to Helm and update your local Helm charts.

**Install Cilium Using Helm**

Now that the repository is added, you can install Cilium into your cluster using the following command:

helm install cilium cilium/cilium --namespace kube-system --set ipam.mode=kubernetes --set global.networkPolicy.enabled=true

A black screen with white text

AI-generated content may be incorrect.

* The --namespace kube-system flag ensures that Cilium is installed in the kube-system namespace (default for system components).
* The --set ipam.mode=kubernetes flag tells Cilium to use Kubernetes IP address management (IPAM).
* The --set global.networkPolicy.enabled=true enables NetworkPolicy enforcement for the cluster.

**Verify the Cilium Installation**

Once the Helm installation is complete, verify that the Cilium pods are running in your cluster:

kubectl get pods -n kube-system -l k8s-app=cilium

You should see the Cilium agent pods (cilium-agent) and Cilium operator pods (cilium-operator).

**Step 2: Check Existing Pod's Labels**

# Get labels of your running pod

kubectl get pod target-pod -n demo --show-labels



**Step 3: Match the labels that already exist on your pod:**

**network-policy.yaml:**

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: certificate-core-network-policy

namespace: demo

labels:

app: platform-platform-certificate-core

purpose: network-policy

environment: local

spec:

podSelector:

matchLabels:

service: **add the correct label here**

policyTypes:

- Ingress

- Egress

ingress:

- from:

- podSelector:

matchLabels:

app.kubernetes.io/name: ps-core

ports:

- protocol: TCP

port: 80

egress:

- {} *# Allow all egress traffic*

**Apply it:**

kubectl apply -f network-policy.yaml

**Verify:**

kubectl describe networkpolicy certificate-core-network-policy -n demo

**Step 4: Deploy Test Pods**

**allowed-pod.yaml (should have access):**

apiVersion: v1

kind: Pod

metadata:

name: allowed-pod

namespace: demo

labels:

app.kubernetes.io/name: ps-core

spec:

containers:

- name: curl

image: curlimages/curl:latest

command: ["sleep", "3600"]

**denied-pod.yaml (should be blocked):**

apiVersion: v1

kind: Pod

metadata:

name: denied-pod

namespace: demo

labels:

app: some-other-app

spec:

containers:

- name: curl

image: curlimages/curl:latest

command: ["sleep", "3600"]

**Apply both:**

kubectl apply -f allowed-pod.yaml

kubectl apply -f denied-pod.yaml

**Wait for pods to be ready:**

kubectl wait --for=condition=ready pod --all -n demo --timeout=120s

A screenshot of a computer screen

AI-generated content may be incorrect.

**Step 5: Test the NetworkPolicy**

**Get target pod IP:**

kubectl get pod target-pod -n demo -o wide

Copy the IP



**Test from allowed-pod (should succeed):**

kubectl exec -it allowed-pod -n demo -- curl -m 5 <http://TARGET_IP>

A computer screen with white text

AI-generated content may be incorrect.

**Expected result: nginx welcome page HTML**

**Test from denied-pod (should fail ):**

kubectl exec -it denied-pod -n demo -- curl -m 5 <http://TARGET_IP>



**Expected result: Connection timeout after 5 seconds**

curl: (28) Connection timed out after 5001 milliseconds

command terminated with exit code 28

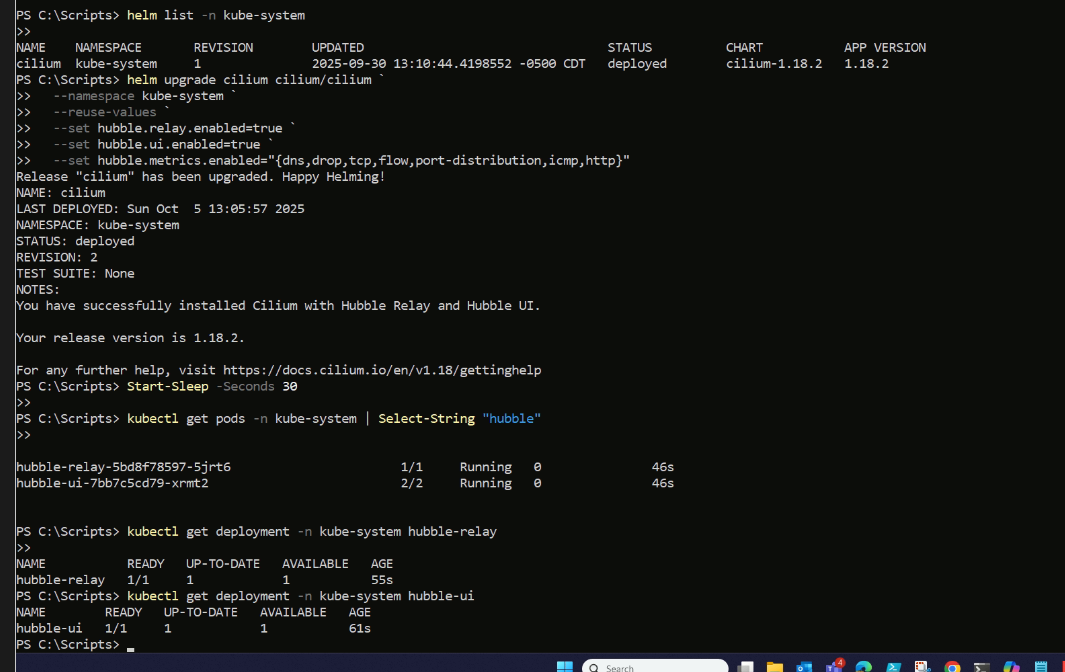
**Enabling Hubble for Observability**

*# Check if Helm has Cilium installed*

helm list -n kube-system

*# Enable Hubble*

helm upgrade cilium cilium/cilium ` --namespace kube-system ` --reuse-values ` --set hubble.relay.enabled=true ` --set hubble.ui.enabled=true ` --set hubble.metrics.enabled="{dns,drop,tcp,flow,port-distribution,icmp,http}"



