KODEKLOUD

www.kodekloud.com

cks.kodekloud.com

Disclaimer

THE INFORMATION FOUND ON THE WEBSITE, E-LEARNING PLATFORM AND WITHIN THE ONLINE COURSES ARE FOR INFORMATIONAL PURPOSES ONLY. KODEKLOUD WILL NOT BE HELD RESPONSIBLE FOR ANY DAMAGES THAT MAY BE INCURRED BY YOU AS A RESULT OF YOUR USE OF SUCH INFORMATION. ALL INFORMATION AND CONTENT ON THE WEBSITE, E-LEARNING PLATFORM AND ONLINE COURSE IS COPYRIGHTED, AND MAY NOT BE REPUBLISHED, COPIED, SOLD OR POSTED ANYWHERE ONLINE OR IN PRINT. KODEKLOUD RESERVES THE RIGHT TO TAKE THE NECESSARY LEGAL ACTION TO PREVENT YOU FROM (RE)-PUBLISHING, COPYING, SELLING, POSTING OR PRINTING ANY COPYRIGHTED INFORMATION AND CONTENT AVAILABLE ON THE WEBSITE, E-LEARNING PLATFORM AND ONLINE COURSE.

For the full terms & conditions visit terms.kodekloud.com

For questions write to support@kodekloud.com

Notice

- This presentation is to refer to course contents only.
- Some of the slides are meant to be animated. So may not be displayed correctly.
- Do not copy and paste command, code or YAML files from this file as it may not be in the right format and may contain hidden characters
- For code refer to the solutions in the lab or the Git repository associated with this course or official Kubernetes documentation pages.
- Some of the code in this deck maybe hidden for brevity

https://github.com/kodekloudhub/certified-kubernetes-security-specialist-cks-course

Perform Behaviour Analytics of syscalls

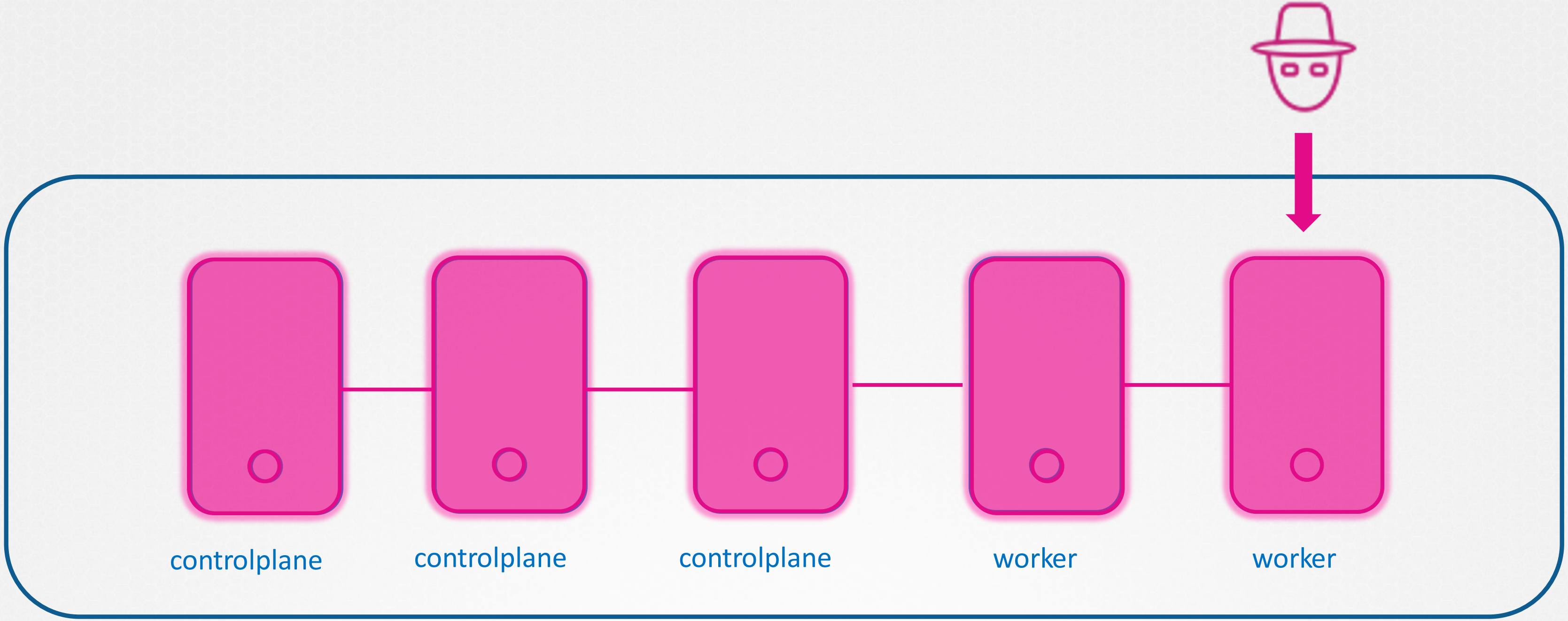
Securing Cluster

Sandboxing Techniques

Restricting Network Access

Minimizing Microservices Vulnerability

MTLS Encryption

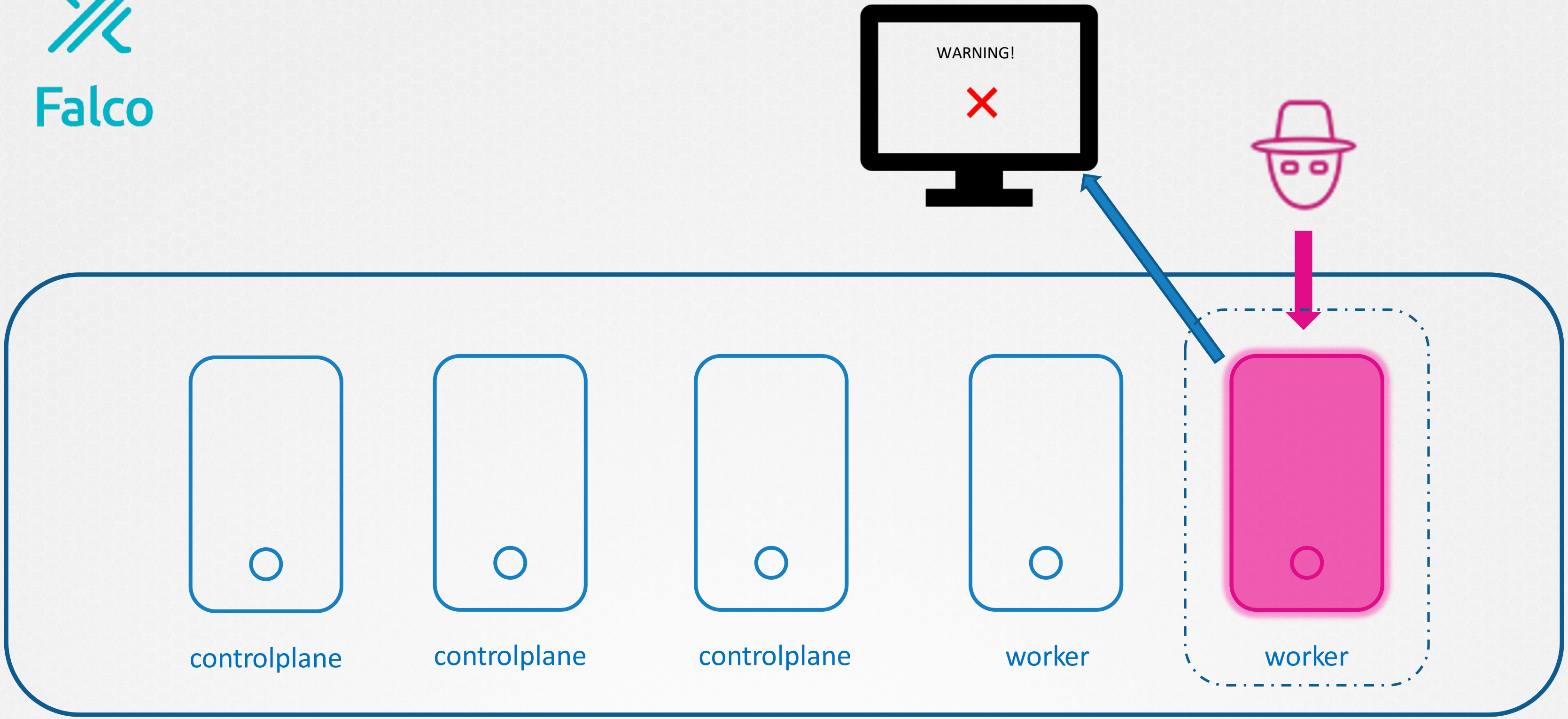

controlplane
controlplane
controlplane
worker
worker

Instant Notifications

Revert Transactions

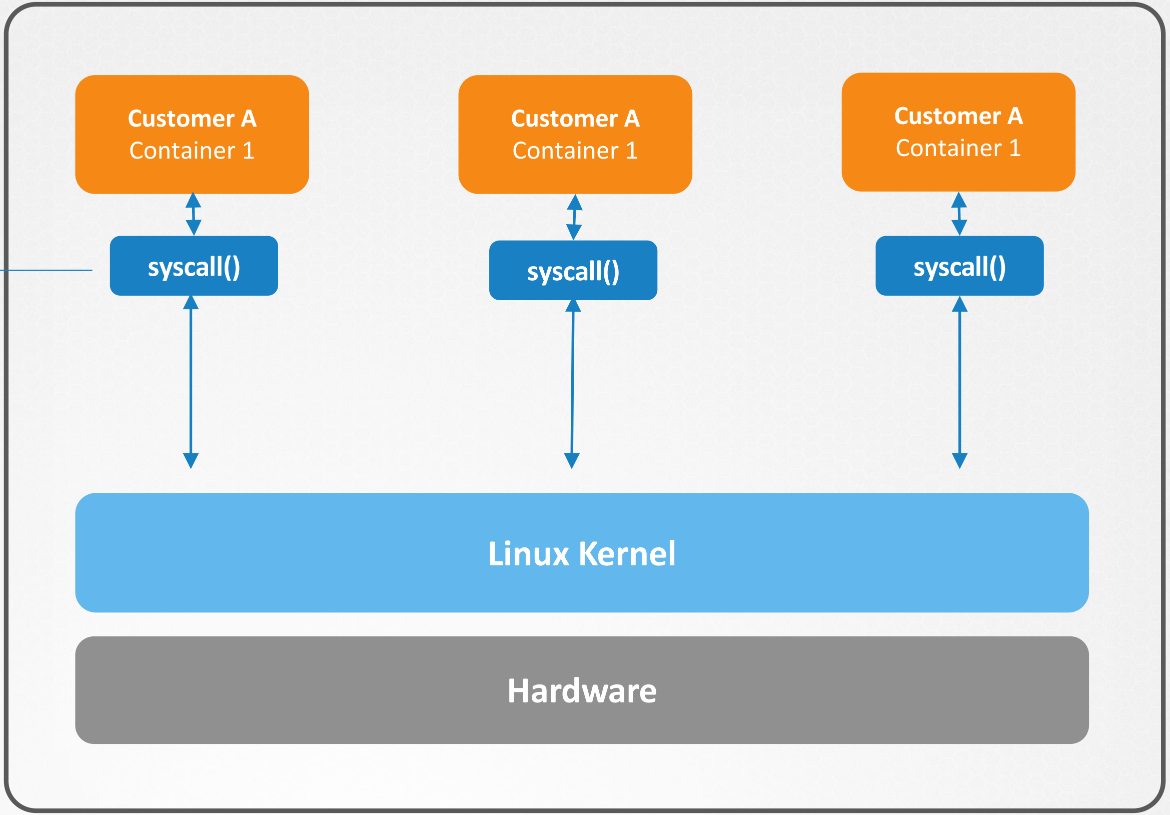
Transaction Limits

Falco

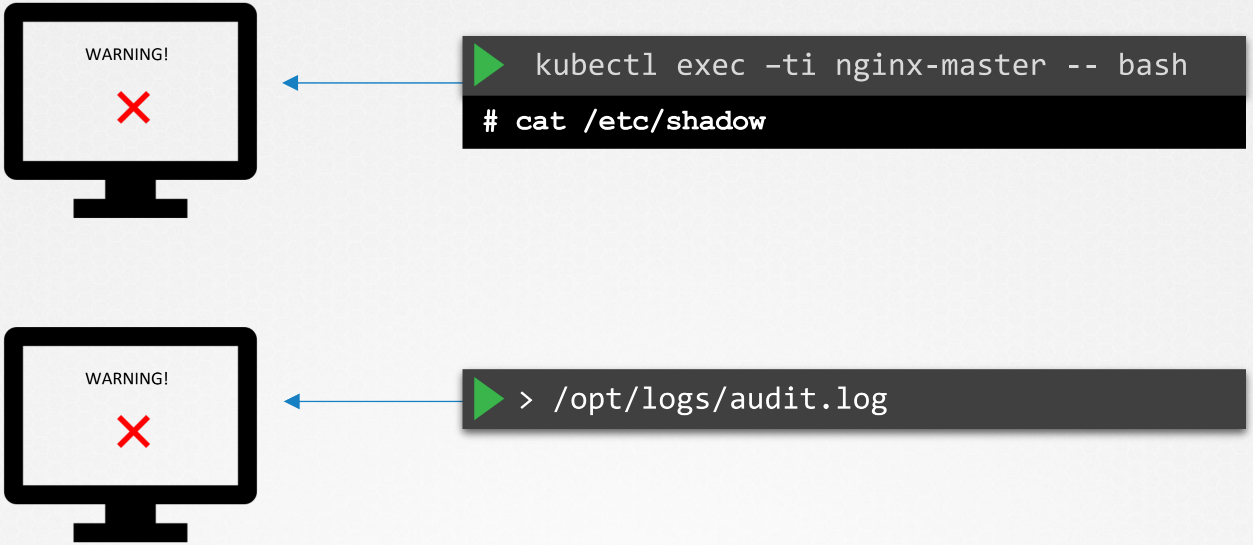

WARNING!
controlplane
controlplane
controlplane
worker
worker

Falco

SYSCALL NAME
close
nanosleep
fcntl
fstatfs
getdents64
exit_group
epoll_ctl
openat



Falco



```
kubectl exec –ti nginx-master -- bash
# cat /etc/shadow
```

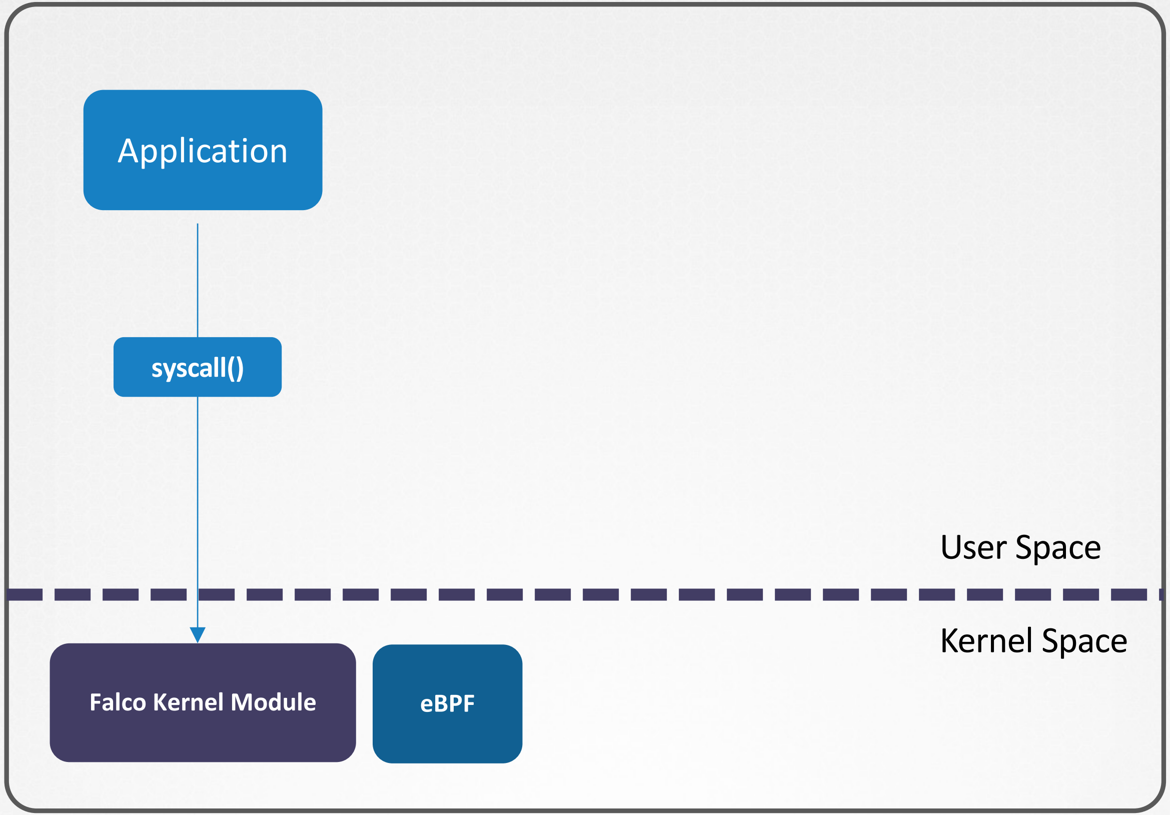
```
> /opt/logs/audit.log
```

www.kodekloud.com

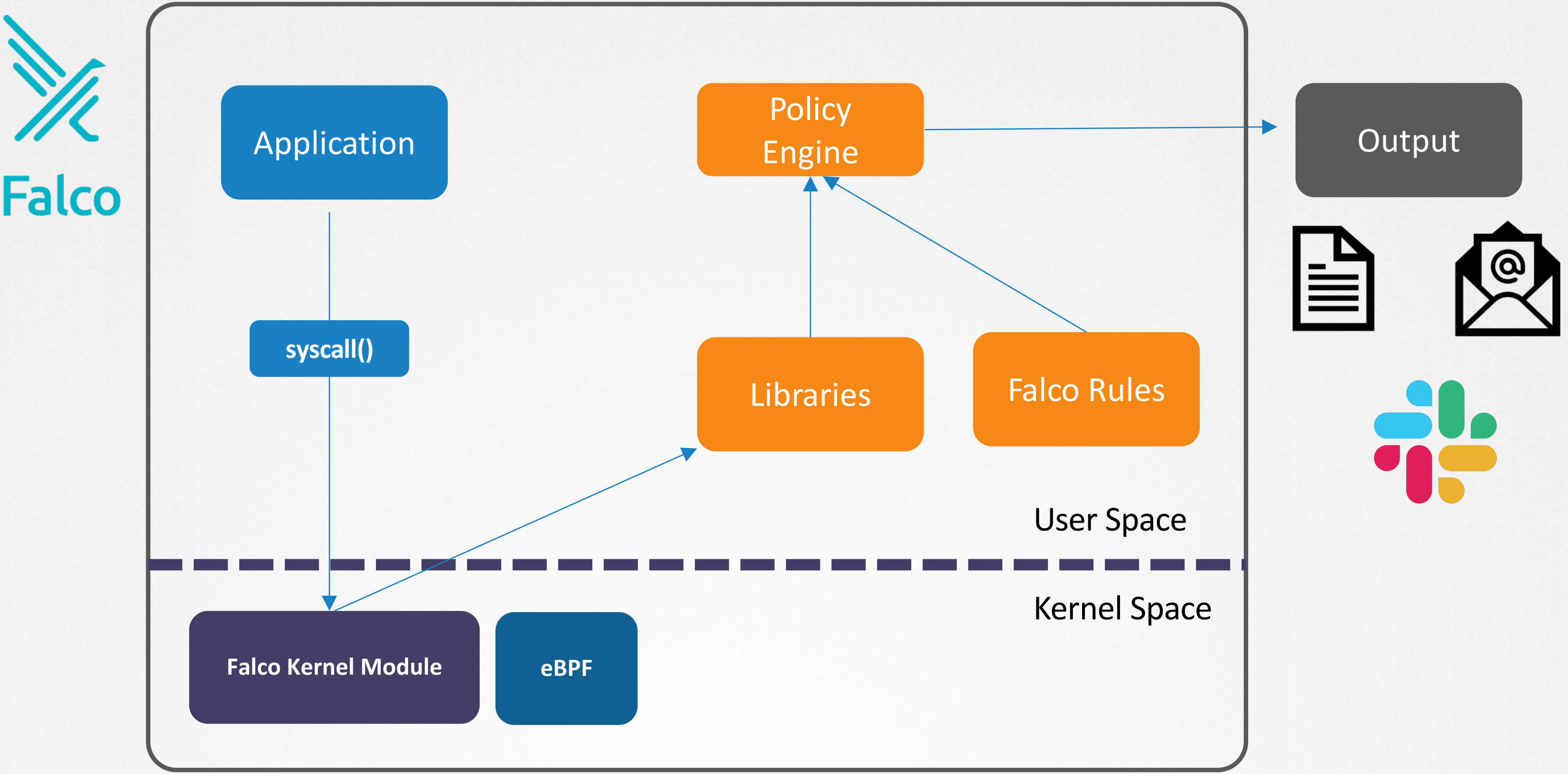
Falco Overview and Installation

Falco Architecture

Falco



Falco Architecture



Install as a Package

```
curl -s https://falco.org/repo/falcosecurity-3672BA8F.asc | apt-key add -
```

```
echo "deb https://download.falco.org/packages/deb stable main" | tee -a /etc/apt/sources.list.d/falcosecurity.list
```

```
apt update -y
```

```
apt get install –y linux-headers-$(uname –r)
```

```
apt install –y falco
```

```
systemctl start falco
```

https://falco.org/docs/getting-started/installation/

Install as a DaemonSet

```
helm repo add falcosecurity https://falcosecurity.github.io/charts
```

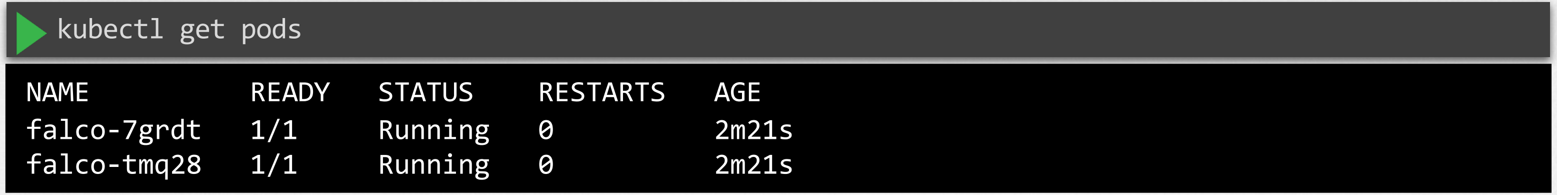
```
helm repo update
```

```
helm install falco falcosecurity/falco
```

```
NAME: falco
LAST DEPLOYED: Wed Mar  7 20:19:25 2021
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Falco agents are spinning up on each node in your cluster. After a few
seconds, they are going to start monitoring your containers looking for
security issues.


No further action should be required.
```

https://github.com/falcosecurity/charts/tree/master/falco

Install as a DaemonSet



```
kubectl get pods
```

```
NAME           READY    STATUS     RESTARTS    AGE
falco-7grdt    1/1      Running    0           2m21s
falco-tmq28    1/1      Running    0           2m21s
```

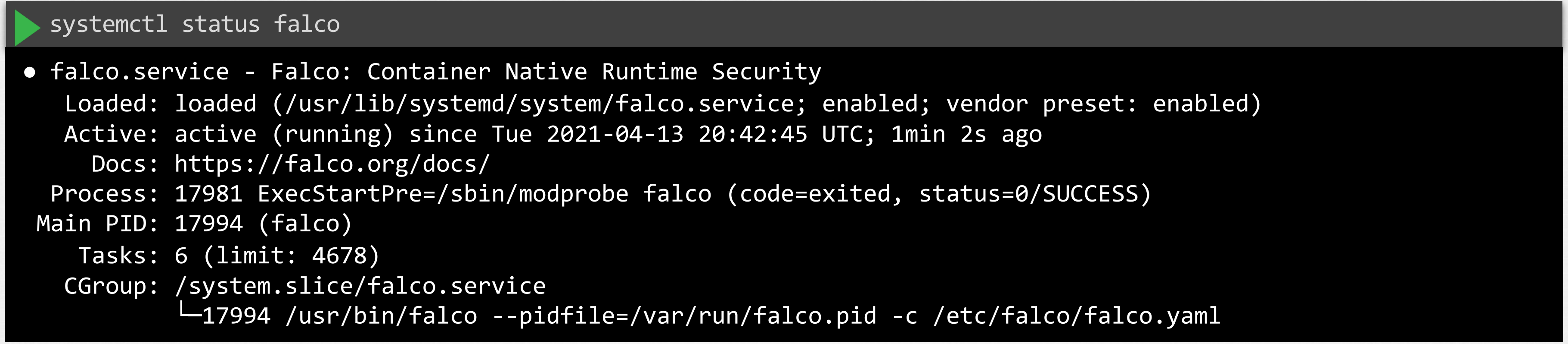
https://github.com/falcosecurity/charts/tree/master/falco

KODEKLOUD

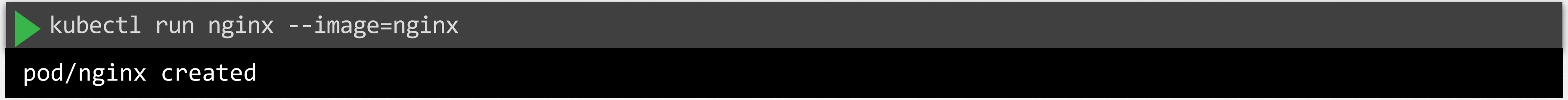
www.kodekloud.com

Use Falco to Detect Threats

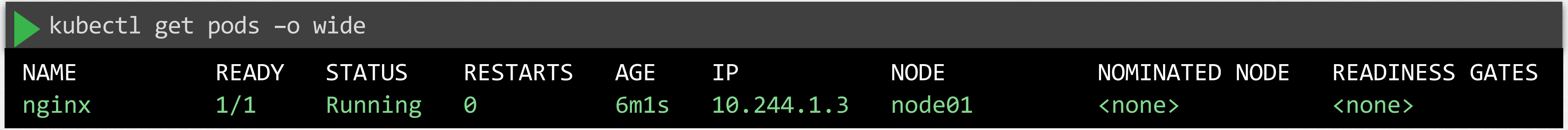
node01



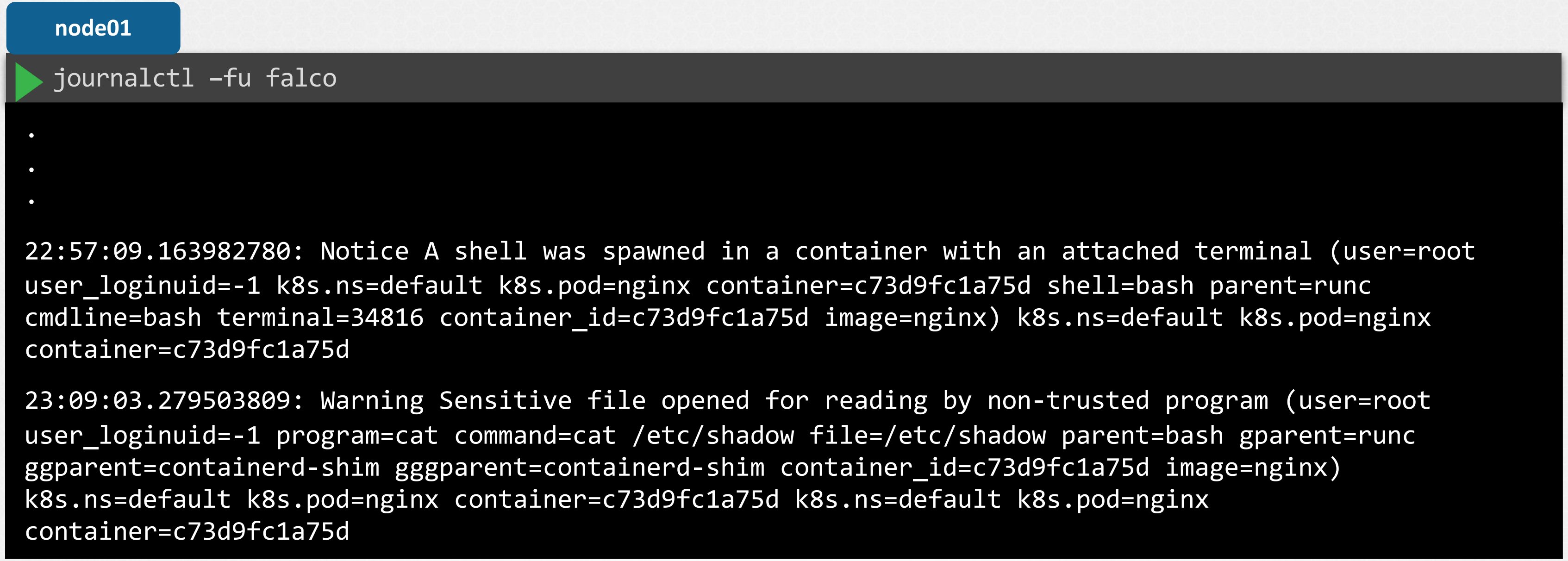
```
systemctl status falco

● falco.service - Falco: Container Native Runtime Security
   Loaded: loaded (/usr/lib/systemd/system/falco.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-04-13 20:42:45 UTC; 1min 2s ago
     Docs: https://falco.org/docs/
  Process: 17981 ExecStartPre=/sbin/modprobe falco (code=exited, status=0/SUCCESS)
 Main PID: 17994 (falco)
    Tasks: 6 (limit: 4678)
   CGroup: /system.slice/falco.service
           └─17994 /usr/bin/falco --pidfile=/var/run/falco.pid -c /etc/falco/falco.yaml
```



```
kubectl run nginx --image=nginx

pod/nginx created
```



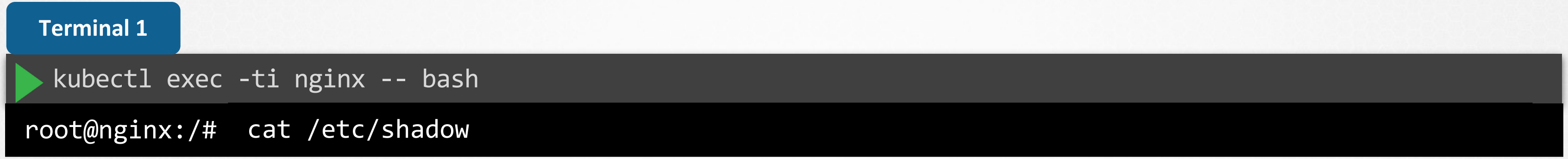
```
kubectl get pods -o wide

NAME          READY   STATUS    RESTARTS   AGE    IP            NODE      NOMINATED NODE   READINESS GATES
nginx         1/1     Running   0          6m1s   10.244.1.3    node01    <none>           <none>
```



node01

```
journalctl -fu falco
```

```
.
.
.

22:57:09.163982780: Notice A shell was spawned in a container with an attached terminal (user=root
user_loginuid=-1 k8s.ns=default k8s.pod=nginx container=c73d9fc1a75d shell=bash parent=runc
cmdline=bash terminal=34816 container_id=c73d9fc1a75d image=nginx) k8s.ns=default k8s.pod=nginx
container=c73d9fc1a75d

23:09:03.279503809: Warning Sensitive file opened for reading by non-trusted program (user=root
user_loginuid=-1 program=cat command=cat /etc/shadow file=/etc/shadow parent=bash gparent=runc
ggparent=containerd-shim gggparent=containerd-shim container_id=c73d9fc1a75d image=nginx)
k8s.ns=default k8s.pod=nginx container=c73d9fc1a75d k8s.ns=default k8s.pod=nginx
container=c73d9fc1a75d
```

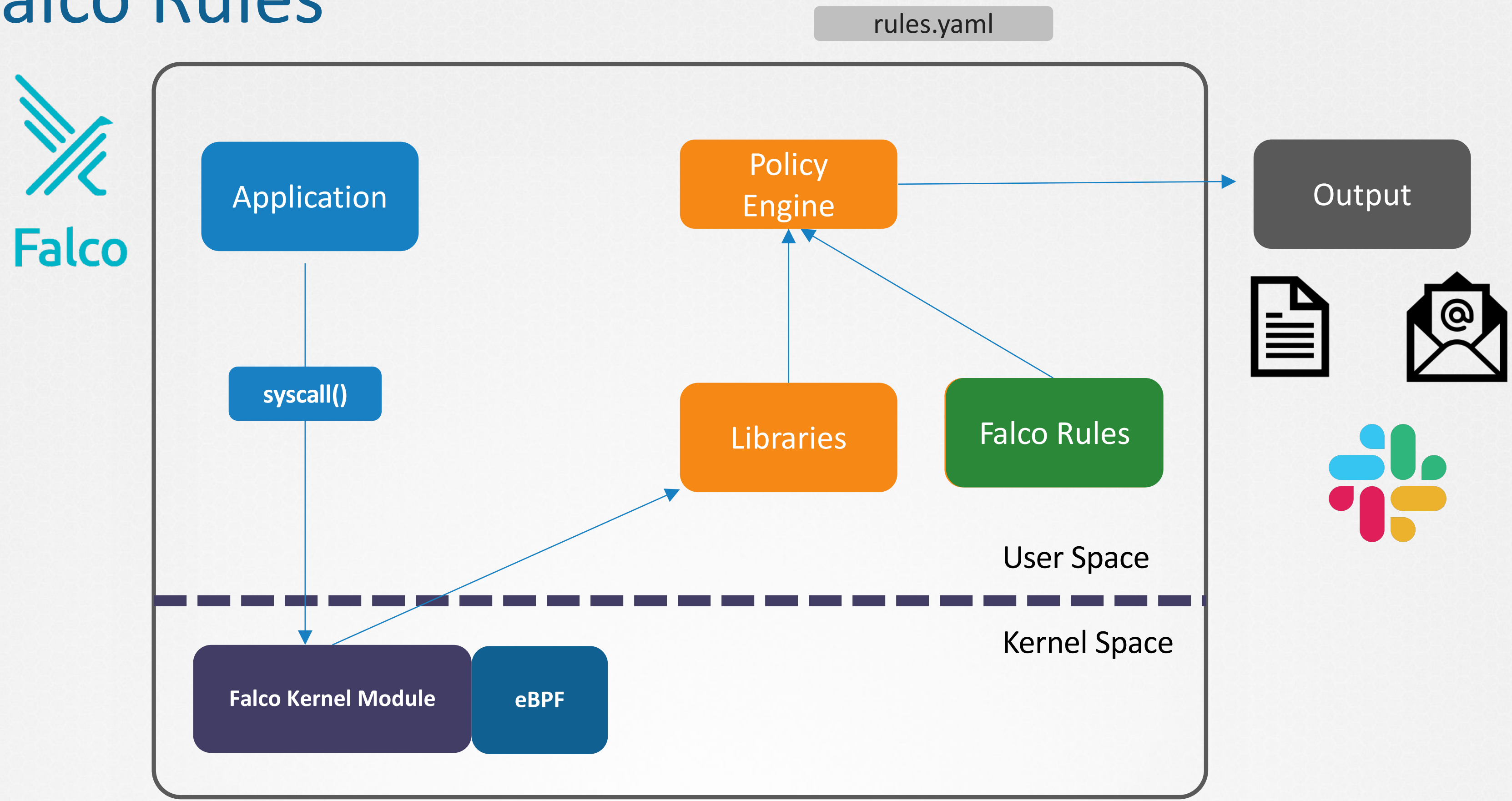


Terminal 1

```
kubectl exec -ti nginx -- bash
```

```
root@nginx:/#  cat /etc/shadow
```

Falco Rules


Falco
rules.yaml
Application
syscall()
Policy Engine
Libraries
Falco Rules
Output
User Space
Kernel Space
Falco Kernel Module
eBPF

Falco Rules

rules.yaml

```yaml
  - rule: <Name of the Rule>
    desc: <Detailed Description of the Rule>
    condition:   <When to filter events matching the rule>
    output:  <Output to be generated for the Event>
    priority:  <Severity of the event>
```

Falco Rules

rules.yaml

```yaml
  - rule:  Detect Shell inside a container
    desc:  Alert if a shell such as bash is open inside the container
    condition:  container.id != host and proc.name = bash
    output:   Bash Shell Opened (user=%user.name %container.id)
    priority: WARNING
```

Falco Rules

rules.yaml

```yaml
- rule:  Detect Shell inside a container
  desc:  Alert if a shell such as bash is open inside the container
  condition:  container.id != host and proc.name = bash
  output:   Bash Opened (user=%user.name container=%container.id)
  priority: WARNING
```

container.id

proc.name

fd.name

evt.type

user.name

container.image.repository

https://falco.org/docs/rules/supported-fields/

rules.yaml

```
- rule:  Detect Shell inside a container
  desc:  Alert if a shell such as bash is open inside the container
  condition:  container.id != host and proc.name = bash
  output:   Bash Opened (user=%user.name container=%container.id)
  priority: WARNING
```

container.id

proc.name

fd.name

evt.type

user.name

container.image.repository

DEBUG

INFORMATIONAL

NOTICE

WARNING

ERROR

CRITICAL

ALERT

EMERGENCY

https://falco.org/docs/rules/supported-fields/

rules.yaml

```yaml
  - rule:  Detect Shell inside a container
    desc:  Alert if a shell such as bash is open inside the container
    condition:  container.id != host and proc.name in (linux_shells)
    output:   Bash Opened (user=%user.name container=%container.id)
    priority: WARNING

  - list: linux_shells
    items: [bash, zsh, ksh, sh, csh]
```

https://falco.org/docs/rules/supported-fields/

rules.yaml

```yaml
- rule:  Detect Shell inside a container
  desc:  Alert if a shell such as bash is open inside the container
  condition: container           and proc.name in (linux_shells)
  output:   Bash Opened (user=%user.name container=%container.id)
  priority: WARNING

- list: linux_shells
  items: [bash, zsh, ksh, sh, csh]
```

```yaml
- macro: container
  condition: container.id != host
```

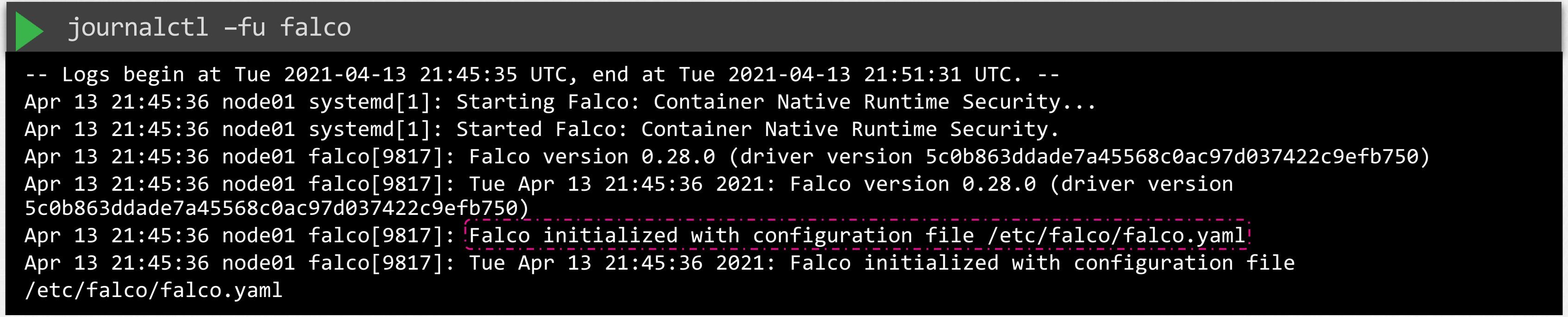
https://falco.org/docs/rules/default-macros/

KODEKLOUD

www.kodekloud.com

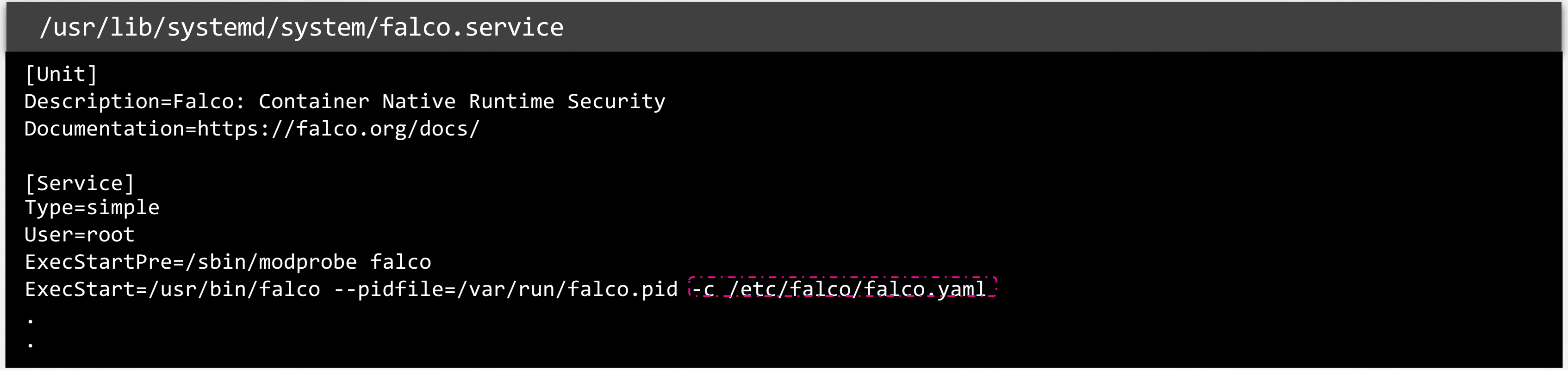
Falco Configuration Files

```
/etc/falco/falco.yaml
```



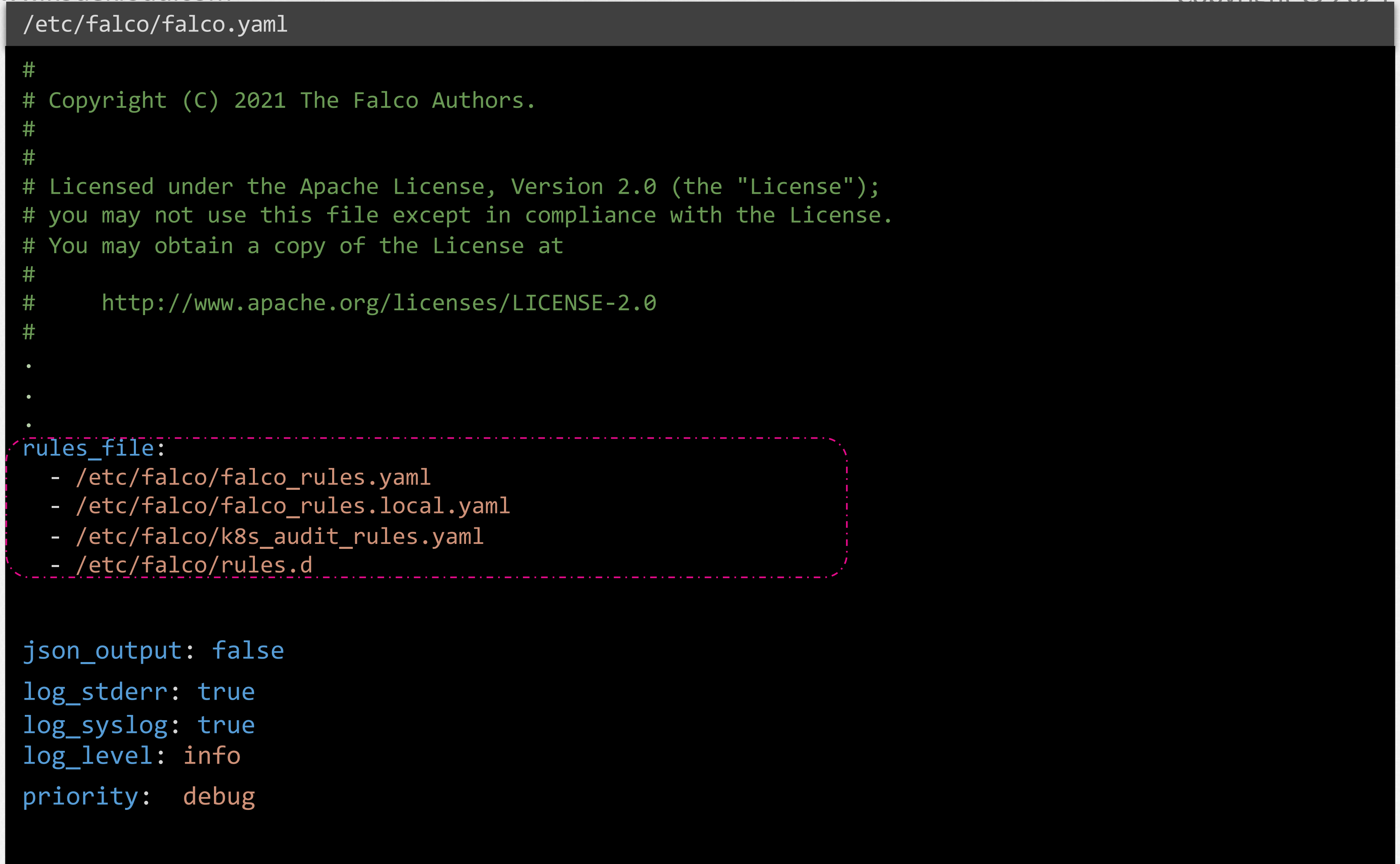
```
journalctl -fu falco
```

```
-- Logs begin at Tue 2021-04-13 21:45:35 UTC, end at Tue 2021-04-13 21:51:31 UTC. --
Apr 13 21:45:36 node01 systemd[1]: Starting Falco: Container Native Runtime Security...
Apr 13 21:45:36 node01 systemd[1]: Started Falco: Container Native Runtime Security.
Apr 13 21:45:36 node01 falco[9817]: Falco version 0.28.0 (driver version 5c0b863ddade7a45568c0ac97d037422c9efb750)
Apr 13 21:45:36 node01 falco[9817]: Tue Apr 13 21:45:36 2021: Falco version 0.28.0 (driver version
5c0b863ddade7a45568c0ac97d037422c9efb750)
Apr 13 21:45:36 node01 falco[9817]: Falco initialized with configuration file /etc/falco/falco.yaml
Apr 13 21:45:36 node01 falco[9817]: Tue Apr 13 21:45:36 2021: Falco initialized with configuration file
/etc/falco/falco.yaml
```



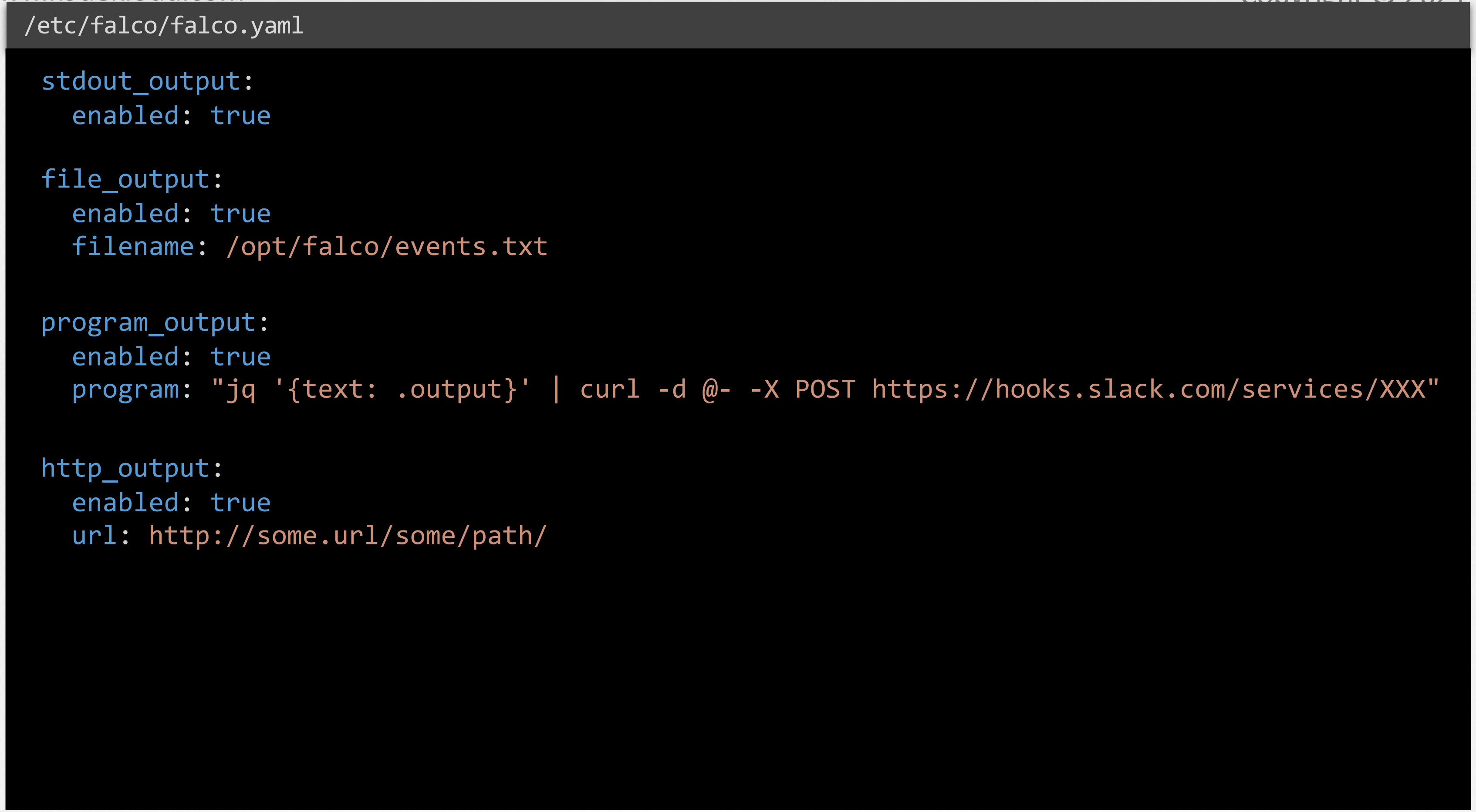
/usr/lib/systemd/system/falco.service

```
[Unit]
Description=Falco: Container Native Runtime Security
Documentation=https://falco.org/docs/

[Service]
Type=simple
User=root
ExecStartPre=/sbin/modprobe falco
ExecStart=/usr/bin/falco --pidfile=/var/run/falco.pid -c /etc/falco/falco.yaml
.
.
```



/etc/falco/falco.yaml

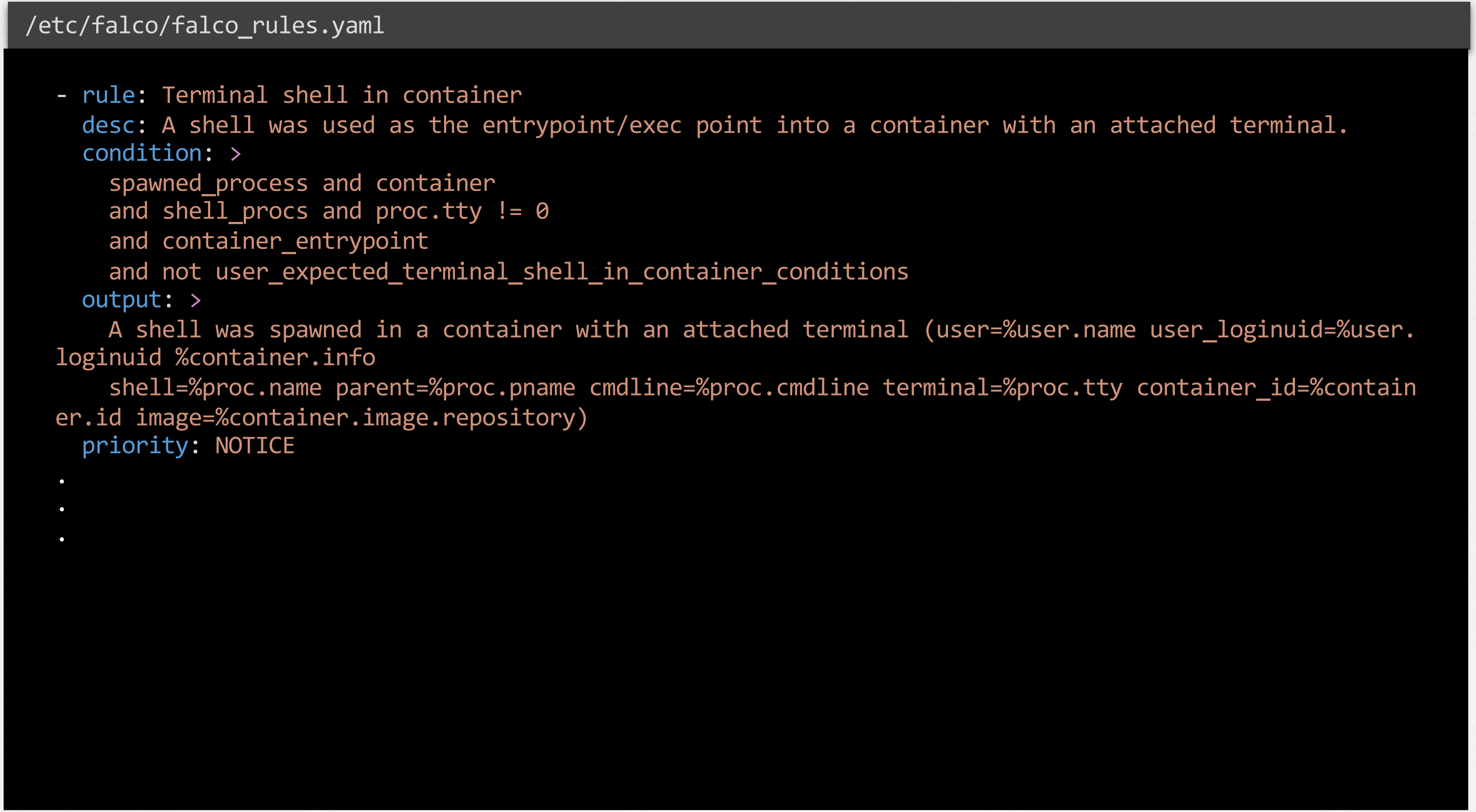
```yaml
#
# Copyright (C) 2021 The Falco Authors.
#
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
.
.
.
rules_file:
  - /etc/falco/falco_rules.yaml
  - /etc/falco/falco_rules.local.yaml
  - /etc/falco/k8s_audit_rules.yaml
  - /etc/falco/rules.d


json_output: false
log_stderr: true
log_syslog: true
log_level: info
priority:  debug
```



/etc/falco/falco.yaml

```yaml
stdout_output:
  enabled: true

file_output:
  enabled: true
  filename: /opt/falco/events.txt

program_output:
  enabled: true
  program: "jq '{text: .output}' | curl -d @- -X POST https://hooks.slack.com/services/XXX"

http_output:
  enabled: true
  url: http://some.url/some/path/
```

https://falco.org/docs/configuration/



/etc/falco/falco_rules.yaml

```
- rule: Terminal shell in container
  desc: A shell was used as the entrypoint/exec point into a container with an attached terminal.
  condition: >
    spawned_process and container
    and shell_procs and proc.tty != 0
    and container_entrypoint
    and not user_expected_terminal_shell_in_container_conditions
  output: >
    A shell was spawned in a container with an attached terminal (user=%user.name user_loginuid=%user.
loginuid %container.info
    shell=%proc.name parent=%proc.pname cmdline=%proc.cmdline terminal=%proc.tty container_id=%contain
er.id image=%container.image.repository)
  priority: NOTICE
.
.
.
```

/etc/falco/falco_rules.local.yaml

```
 - rule: Terminal shell in container
   desc: A shell was used as the entrypoint/exec point into a container with an attached terminal.
   condition: >
     spawned_process and container
     and shell_procs and proc.tty != 0
     and container_entrypoint
     and not user_expected_terminal_shell_in_container_conditions
   output: >
     A shell was spawned in a container with an attached terminal (user=%user.name user_loginuid=%user.
 loginuid %container.info
     shell=%proc.name parent=%proc.pname cmdline=%proc.cmdline terminal=%proc.tty container_id=%contain
 er.id image=%container.image.repository)
   priority  WARNING


- rule: Anomalous read in kodekloud/webapp pod
  desc: Detect Suspicious reads in custom webapp container
  condition: >
    open_read and container
    and container.image.repository="kodekloud/simple-webapp"
    and fd.directory != "/opt/app"
  output: >
    A file was opened and read outside the /opt/app directory(user=%user.name user_loginuid=%user.logi
nuid
    container_id=%container.id image=%container.image.repository)
  priority: CRITICAL
```

Hot Reload

```
cat /var/run/falco.pid
7183
```

```
kill -1 $(cat /var/run/falco.pid)
```

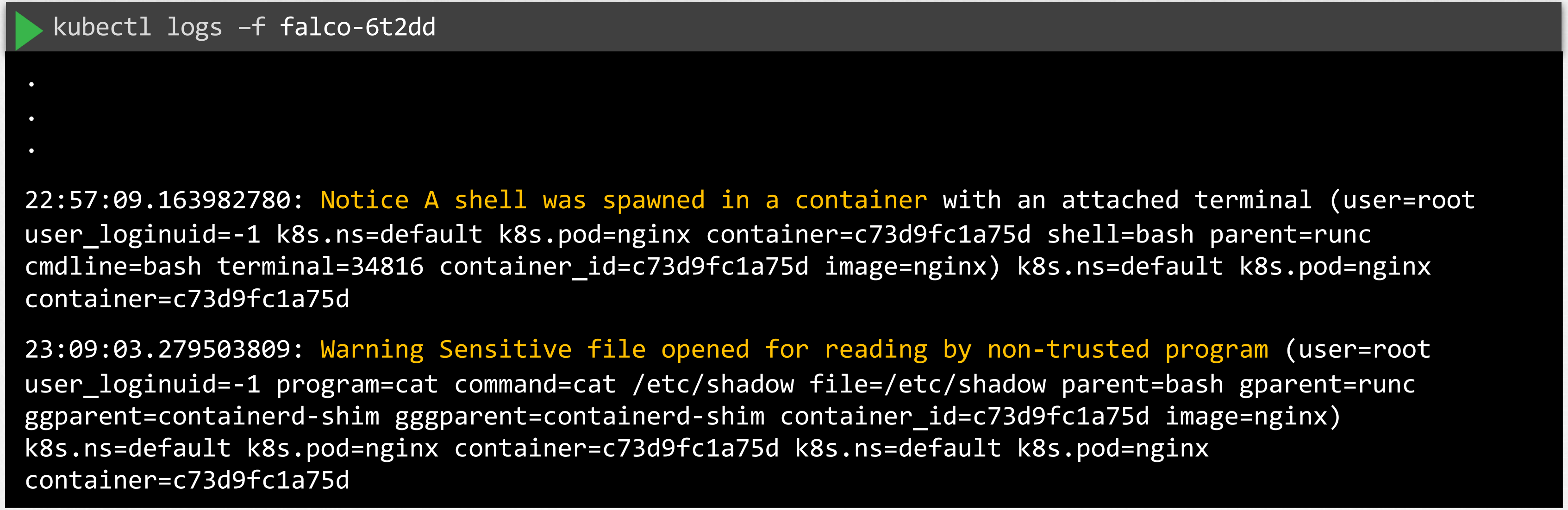
Hands-on Labs

cks.kodekloud.com

KODEKLOUD

www.kodekloud.com

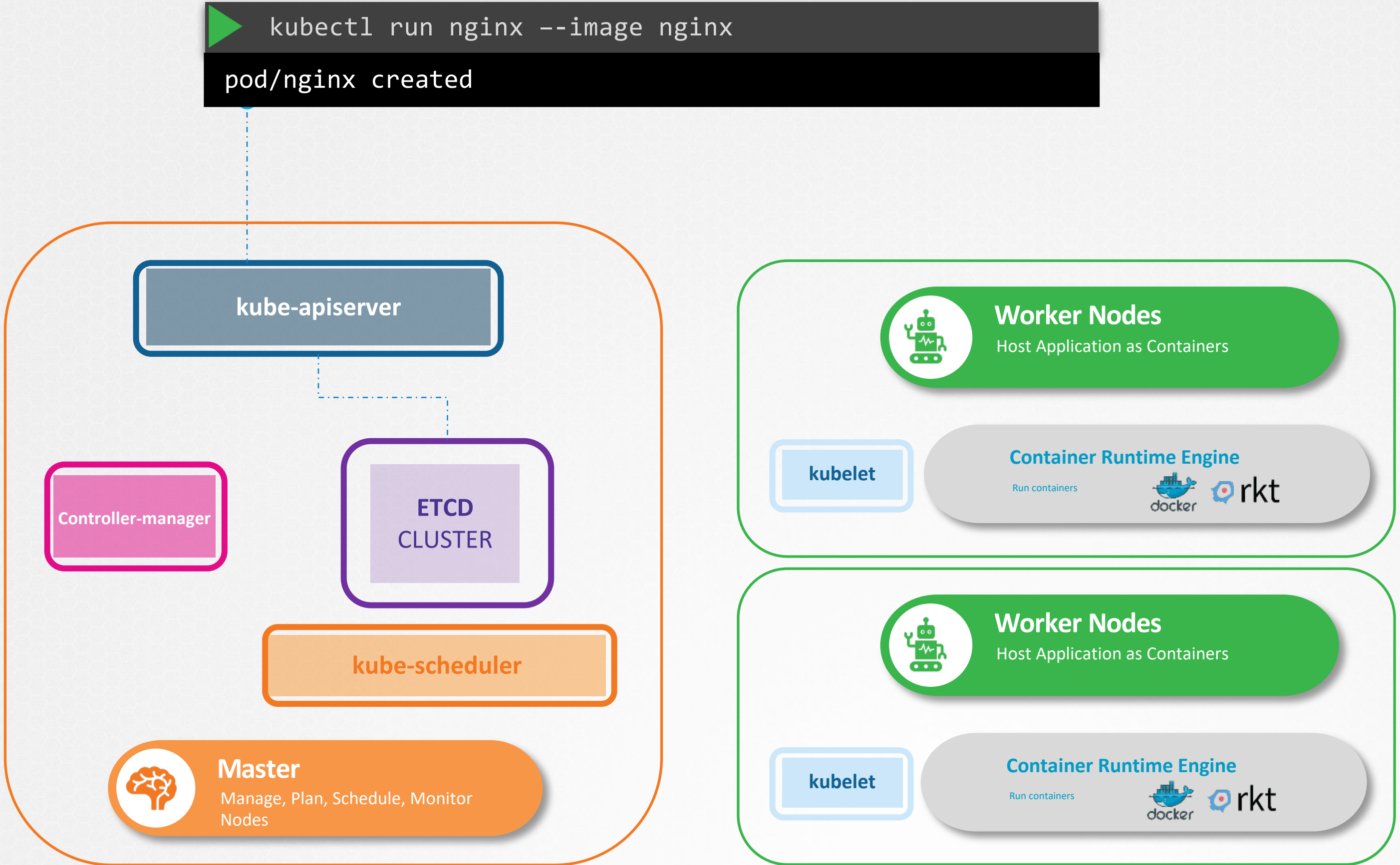
Kubernetes Auditing



```
kubectl logs -f falco-6t2dd
.
.
.
22:57:09.163982780: Notice A shell was spawned in a container with an attached terminal (user=root
user_loginuid=-1 k8s.ns=default k8s.pod=nginx container=c73d9fc1a75d shell=bash parent=runc
cmdline=bash terminal=34816 container_id=c73d9fc1a75d image=nginx) k8s.ns=default k8s.pod=nginx
container=c73d9fc1a75d

23:09:03.279503809: Warning Sensitive file opened for reading by non-trusted program (user=root
user_loginuid=-1 program=cat command=cat /etc/shadow file=/etc/shadow parent=bash gparent=runc
ggparent=containerd-shim gggparent=containerd-shim container_id=c73d9fc1a75d image=nginx)
k8s.ns=default k8s.pod=nginx container=c73d9fc1a75d k8s.ns=default k8s.pod=nginx
container=c73d9fc1a75d
```

pod

namespace



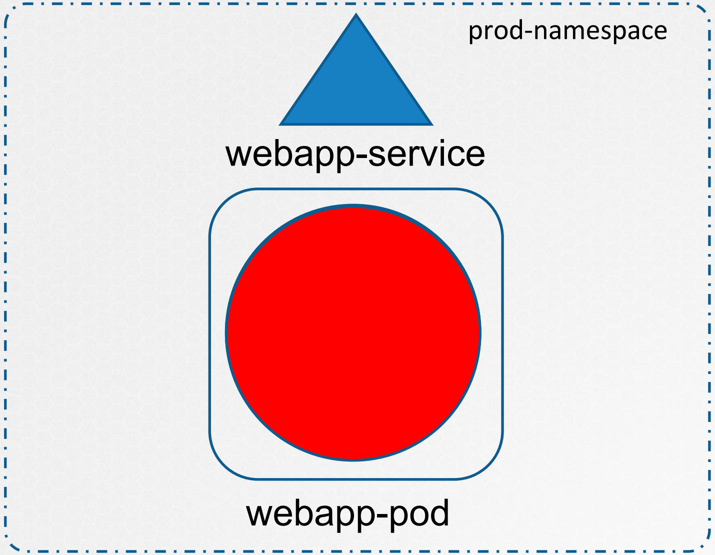
```
kubectl run nginx --image nginx
pod/nginx created
```

1. RequestReceived

2. RequestStarted

3. RequestComplete

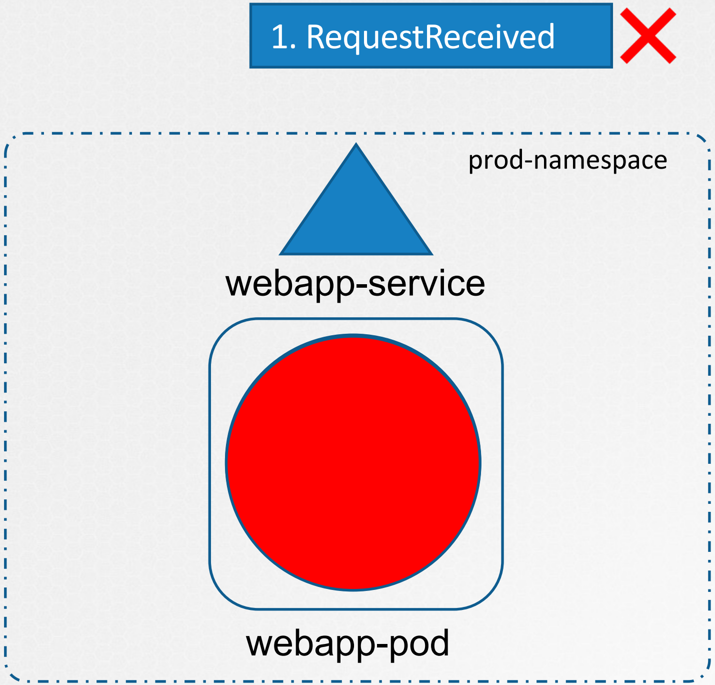
4. Panic



```
webapp-pod deleted in prod-namespace!
```

audit-policy.yaml

```yaml
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages:
rules:
```

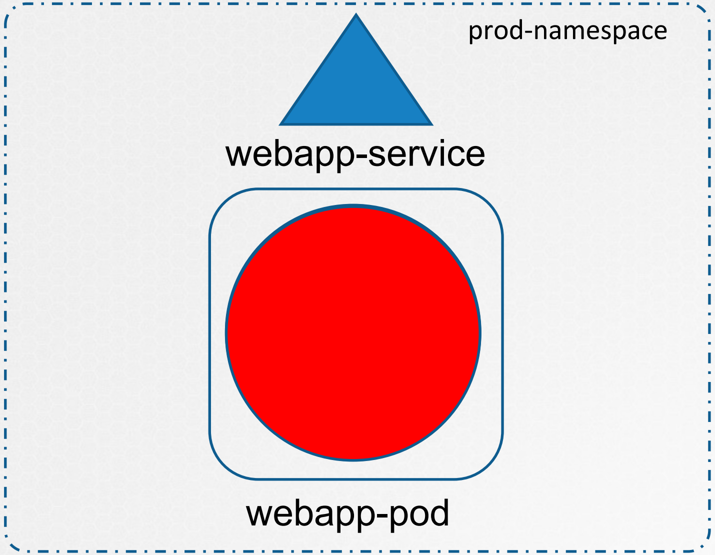


```
webapp-pod deleted in prod-namespace!
```

audit-policy.yaml

```yaml
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages: ["RequestReceived"]

rules:
  - namespace: ["prod-namespace"]
```

1. RequestReceived

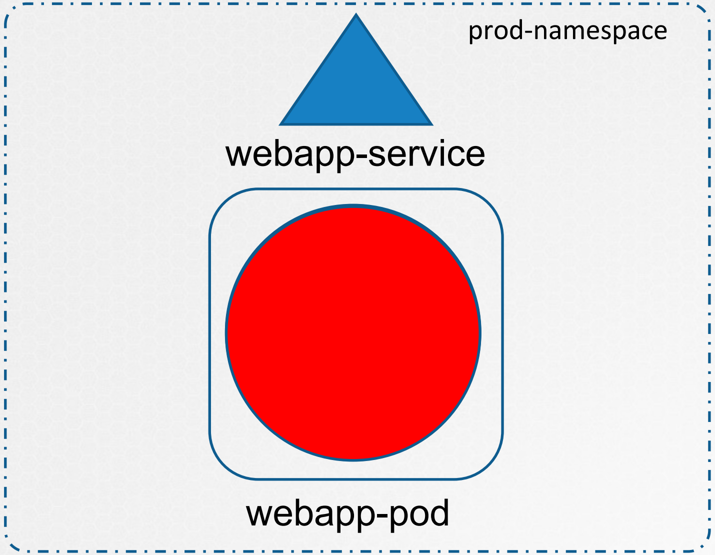


```
webapp-pod deleted in prod-namespace!
```

audit-policy.yaml

```yaml
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages: ["RequestReceived"]
rules:
  - namespace: ["prod-namespace"]

    verb: ["delete"]
```

1. RequestReceived ✗

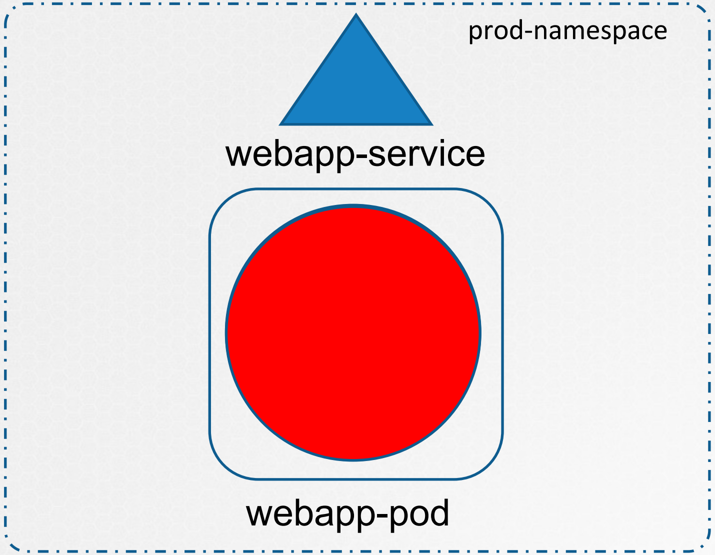


```
webapp-pod deleted in prod-namespace!
```

audit-policy.yaml

```
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages: ["RequestReceived"]
rules:
  - namespace: ["prod-namespace"]
    verb: ["delete"]
    resources:
    - groups: " "
      resources: ["pods"]
```

1. RequestReceived ❌

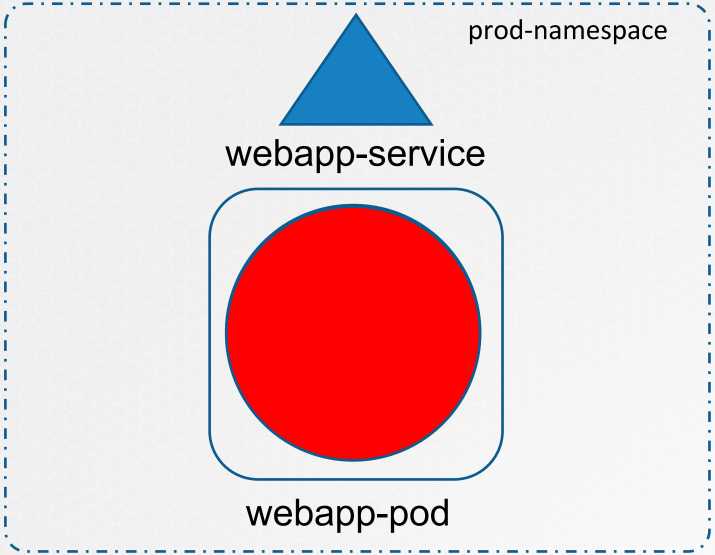


```
webapp-pod deleted in prod-namespace!
```

audit-policy.yaml

```yaml
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages: ["RequestReceived"]
rules:
  - namespace: ["prod-namespace"]
    verb: ["delete"]
    resources:
    - groups: " "
      resources: ["pods"]
      resourceNames: ["webapp-pod"]
    level: RequestResponse
```

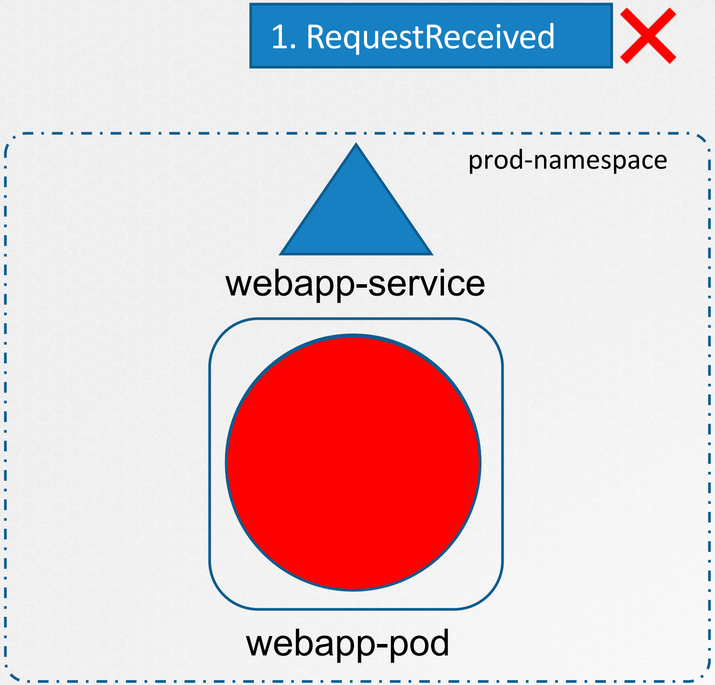
1. RequestReceived ❌



```
webapp-pod deleted in prod-namespace!
```

audit-policy.yaml

```
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages: ["RequestReceived"]
rules:
  - namespace: ["prod-namespace"]
    verb: ["delete"]
    resources:
    - groups: " "
      resources: ["pods"]
      resourceNames: ["webapp-pod"]
    level:  RequestResponse

  - level: Metadata
    resources:
    - groups: " "
      resources: ["secrets"]
```



```
webapp-pod deleted in prod-namespace!
```

audit-policy.yaml

```
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages: ["RequestReceived"]
rules:
  - namespace: ["prod-namespace"]
    verb: ["delete"]
    resources:
    - groups: " "
      resources: ["pods"]
      resourceNames: ["webapp-pod"]
    level:  RequestResponse

  - level: Metadata
    resources:
    - groups: " "
      resources: ["secrets"]
```

Falco

kube-apiserver.service

```
ExecStart=/usr/local/bin/kube-apiserver \\
  --advertise-address=${INTERNAL_IP} \\
  --allow-privileged=true \\
  --apiserver-count=3 \\
  --authorization-mode=Node,RBAC \\
  --bind-address=0.0.0.0 \\
  --enable-swagger-ui=true \\
  --etcd-servers=https://127.0.0.1:2379 \\
  --event-ttl=1h \\
  --runtime-config=api/all \\
  --service-cluster-ip-range=10.32.0.0/24 \\
  --service-node-port-range=30000-32767 \\
  --v=2
  --audit-log-path=/var/log/k8-audit.log
  --audit-policy-file=/etc/kubernetes/audit-policy.yaml
```

/etc/kubernetes/manifests/kube-apiserver.yaml

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-apiserver
    - --authorization-mode=Node,RBAC
    - --advertise-address=172.17.0.107
    - --allow-privileged=true
    - --enable-bootstrap-token-auth=true
    - --audit-log-path=/var/log/k8-audit.log
    - --audit-policy-file=/etc/kubernetes/audit-policy.yaml
```

Falco

/etc/kubernetes/manifests/kube-apiserver.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  name: kube-apiserver
  namespace: kube-system
spec:
  containers:
  - command:
    - kube-apiserver
    - --authorization-mode=Node,RBAC
    - --advertise-address=172.17.0.107
    - --allow-privileged=true
    - --enable-bootstrap-token-auth=true
    - --audit-log-path=/var/log/k8-audit.log
    - --audit-policy-file=/etc/kubernetes/audit-policy.yaml
    - --audit-log-maxage=10
    - --audit-log-maxbackup=5
    - --audit-log-maxsize=100
```

kube-apiserver.service

```
ExecStart=/usr/local/bin/kube-apiserver \\
  --advertise-address=${INTERNAL_IP} \\
  --allow-privileged=true \\
  --apiserver-count=3 \\
  --authorization-mode=Node,RBAC \\
  --bind-address=0.0.0.0 \\
  --enable-swagger-ui=true \\
  --etcd-servers=https://127.0.0.1:2379 \\
  --event-ttl=1h \\
  --runtime-config=api/all \\
  --service-cluster-ip-range=10.32.0.0/24 \\
  --service-node-port-range=30000-32767 \\
  --v=2
  --audit-log-path=/var/log/k8-audit.log
  --audit-policy-file=/etc/kubernetes/audit-policy.yaml
  --audit-log-maxage=10
  --audit-log-maxbackup=5
  --audit-log-maxsize=100
```

audit-policy.yaml

```yaml
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages:
  - "RequestReceived"
rules:
  - level: Metadata
    namespace: ["prod-namespace"]
    verb: ["delete"]
    resources:
    - group: ""
      resources: ["pods"]
```

```
{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Metadata","auditID":"da2ad1a3-df15-4b10-a44d-
79e73d7ec3c0","stage":"ResponseComplete","requestURI":"/api/v1/namespaces/prod-namespace/pods/webapp-pod","verb":"delete",
"user":{"username":"kubernetes-admin","groups":["system:masters","system:authenticated"]},"sourceIPs":["172.17.0.36"],
"userAgent":"kubectl/v1.19.0 (linux/amd64) kubernetes/e199641","objectRef":{"resource":"pods","namespace":"prod-namespace",
"name":"webapp-pod","apiVersion":"v1"},"responseStatus":{"metadata":{},"code":200},
"requestReceivedTimestamp":"2021-04-12T05:15:24.182178Z","stageTimestamp":"
```

Hands-on Labs

cks.kodekloud.com

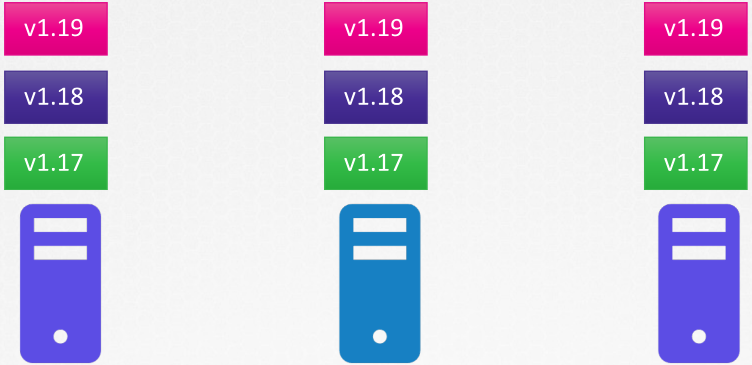
KODEKLOUD

www.kodekloud.com

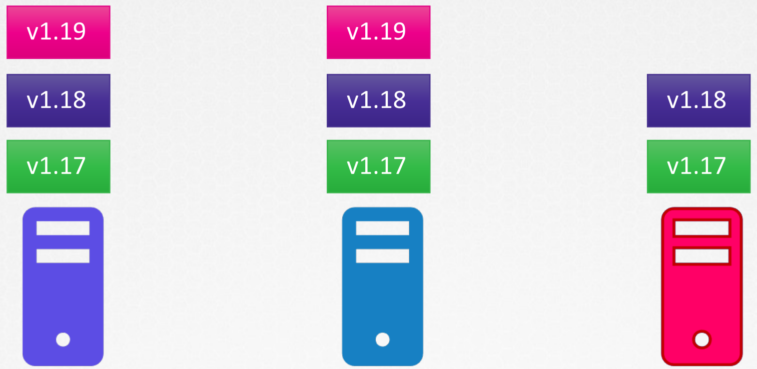
Immutable Infrastructure

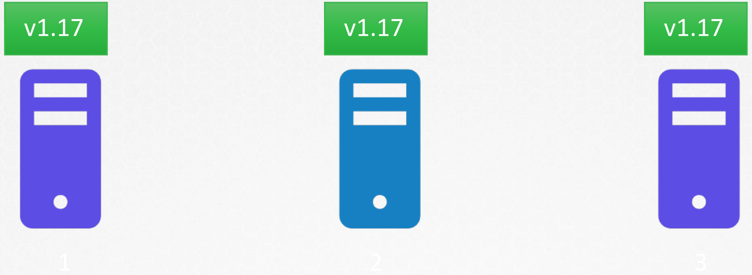
v1.19
v1.18
v1.17
Scripts
ANSIBLE

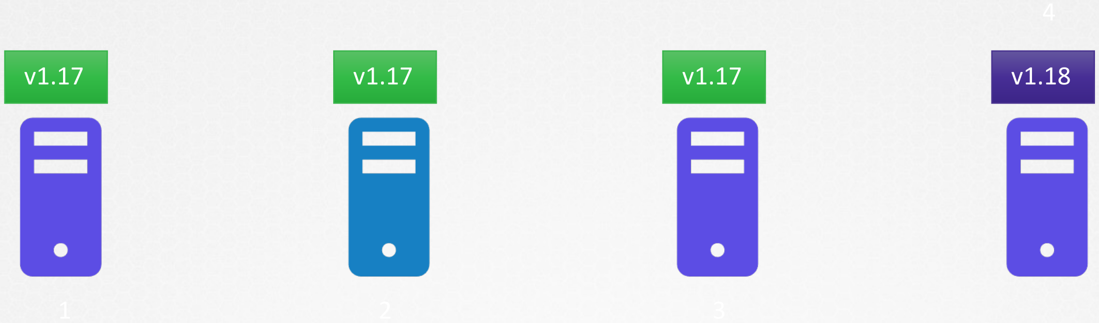
Mutable Infrastructure

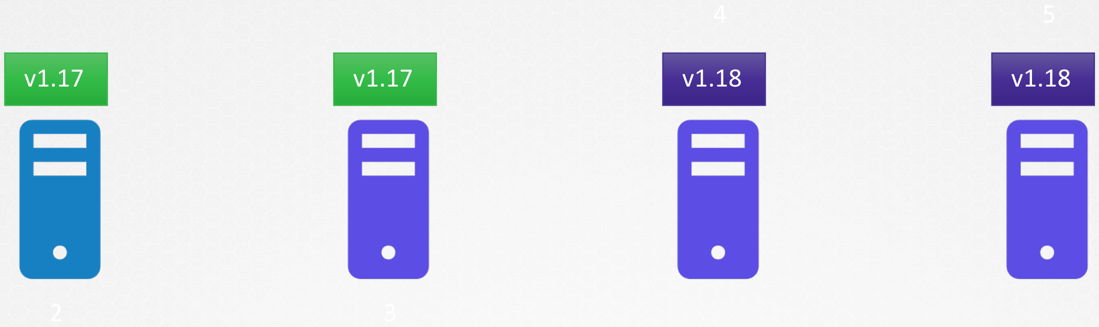


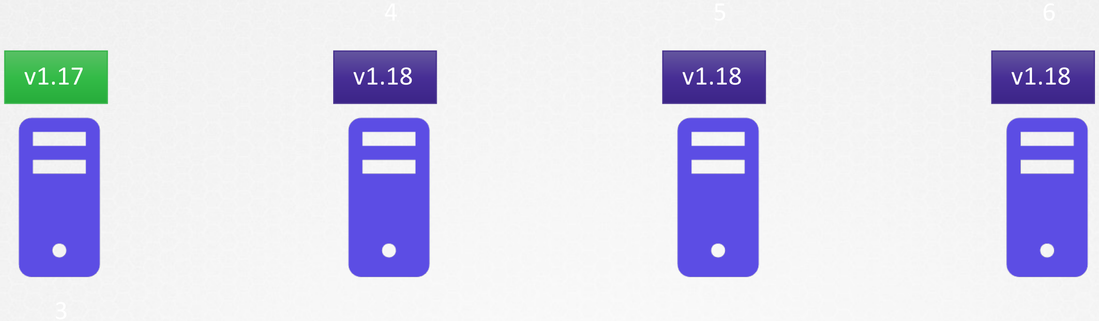
Configuration Drift




v1.17
v1.17
v1.17
1
2
3

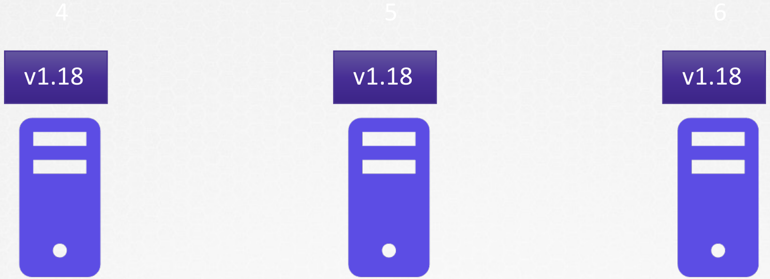

4
v1.17
v1.17
v1.17
v1.18
1
2
3


4
5
v1.17
v1.17
v1.18
v1.18
2
3



4
5
6
v1.17
v1.18
v1.18
v1.18
3

Immutable Infrastructure



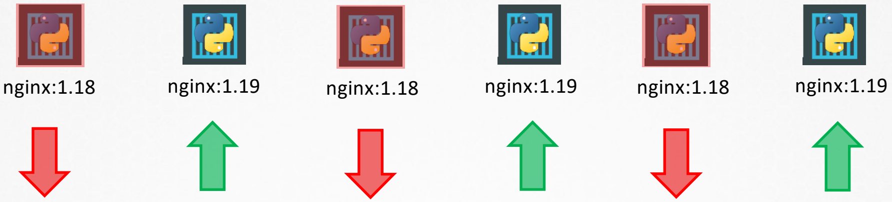
Immutable Infrastructure

Dockerfile – My Custom Webapp

```
FROM nginx:1.19

COPY nginx.conf /etc/nginx

ENTRYPOINT ["sh", "entrypoint.sh"]
```

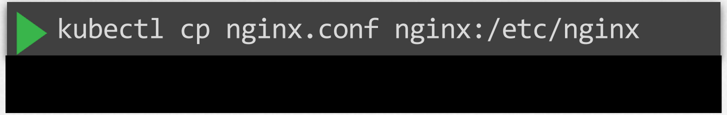
```
▶ kubectl cp nginx.conf nginx:/etc/nginx
```

Rolling Update

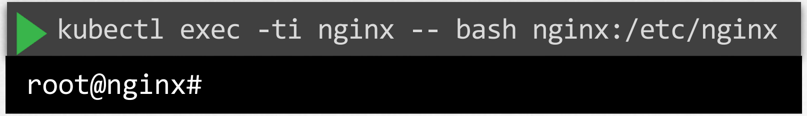


www.kodekloud.com

Ensure Immutability of Containers at Runtime



```
kubectl cp nginx.conf nginx:/etc/nginx
```



```
kubectl exec -ti nginx -- bash nginx:/etc/nginx
root@nginx#
```

```
▶ kubectl create –f nginx.yaml
pod/nginx created
```

```
▶ kubectl get pods
NAME     READY    STATUS    RESTARTS    AGE
nginx    0/1      Error     0           20s
```

nginx.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx
    securityContext:
          readOnlyRootFilesystem: true
```

```
▶ kubectl create -f nginx.yaml
pod/nginx created
```

```
▶ kubectl get pods
NAME     READY    STATUS    RESTARTS    AGE
nginx    0/1      Error     0           20s
```

```
▶ kubectl logs nginx
root@controlplane:~# kubectl logs nginx
.
.
2021/04/12 15:14:39 [emerg] 1#1: mkdir()
"/var/cache/nginx/client_temp" failed (30: Read-only
file system)
.
2021/04/12 16:11:26 [emerg] 1#1: open()
"/var/run/nginx.pid" failed (30: Read-only file system)
nginx: [emerg] open() "/var/run/nginx.pid" failed (30:
Read-only file system)
```

nginx.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx

    securityContext:
          readOnlyRootFilesystem: true
```

```
▶ kubectl create –f nginx.yaml
pod/nginx created
```

```
▶ kubectl get pods
NAME     READY    STATUS    RESTARTS    AGE
nginx    0/1      Running   0           20s
```

```
▶ kubectl logs nginx
root@controlplane:~# kubectl logs nginx
.
.
2021/04/12 15:14:39 [emerg] 1#1: mkdir()
"/var/cache/nginx/client_temp" failed (30: Read-only
file system)
.
2021/04/12 16:11:26 [emerg] 1#1: open()
"/var/run/nginx.pid" failed (30: Read-only file system)
nginx: [emerg] open() "/var/run/nginx.pid" failed (30:
Read-only file system)
```

nginx.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx

    securityContext:
          readOnlyRootFilesystem: true
    volumeMounts:
        - name: cache-volume
          mountPath: /var/cache/nginx
        - name: runtime-volume
          mountPath: /var/run

    volumes:
      - name: cache-volume
        emptyDir: {}
      - name: runtime-volume
        emptyDir: {}
```

```
▶ kubectl cp nginx.conf nginx:/etc/nginx
```

```
tar: nginx.yaml: Cannot open: Read-only file system
tar: Exiting with failure status due to previous errors
command terminated with exit code 2
```

```
▶ kubectl exec -ti nginx – apt update
```

```
Reading package lists... Done
E: List directory /var/lib/apt/lists/partial is missing. - Acquire (30: Read-only file system)
command terminated with exit code 100
```

```
▶ kubectl create –f nginx.yaml
pod/nginx created
```

```
▶ kubectl get pods
NAME     READY    STATUS    RESTARTS    AGE
nginx    1/1      Running   0           20s
```

```
▶ kubectl exec -ti nginx – apt update
Reading package lists... