Tim Allclair
Software Engineer, Google
@tallclair - tallclair@google.com

Layers of Isolation in Kubernetes

Google Cloud

What is isolation?

Confidentiality.

A process cannot read information outside its isolation boundary.

Integrity.

A process cannot alter data or behavior outside its isolation boundary.

Availability.

A process cannot disrupt services or processes outside its isolation boundary.

Why is it difficult?

Multi dimensional

Resource isolation, data isolation, and process isolation can be independent axes.

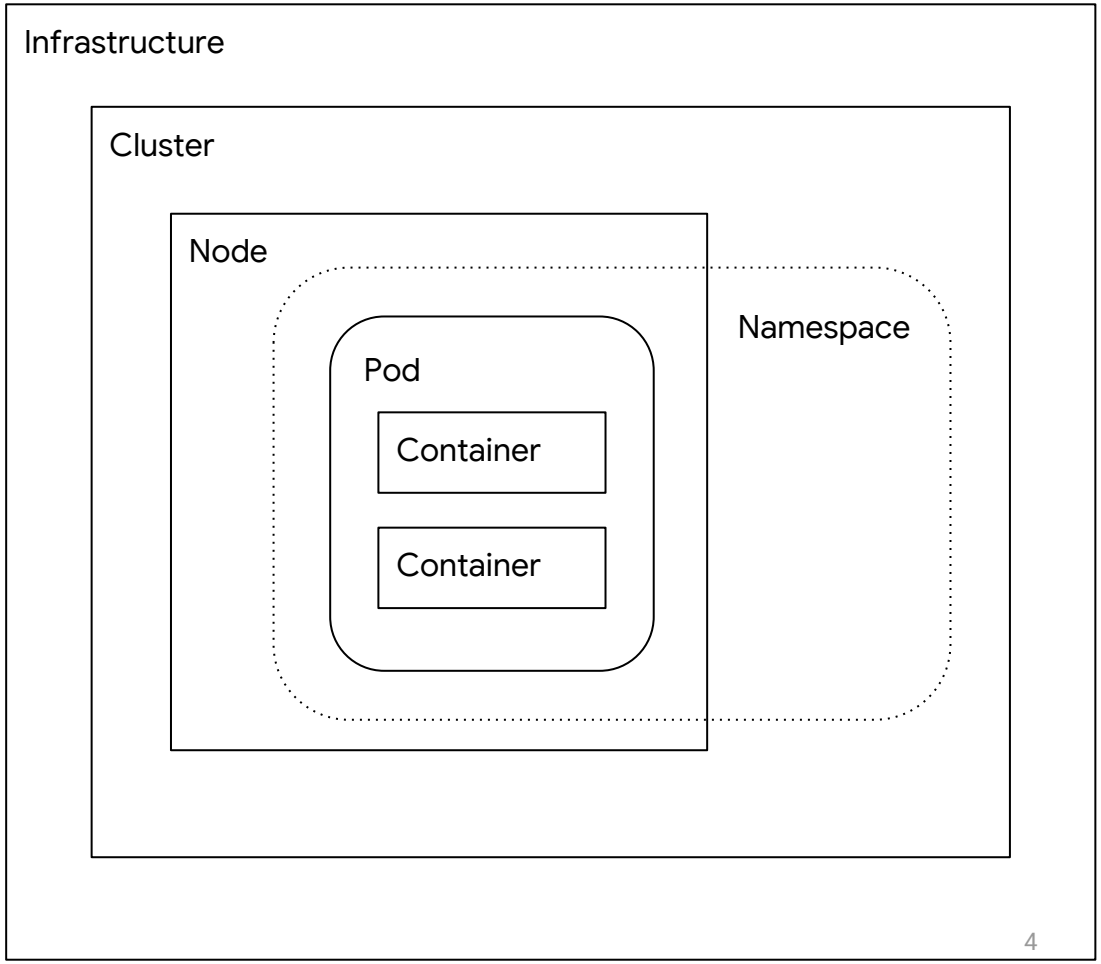
Security requires a holistic approach - attackers will find the weakest link.

Directional

Isolating the Kubelet from a container does not mean the container is isolated from the Kubelet.

Layers

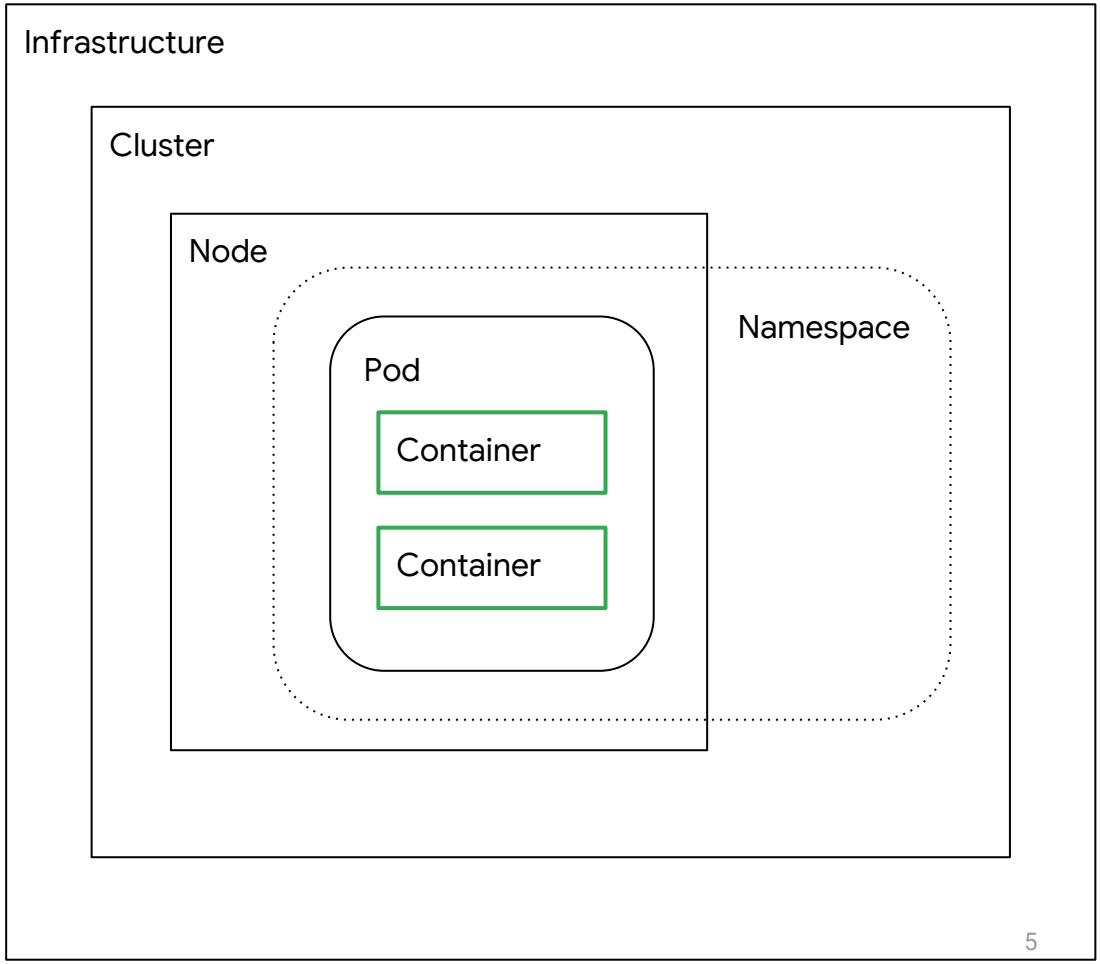
1. Containers
2. Pods
3. Namespaces
4. Nodes
5. Clusters
6. Infrastructure



Google Cloud

Layers

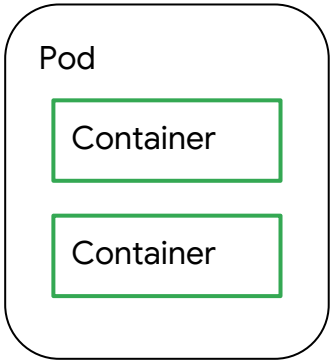
1. **Containers**
2. Pods
3. Namespaces
4. Nodes
5. Clusters
6. Infrastructure



Google Cloud

How much isolation is there between 2 containers in the same pod?

A lot, actually.



Container Isolation

Hardware Resources

Requests & Limits

Cgroups: CPU, memory

Kubelet: disk usage

Kernel Resources

Namespaces:
- filesystem (mount)
- PIDs

Attack Surface Reduction

Defaults:
- Capabilities
- LSM (AppArmor/SELinux)

Best Practices:
- Seccomp
- **Non-root!**

What isn't isolated?

Network - shared namespace, loopback, veth, IP address

Hardware resources - disk contention (IOPs), bandwidth

Kernel resource exhaustion - PIDs, file descriptors

Identity - shared service account

Example Shutting down a node

```
$
```

Example Shutting down a node

```
$ kubectl run --rm -it alpine --image=alpine sh
```

Example Shutting down a node

```
$ kubectl run --rm -it alpine --image=alpine sh

/ #
```

Example Shutting down a node

```
$ kubectl run --rm -it alpine --image=alpine sh

/ # uptime
```

Example Shutting down a node

```
$ kubectl run --rm -it alpine --image=alpine sh

/ # uptime
22:20:00 up 18 days, 23:08,  load average: 0.00, 0.05, 0.02

/ #
```

Example Shutting down a node

```
$ kubectl run --rm -it alpine --image=alpine sh

/ # uptime
22:20:00 up 18 days, 23:08,  load average: 0.00, 0.05, 0.02

/ # poweroff -f
```

Example Shutting down a node

```
$ kubectl run --rm -it alpine --image=alpine sh

/ # uptime
22:20:00 up 18 days, 23:08,  load average: 0.00, 0.05, 0.02

/ # poweroff -f
poweroff: Operation not permitted

/ #
```

Example Shutting down a node

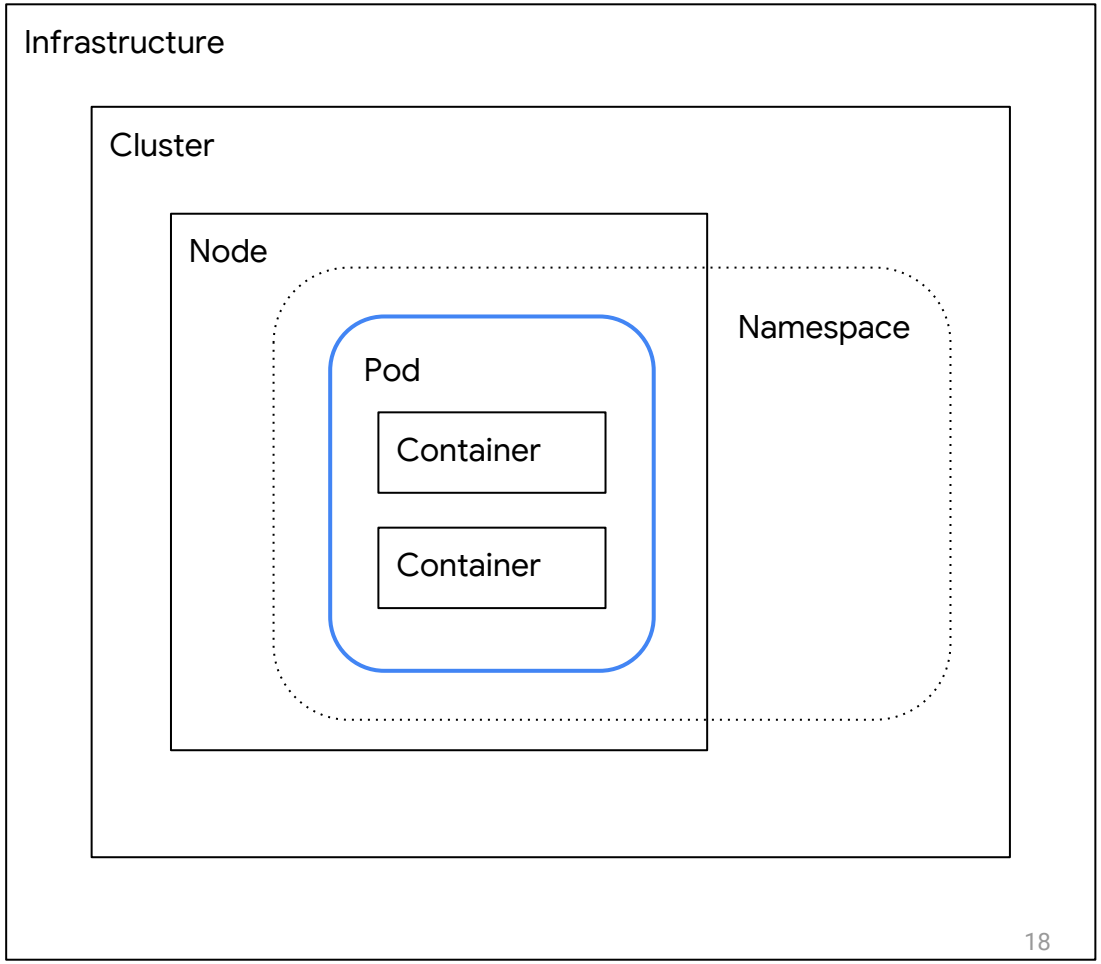
```
$ kubectl run --rm -it alpine --image=alpine sh

/ # uptime
22:20:00 up 18 days, 23:08,  load average: 0.00, 0.05, 0.02

/ # poweroff -f
poweroff: Operation not permitted

/ # f(){ f|f& };f   # WARNING: Don't try this!
```

Example Shutting down a node

```
kubelet \
  --feature-gates="SupportPodPidsLimit=true" \
  --pod-max-pids=1000 \
  ...
```

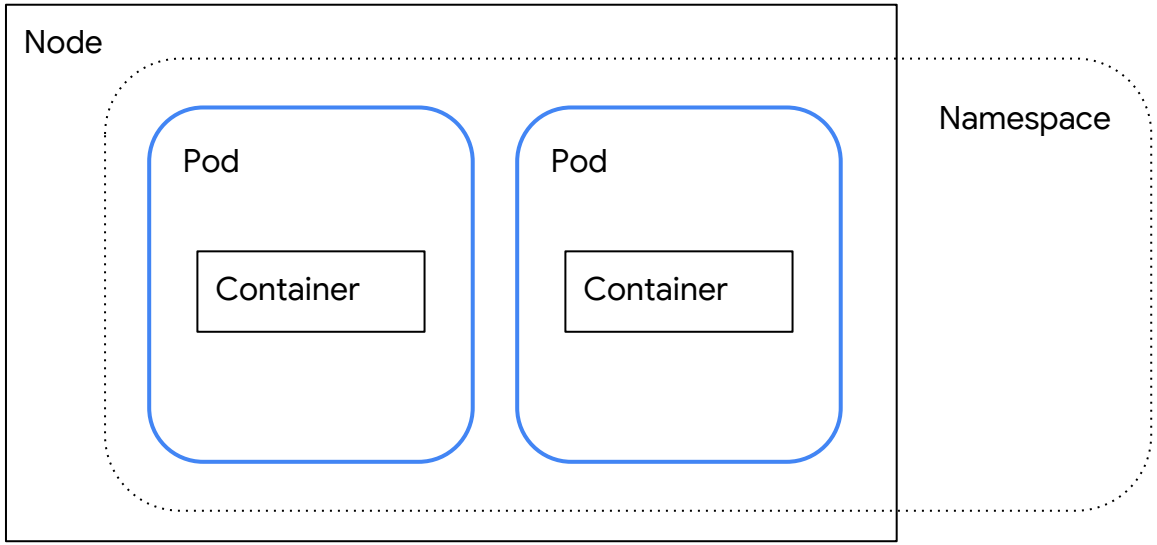
Layers

1. Containers
2. **Pods**
3. Namespaces
4. Nodes
5. Clusters
6. Infrastructure



Google Cloud

How much isolation is there between 2 pods on the same node?



Pod Isolation

Network - namespace, loopback, veth, IP address, NetworkPolicy

Identity - ServiceAccounts

Policy - PodSecurityPolicy, NetworkPolicy, SchedulingPolicy (WIP)

Volumes - EmptyDir

What isn't isolated?

Hardware resources - IOps, bandwidth

Kernel resource exhaustion - PIDs, file descriptors

Still only a single security boundary!

Example What's on the network?

```
$
```

Example What's on the network?

```
$ kubectl run --rm -it alpine --image=alpine sh
```

Example What's on the network?

```
$ kubectl run --rm -it alpine --image=alpine sh

/ #
```

Example What's on the network?

```
$ kubectl run --rm -it alpine --image=alpine sh

/ # apk add --no-cache nmap
```

Example What's on the network?

```
$ kubectl run --rm -it alpine --image=alpine sh

/ # apk add --no-cache nmap
...
OK: 18 MiB in 17 packages

/ #
```

Example What's on the network?

```
$ kubectl run --rm -it alpine --image=alpine sh

/ # apk add --no-cache nmap
...
OK: 18 MiB in 17 packages

/ # nmap -p- 10.0.0.0/8
```

Example What's on the network?

```
$ kubectl run --rm -it alpine --image=alpine sh

/ # apk add --no-cache nmap
...
OK: 18 MiB in 17 packages

/ # nmap -p- 10.0.0.0/8
^C

/ #
```

Example What's on the network?

```
$
```

Example What's on the network?

```
$ kubectl get nodes \
  -o jsonpath="{.items[0].status.addresses[?(.type=='InternalIP')].address}"
```

Example What's on the network?

```
$ kubectl get nodes \
  -o jsonpath="{.items[0].status.addresses[?(.type=='InternalIP')].address}"
10.240.0.4

$
```

Example What's on the network?

```
$ kubectl get nodes \
  -o jsonpath="{.items[0].status.addresses[?(.type=='InternalIP')].address}"
10.240.0.4

$ kubectl get nodes -o jsonpath="{.items[0].status.addresses}"
```

Example What's on the network?

```
$ kubectl get nodes \
  -o jsonpath="{.items[0].status.addresses[?(.type=='InternalIP')].address}"
10.240.0.4

$ kubectl get nodes -o jsonpath="{.items[0].status.addresses}"
10.64.2.0/24

$
```

Example What's on the network?

```
/ #
```

Example What's on the network?

```
/ # nmap -p- 10.240.0.4
```

Example What's on the network?

```
/ # nmap -p- 10.240.0.4

Nmap scan report for kubernetes-master (10.240.0.4)
Host is up (0.00031s latency).
Not shown: 65523 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https
2380/tcp  filtered etcd-server
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown
```

Example What's on the network?

```
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https
2380/tcp  filtered etcd-server
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown
10251/tcp open     unknown
10252/tcp open     apollo-relay
10255/tcp open     unknown
10257/tcp open     unknown
24231/tcp open     unknown
```

Example What's on the network?

```
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https              <-- Kubernetes API
2380/tcp  filtered etcd-server
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown
10251/tcp open     unknown
10252/tcp open     apollo-relay
10255/tcp open     unknown
10257/tcp open     unknown
24231/tcp open     unknown
```

Example What's on the network?

```
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https
2380/tcp  filtered etcd-server        <-- etcd
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown
10251/tcp open     unknown
10252/tcp open     apollo-relay
10255/tcp open     unknown
10257/tcp open     unknown
24231/tcp open     unknown
```

Example What's on the network?

```
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https
2380/tcp  filtered etcd-server
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown            <-- Authenticated Kubelet port
10251/tcp open     unknown
10252/tcp open     apollo-relay
10255/tcp open     unknown
10257/tcp open     unknown
24231/tcp open     unknown
```

Example What's on the network?

```
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https
2380/tcp  filtered etcd-server
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown
10251/tcp open     unknown            <-- Scheduler port
10252/tcp open     apollo-relay
10255/tcp open     unknown
10257/tcp open     unknown
24231/tcp open     unknown
```

Example What's on the network?

```
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https
2380/tcp  filtered etcd-server
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown
10251/tcp open     unknown
10252/tcp open     apollo-relay       <-- "InsecureKubeControllerManagerPort"
10255/tcp open     unknown
10257/tcp open     unknown
24231/tcp open     unknown
```

Example What's on the network?

```
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https
2380/tcp  filtered etcd-server
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown
10251/tcp open     unknown
10252/tcp open     apollo-relay
10255/tcp open     unknown            <-- Kubelet unauthenticated port (read only)
10257/tcp open     unknown
24231/tcp open     unknown
```

Example What's on the network?

```
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https
2380/tcp  filtered etcd-server
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown
10251/tcp open     unknown
10252/tcp open     apollo-relay
10255/tcp open     unknown
10257/tcp open     unknown            <-- Kube controller manager port (secure)
24231/tcp open     unknown
```

Example What's on the network?

```
PORT      STATE    SERVICE
22/tcp    open     ssh
443/tcp   open     https
2380/tcp  filtered etcd-server
2381/tcp  filtered compaq-https
5355/tcp  filtered llmnr
8086/tcp  open     d-s-n
10250/tcp open     unknown
10251/tcp open     unknown
10252/tcp open     apollo-relay
10255/tcp open     unknown
10257/tcp open     unknown
24231/tcp open     unknown
```

Example What's on the network?

```
/ # curl 10.240.0.4:10255/pods
```

Example What's on the network?

```
/ # curl 10.240.0.4:10255/pods
{"kind":"PodList","apiVersion":"v1","metadata":{},"items":[{"metadata":{"name":"etcd-
server-events-kubernetes-master","namespace":"kube-system","selfLink":"/api/v1/namesp
aces/kube-system/pods/etcd-server-events-kubernetes-master","uid":"d43ea5c0364c7b5eea
affd278fe30852","creationTimestamp":null,"annotations":{"kubernetes.io/config.hash":"
d43ea5c0364c7b5eeaaffd278fe30852","kubernetes.io/config.seen":"2018-10-19T00:59:41.28
0276889Z","kubernetes.io/config.source":"file","scheduler.alpha.kubernetes.io/critica
l-pod":"","seccomp.security.alpha.kubernetes.io/pod":"docker/default"}},"spec":{"volu
mes":[{"name":"varetcd","hostPath":{"path":"/mnt/disks/master-pd/var/etcd","type":""}
},{"name":"varlogetcd","hostPath":{"path":"/var/log/etcd-events.log","type":"FileOrCr
eate"}},{"name":"etc","hostPath":{"path":"/etc/srv/kubernetes","type":""}}],"containe
rs":[{"name":"etcd-container","image":"k8s.gcr.io/etcd:3.2.24-1","command":["/bin/sh"
,"-c","if [ -e /usr/local/bin/migrate-if-needed.sh ]; then
```

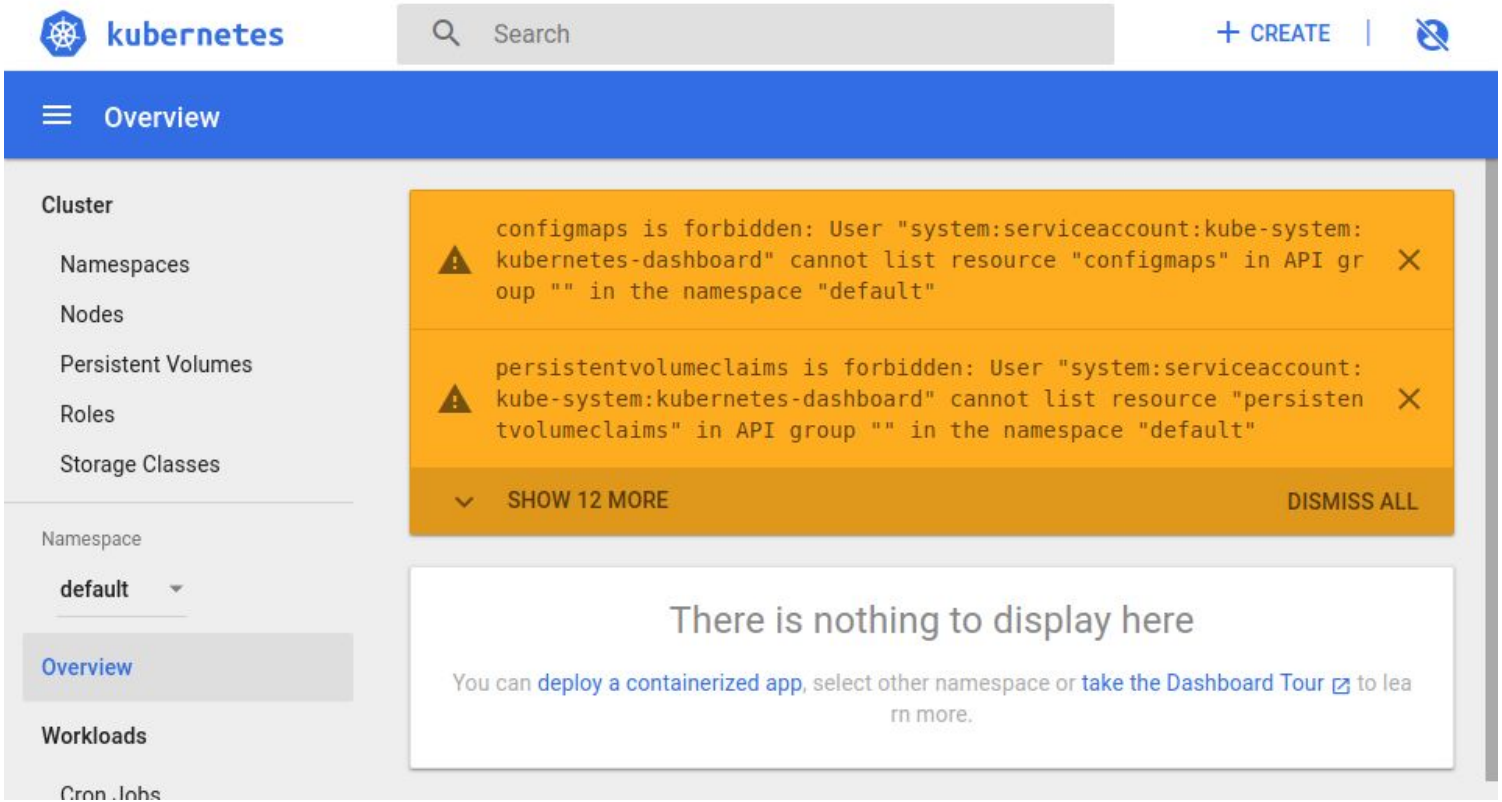
Example What's on the network?

```
/ # curl -k https://10.64.3.3:8443/
```

Example What's on the network?

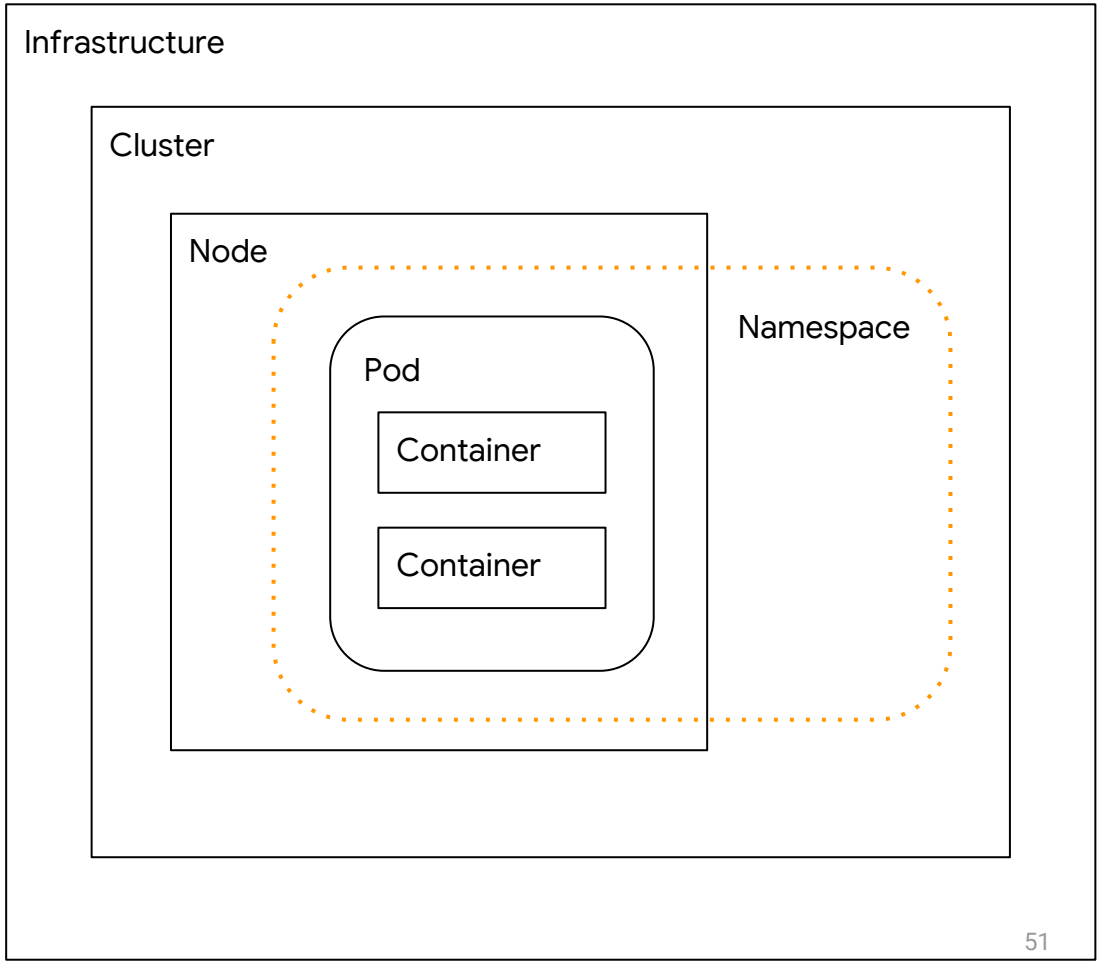
```
/ # curl -k https://10.64.3.3:8443/
<!doctype html> <html ng-app="kubernetesDashboard"> <head> <meta charset="utf-8">
<title ng-controller="kdTitle as $ctrl" ng-bind="$ctrl.title()"></title> <link
rel="icon" type="image/png" href="assets/images/kubernetes-logo.png"> <meta
name="viewport" content="width=device-width"> <link rel="stylesheet"
href="static/vendor.93db0a0d.css"> <link rel="stylesheet"
href="static/app.ef45991b.css"> </head> <body ng-controller="kdMain as $ctrl">
<!--[if lt IE 10]>
      <p class="browsehappy">You are using an <strong>outdated</strong> browser.
      Please <a href="http://browsehappy.com/">upgrade your browser</a> to improve
your
      experience.</p>
    <![endif]--> <kd-login layout="column" layout-fill ng-if="$ctrl.isLoginState()">
```

Example What's on the network?



Layers

1. Containers
2. Pods
3. **Namespaces**
4. Nodes
5. Clusters
6. Infrastructure



Google Cloud

Namespace Isolation

Identity - Service accounts scoped to namespace

Authorization - Roles & Rolebindings scoped to namespace

Resources - Secrets, ConfigMaps, PersistentVolumeClaim, ...

No effect at the node!

Example Reading Secrets

```
$
```

Example Reading Secrets

```
$ kubectl auth can-i get secrets
```

Example Reading Secrets

```
$ kubectl auth can-i get secrets
no - no RBAC policy matched

$
```

Example Reading Secrets

```
$ kubectl auth can-i get secrets
no - no RBAC policy matched

$ kubectl auth can-i create pods
```

Example Reading Secrets

```
$ kubectl auth can-i get secrets
no - no RBAC policy matched

$ kubectl auth can-i create pods
yes

$
```

Example Reading Secrets

```
$ cat secret-reader.yaml
```

Example Reading Secrets

```yaml
apiVersion: v1
kind: Pod
metadata:
  name:   secret-reader
spec:
  containers:
    - name:    alpine
      image:   alpine:latest
      volumeMounts:
        - mountPath: /sec
          name: secret-volume
  volumes:
  - name: secret-volume
    secret:
      secretName: classified
```

Example Reading Secrets

```yaml
apiVersion: v1
kind: Pod
metadata:
  name:   secret-reader
spec:
  containers:
    - name:    alpine
      image:   alpine:latest
      volumeMounts:
        - mountPath: /sec
          name: secret-volume
  volumes:
  - name: secret-volume
    secret:
      secretName: classified
```

Example Reading Secrets

```
$
```

Example Reading Secrets

```
$ kubectl create -f secret-reader.yaml
```

Example Reading Secrets

```
$ kubectl create -f secret-reader.yaml
pod/secret-reader created

$
```

Example Reading Secrets

```
$ kubectl create -f secret-reader.yaml
pod/secret-reader created

$ kubectl cp secret-reader:/sec/classified .
```

Example Reading Secrets

```
$ kubectl create -f secret-reader.yaml
pod/secret-reader created

$ kubectl cp secret-reader:/sec/classified .

$
```

Example Reading Secrets

```
$ kubectl create -f secret-reader.yaml
pod/secret-reader created

$ kubectl cp secret-reader:/sec/classified .

$ cat classified
```

Example Reading Secrets

```
$ kubectl create -f secret-reader.yaml
pod/secret-reader created

$ kubectl cp secret-reader:/sec/classified .

$ cat classified
¯\_(ツ)_/¯

$
```

Example Reading Secrets

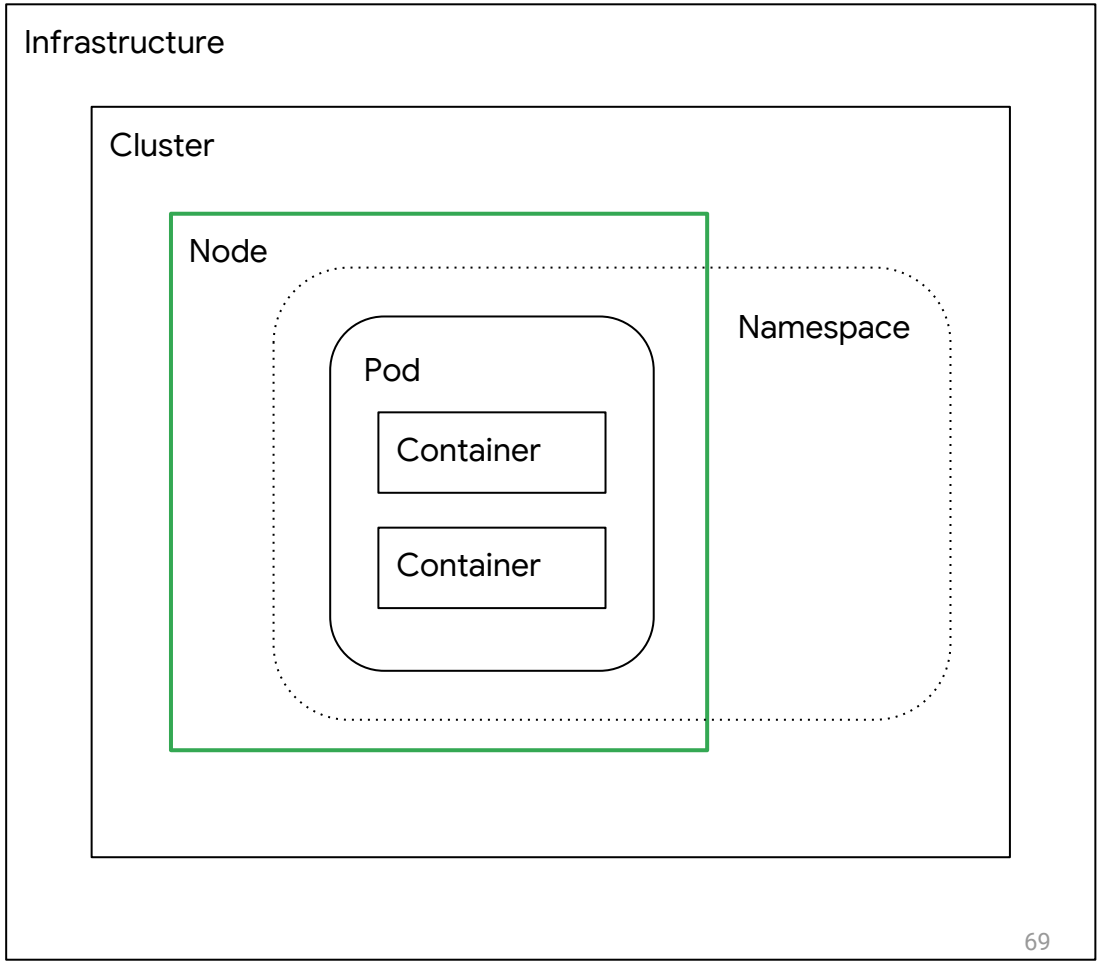
clusterrole/view - Read-only access to non-secret resources

clusterrole/edit - Edit access to namespaced resources

```
$ kubectl create rolebinding --clusterrole=view --user=tallclair@google.com
```

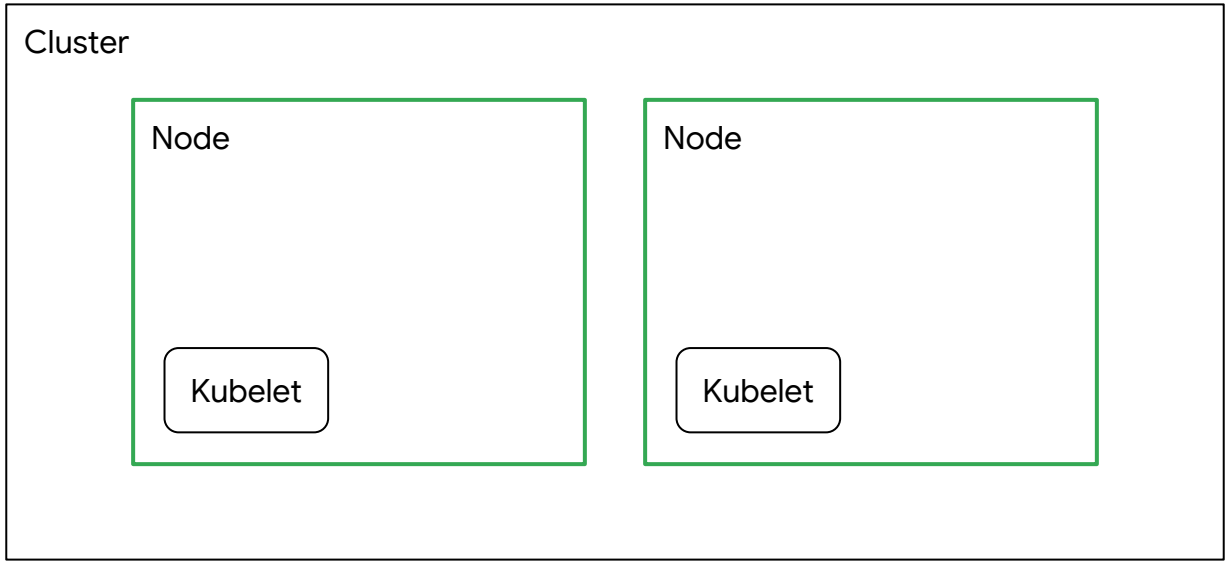
Layers

1. Containers
2. Pods
3. Namespaces
4. **Nodes**
5. Clusters
6. Infrastructure



Google Cloud

Node Isolation



Node Isolation

Resources - Hardware-level resource isolation *

Layers - A second-layer security boundary

* Or as good as our infrastructure

What isn't isolated?

Network - Anything exposed on the cluster network

Metadata - What is running in the cluster?

Control Plane - Node credentials, pod credentials, secrets, ...

Example Node Restriction

```
$
```

Example Node Restriction

```
$ ssh node-1
```

Example Node Restriction

```
$ ssh node-1

node-1 $
```

Example Node Restriction

```
$ ssh node-1

node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig get pods -o wide
```

Example Node Restriction

```
$ ssh node-1

node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig get pods -o wide
NAME     READY     STATUS     RESTARTS    AGE        IP            NODE
pod-1    1/1       Running    0           1d         10.64.1.18    node-1
pod-2    1/1       Running    0           22h        10.64.0.19    node-2

node-1 $
```

Example Node Restriction

```
node-1 $
```

Example Node Restriction

```
node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig describe pod-2
```

Example Node Restriction

```
node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig describe pod-2
Name:               pod-2
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               node-2/10.240.0.6
...
Volumes:
  secret:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  secret-2
    Optional:    false
  ...
```

Example Node Restriction

```
node-1 $
```

Example Node Restriction

```
node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig get secret secret-2
```

Example Node Restriction

```
node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig get secret secret-2
Error from server (Forbidden): secrets "secret-2" is forbidden: User
"system:node:node-1" cannot get resource "secrets" in API group "" in the namespace
"default"

node-1 $
```

Example Node Restriction

```
node-1 $
```

Example Node Restriction

```
node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig describe pod-1
```

Example Node Restriction

```
node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig describe pod-1
Name:               pod-1
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               node-1/10.240.2.6
...
Volumes:
  secret:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  secret-1
    Optional:    false
  ...
```

Example Node Restriction

```
node-1 $
```

Example Node Restriction

```
node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig get secret secret-1
```

Example Node Restriction

```
node-1 $ kubectl --kubeconfig=/var/lib/kubelet/kubeconfig get secret secret-1
NAME      TYPE      DATA      AGE
bkbt      Opaque    1         2h

node-1 $
```

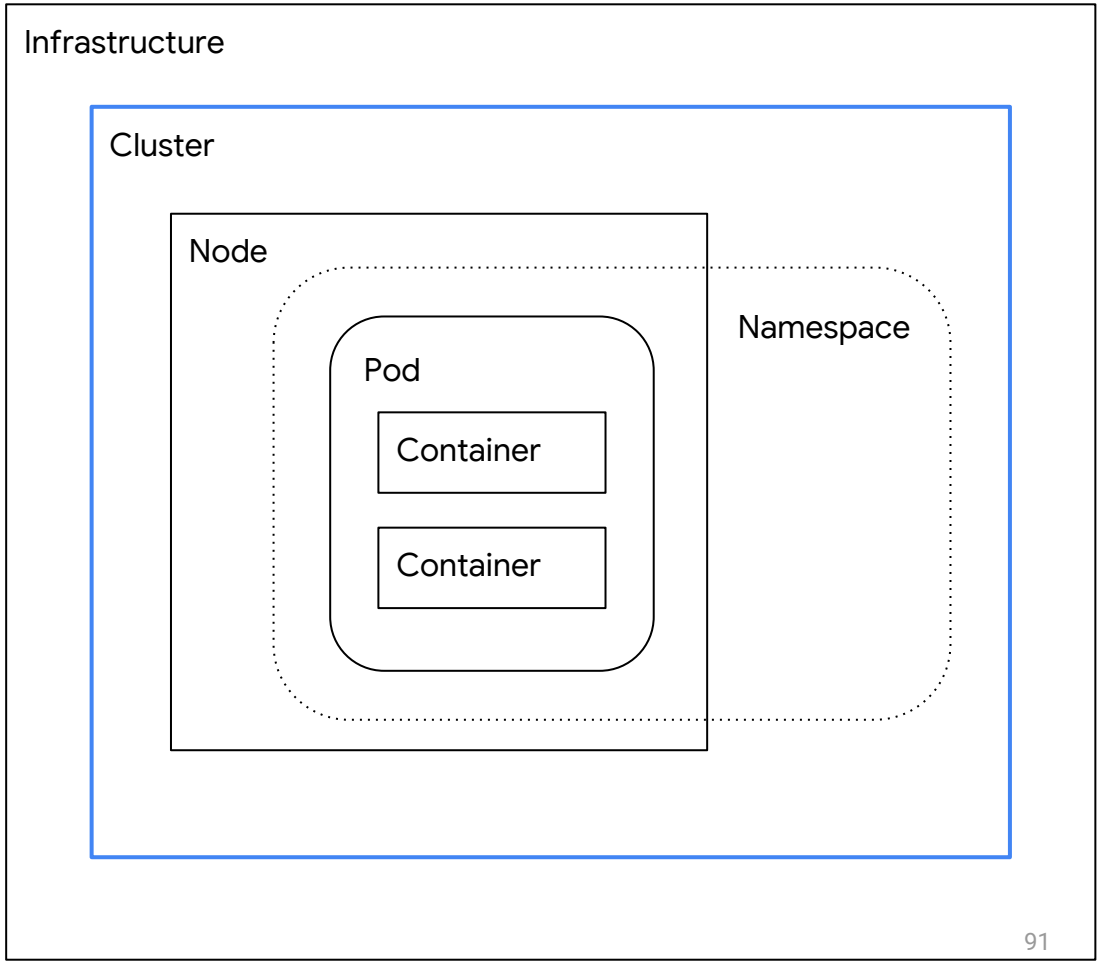
Aside: Sandbox Isolation

Bringing **node-level isolation** to the pod

Sandbox workloads with **gVisor** or **Kata Containers**

Layers

1. Containers
2. Pods
3. Namespaces
4. Nodes
5. **Clusters**
6. Infrastructure



Google Cloud

Cluster Isolation

Network Perimeter - Much stronger network isolation

Separate Datastore - Much stronger data isolation

Separate Control Plane - Much stronger identity, authorization, and metadata isolation

What isn't isolated?

Are we done?

What isn't isolated?

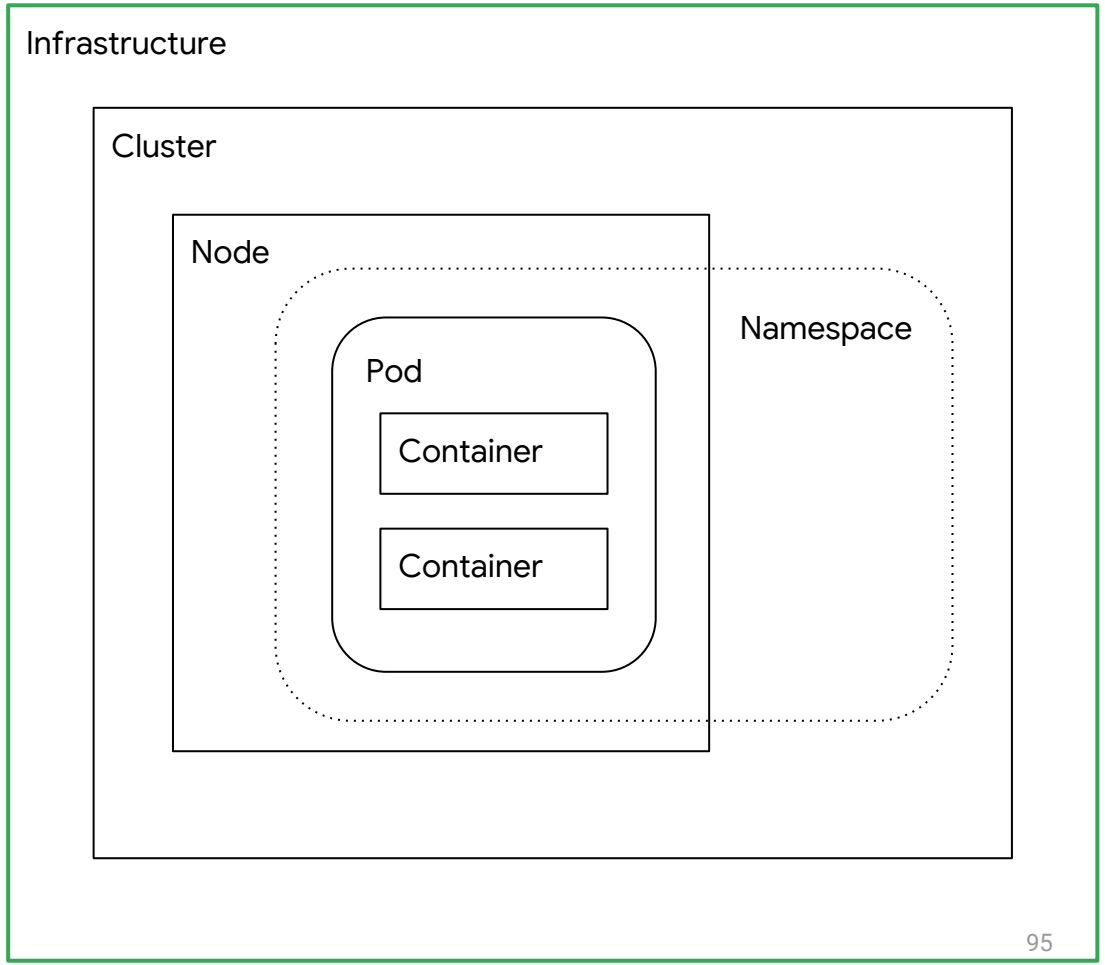
What other resources are exposed on the network?

What other services are exposed?

Metadata service?

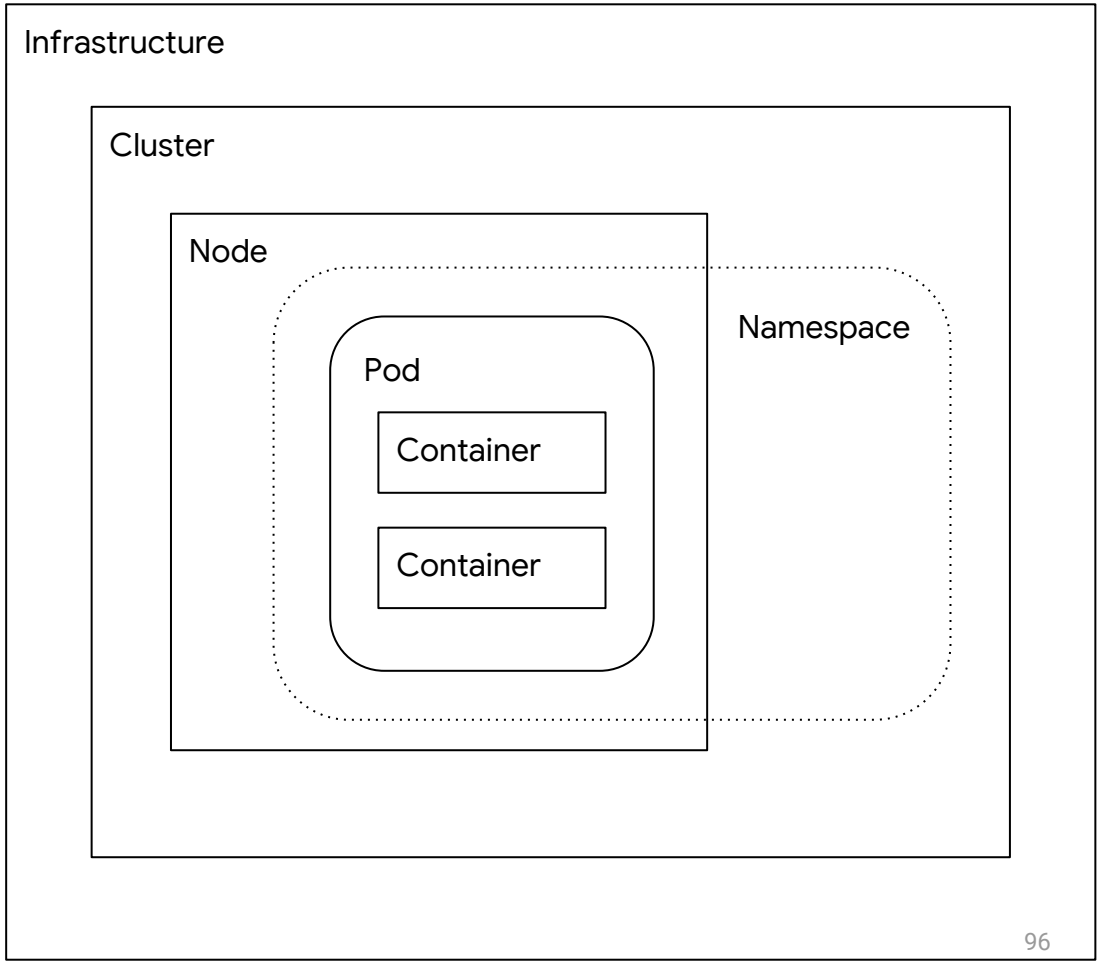
Layers

1. Containers
2. Pods
3. Namespaces
4. Nodes
5. Clusters
6. **Infrastructure**

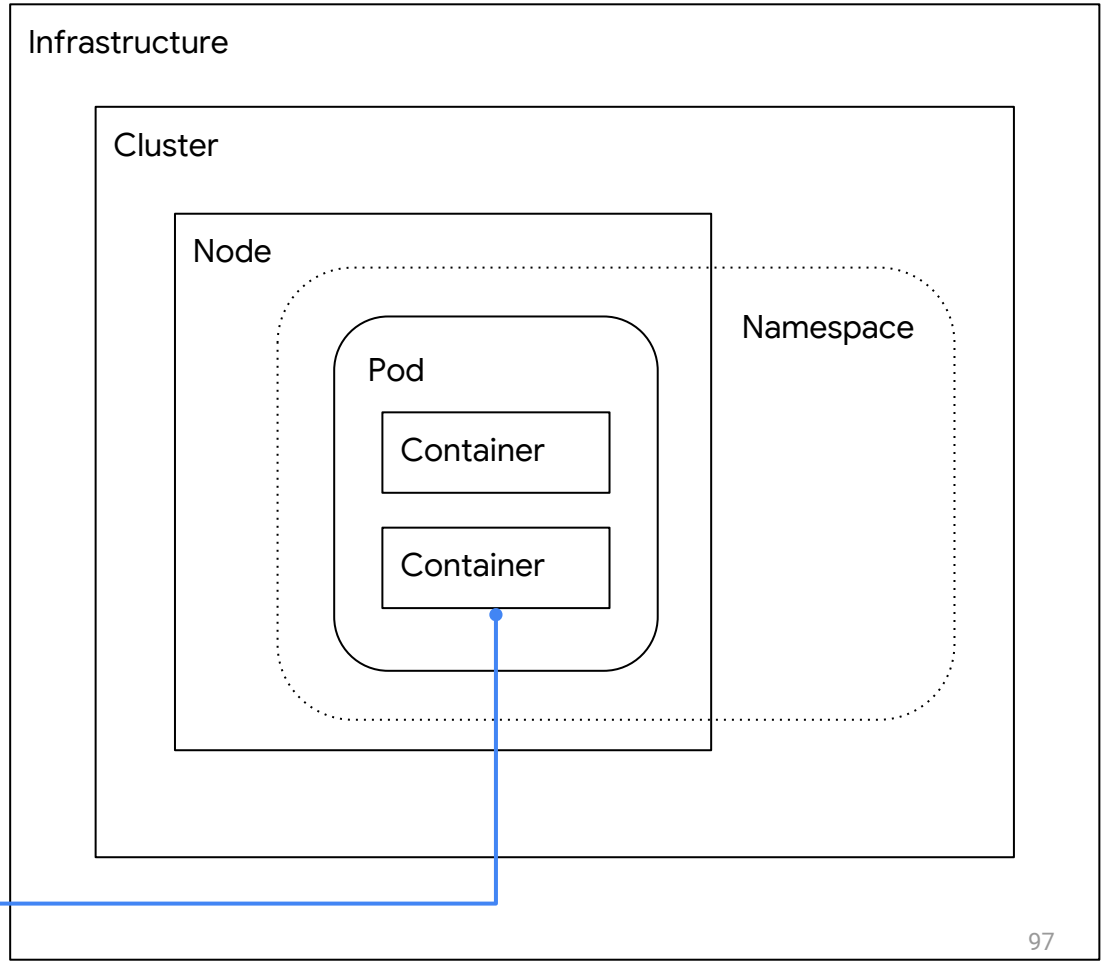


Google Cloud

Conclusion

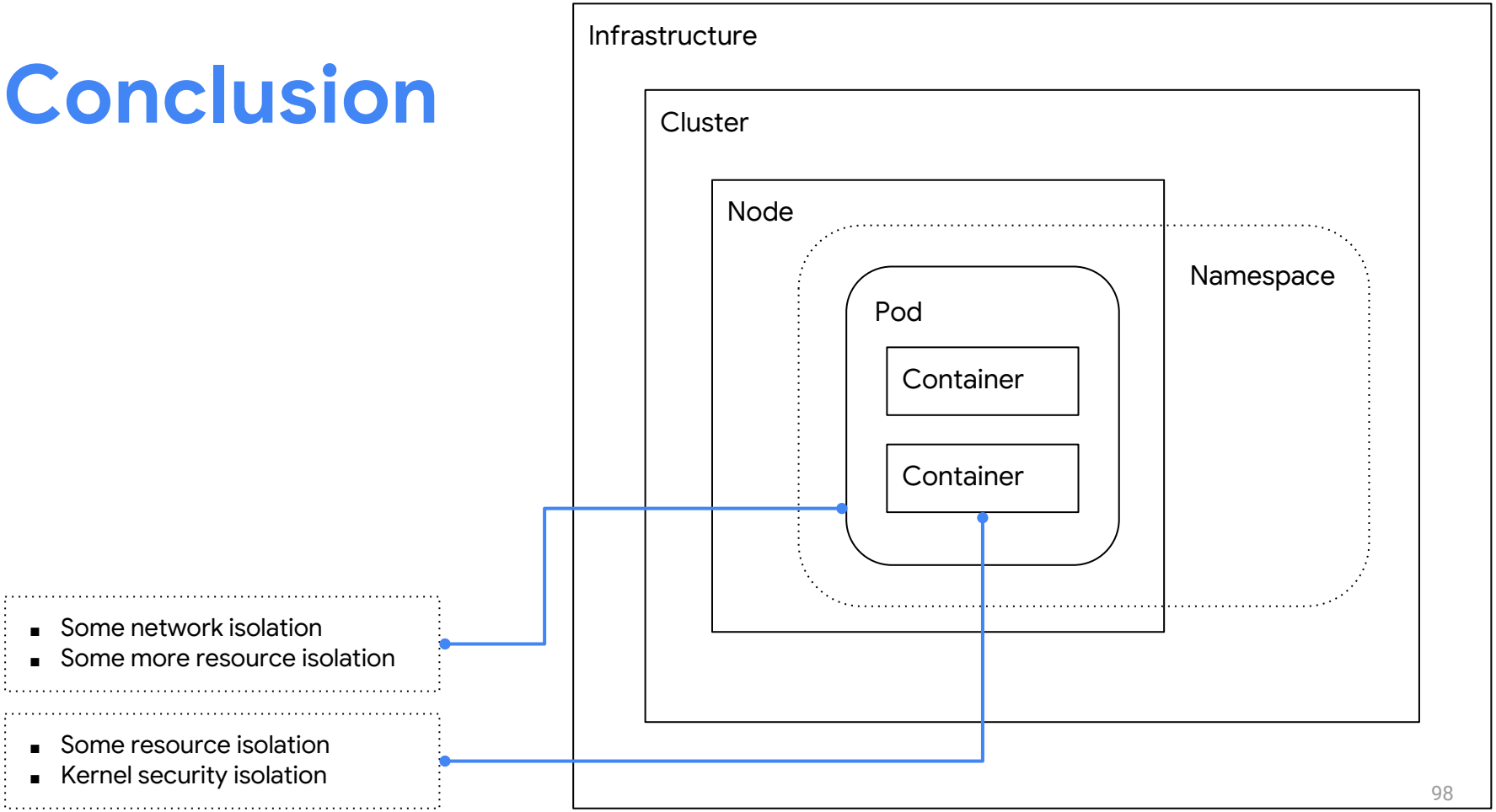

Infrastructure
Cluster
Node
Namespace
Pod
Container
Container

Conclusion

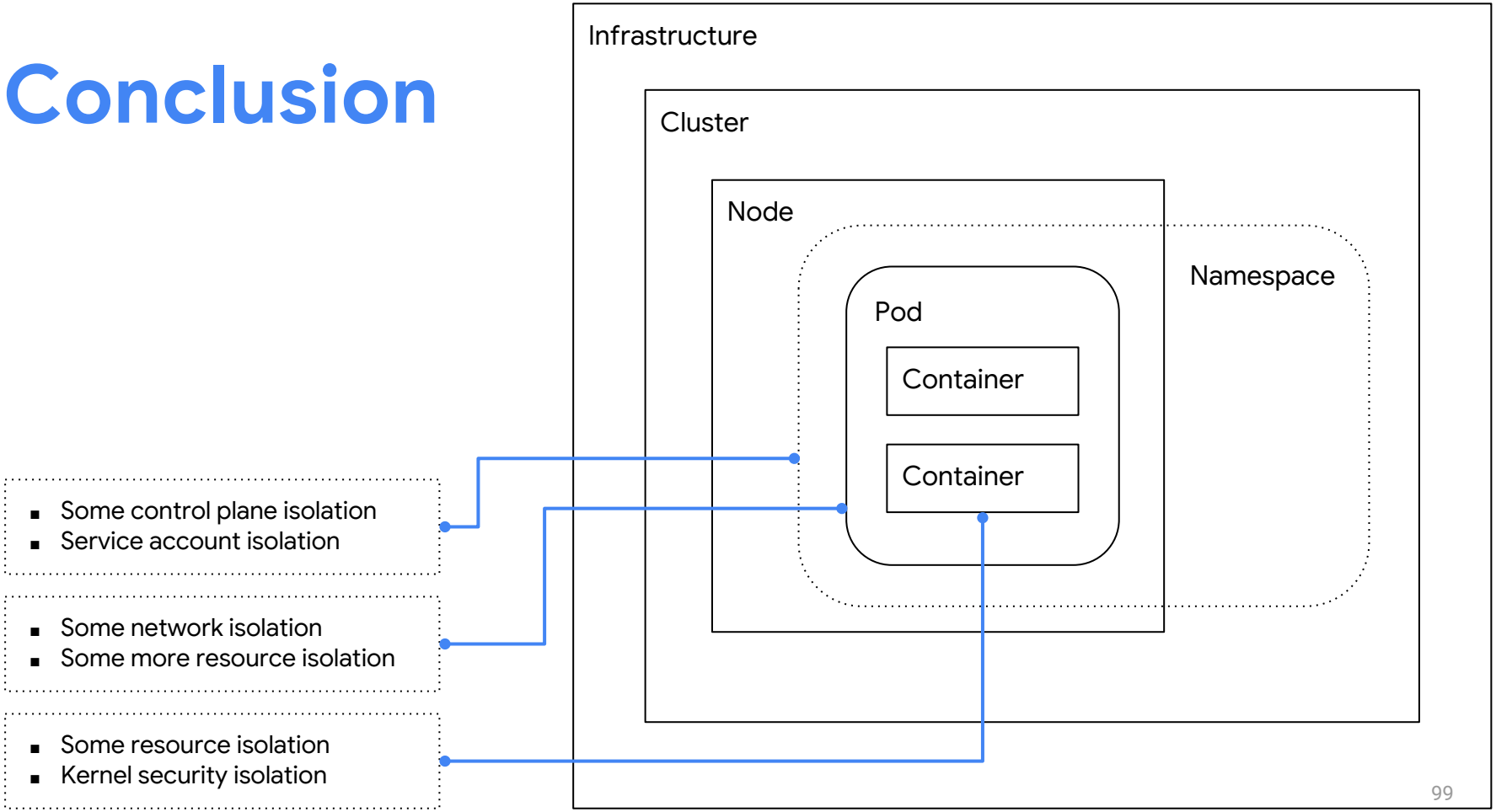


- Some resource isolation
- Kernel security isolation

Conclusion


Infrastructure
Cluster
Node
Namespace
Pod
Container
Container
Some network isolation
Some more resource isolation
Some resource isolation
Kernel security isolation

Conclusion


Infrastructure
Cluster
Node
Namespace
Pod
Container
Container
Some control plane isolation
Service account isolation
Some network isolation
Some more resource isolation
Some resource isolation
Kernel security isolation

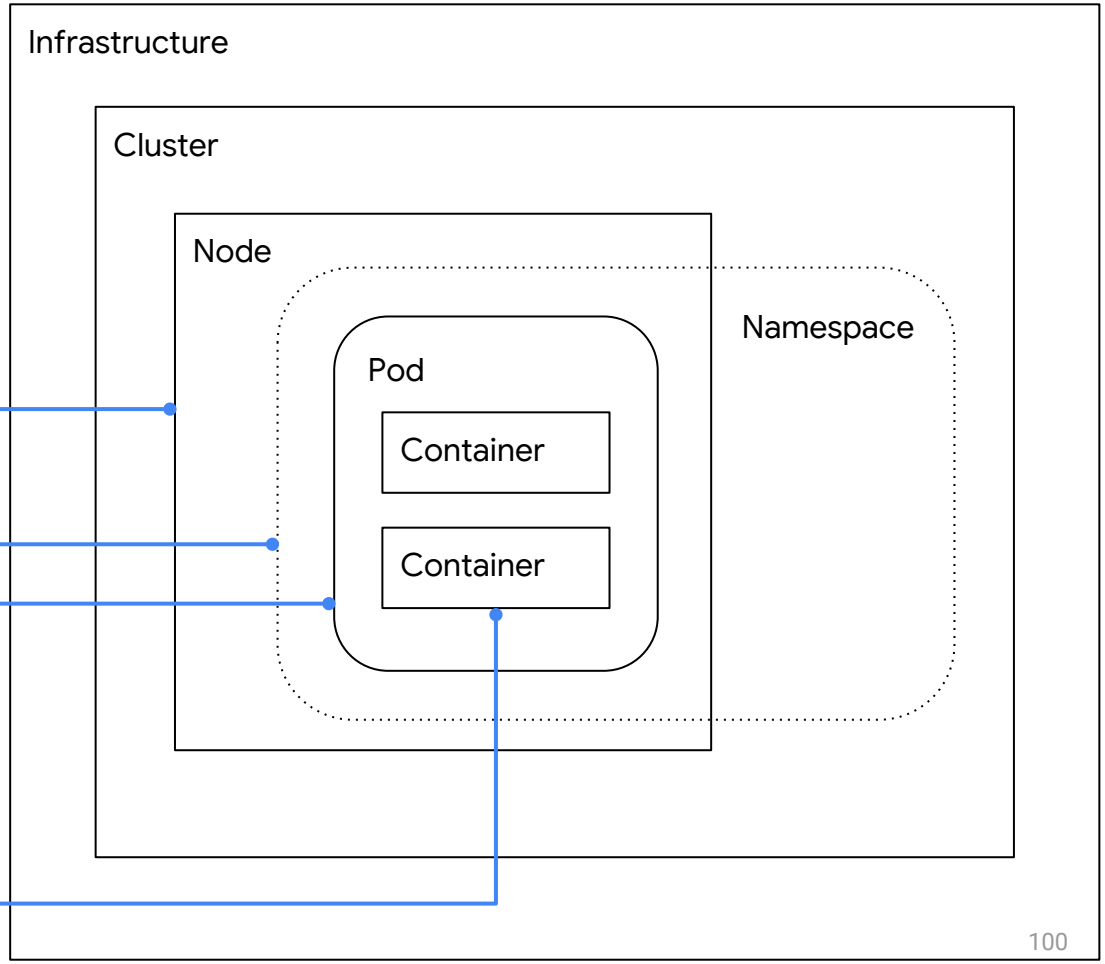
Conclusion

- Stronger resource isolation
- Stronger data isolation
- Second security boundary

- Some control plane isolation
- Service account isolation

- Some network isolation
- Some more resource isolation

- Some resource isolation
- Kernel security isolation



Conclusion

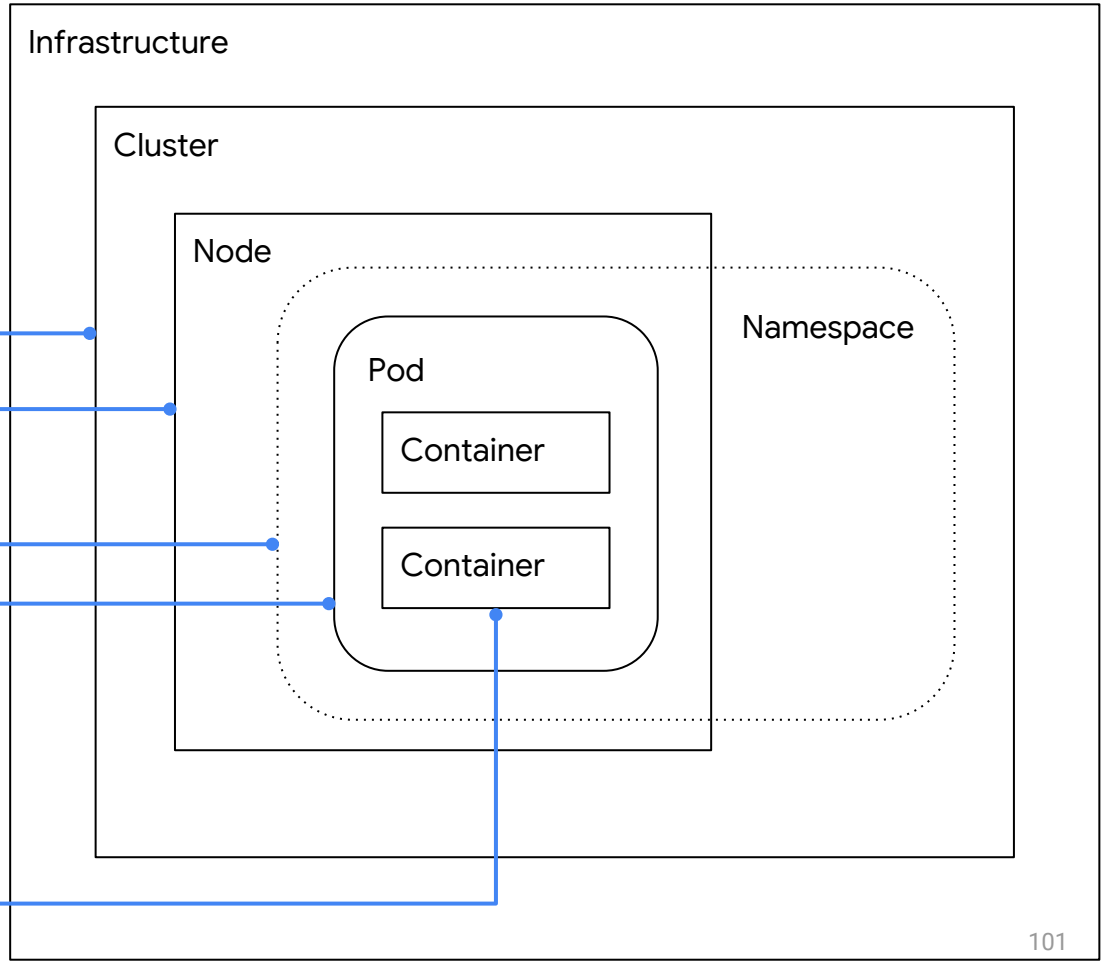
- Strongest control plane isolation
- Stronger network isolation
- Stronger data isolation
- Strong metadata isolation

- Stronger resource isolation
- Stronger data isolation
- Second security boundary

- Some control plane isolation
- Service account isolation

- Some network isolation
- Some more resource isolation

- Some resource isolation
- Kernel security isolation



- Strongest network isolation
- Strongest data isolation
- Strongest metadata isolation

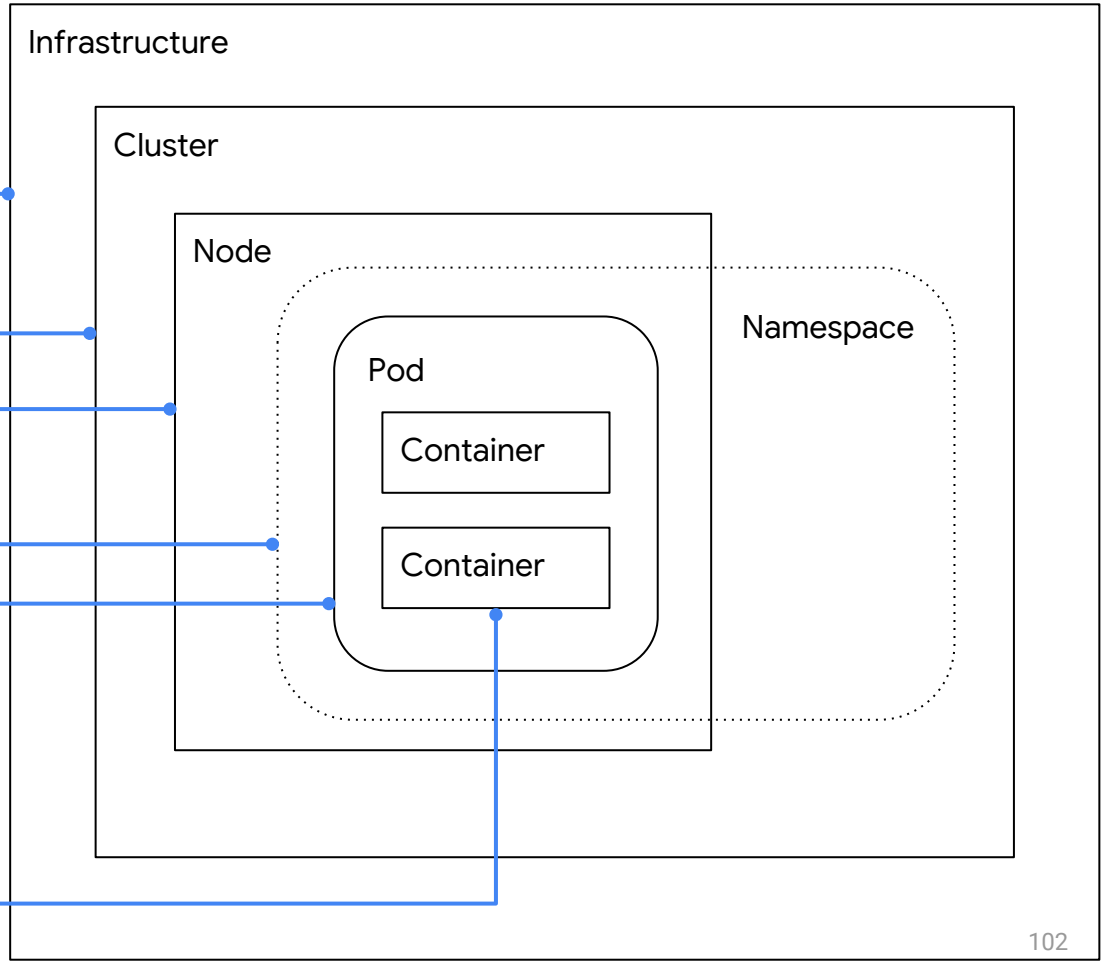
- Strongest control plane isolation
- Stronger network isolation
- Stronger data isolation
- Strong metadata isolation

- Stronger resource isolation
- Stronger data isolation
- Second security boundary

- Some control plane isolation
- Service account isolation

- Some network isolation
- Some more resource isolation

- Some resource isolation
- Kernel security isolation



Conclusion

Which layer should I design for?

Know your threat model!

Thank you!

Learn More:

Blog Post on Isolation Layers:
https://cloud.google.com/blog/products/gcp/exploring-container-security-isolation-at-different-layers-of-the-kubernetes-stack

Securing a Cluster: https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster/

Understanding and Hardening Linux Containers:
https://www.nccgroup.trust/us/our-research/understanding-and-hardening-linux-containers/