**Wait and Check Hubble Pods**

*# Wait a bit*

Start-Sleep -Seconds 30

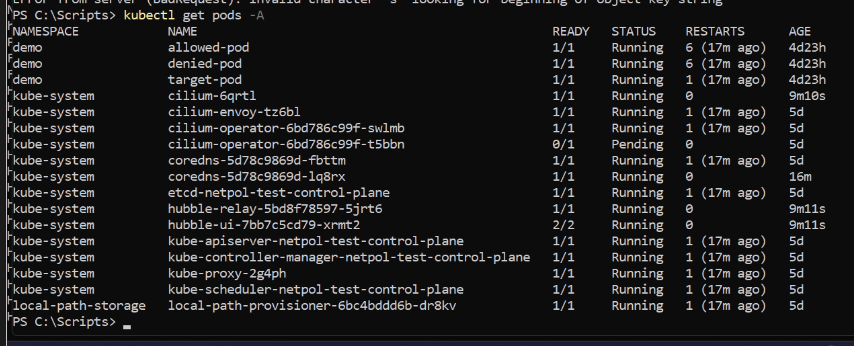
*# Check for Hubble pods*

kubectl get pods -n kube-system | Select-String "hubble"

*# Check deployment status*

kubectl get deployment -n kube-system hubble-relay

kubectl get deployment -n kube-system hubble-ui

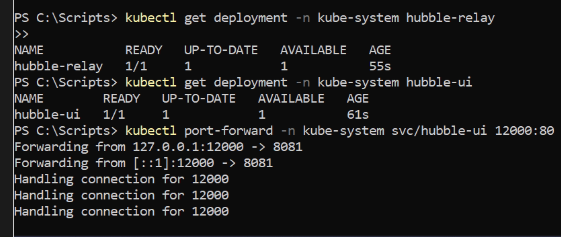


**Access Hubble UI**

*# Port-forward Hubble UI to your local machine*

kubectl port-forward -n kube-system svc/hubble-ui 12000:80

Keep this window open, then open your browser to: [**http://localhost:12000**](http://localhost:12000)



**Generate Traffic to See in Hubble**

In a **new PowerShell window**:

*# Get target pod IP*

$TARGET\_IP = kubectl get pod target-pod -n demo -o jsonpath='{.status.podIP}'

Write-Host "Target IP: $TARGET\_IP" -ForegroundColor Cyan

*# Test allowed pod (should succeed)*

Write-Host "`nTesting ALLOWED pod..." -ForegroundColor Green

kubectl exec -it allowed-pod -n demo -- curl -s http://$TARGET\_IP

*# Test denied pod (should timeout/fail)*

Write-Host "`nTesting DENIED pod..." -ForegroundColor Red

kubectl exec -it denied-pod -n demo -- curl -m 5 http://$TARGET\_IP

**In Hubble UI (**[**http://localhost:12000**](http://localhost:12000)**):**

1. Select namespace: demo

A screenshot of a computer

AI-generated content may be incorrect.

1. You'll see a service map with:
   * **Green line**: allowed-pod → target-pod (allowed by NetworkPolicy)
   * **Red X**: denied-pod → target-pod (blocked by NetworkPolicy)

A screenshot of a computer

AI-generated content may be incorrect.

**Install Hubble CLI**

curl -LO "https://raw.githubusercontent.com/cilium/hubble/master/stable.txt"

set /p HUBBLE\_VERSION=<stable.txt

curl -L --fail -O "https://github.com/cilium/hubble/releases/download/%HUBBLE\_VERSION%/hubble-windows-amd64.tar.gz"

curl -L --fail -O "https://github.com/cilium/hubble/releases/download/%HUBBLE\_VERSION%/hubble-windows-amd64.tar.gz.sha256sum"

certutil -hashfile hubble-windows-amd64.tar.gz SHA256

type hubble-windows-amd64.tar.gz.sha256sum

:: verify that the checksum from the two commands above match

tar zxf hubble-windows-amd64.tar.gz

**Now you can validate that you can access the Hubble API via the installed CLI:**

hubble status -P

Healthcheck (via 127.0.0.1:4245): Ok

Current/Max Flows: 11917/12288 (96.98%)

Flows/s: 11.74

Connected Nodes: 3/3

**Useful Hubble CLI Commands**

*# See only dropped traffic (policy denials)*

hubble observe --server localhost:4245 --verdict DROPPED --namespace demo

*# See only allowed traffic*

hubble observe --server localhost:4245 --verdict FORWARDED --namespace demo

*# See traffic from denied-pod*

hubble observe --server localhost:4245 --from-pod demo/denied-pod

*# See traffic to target-pod*

hubble observe --server localhost:4245 --to-pod demo/target-pod

*# See last 50 flows*

hubble observe --server localhost:4245 --last 50 --namespace demo

**Summary: Windows You Need Open**

1. **Window 1**: Hubble UI port-forward → kubectl port-forward -n kube-system svc/hubble-ui 12000:80
2. **Window 2**: Hubble Relay port-forward → kubectl port-forward -n kube-system svc/hubble-relay 4245:80
3. **Window 3**: Hubble observe → hubble observe --server localhost:4245 --namespace demo --follow
4. **Browser**: Open <http://localhost:12000>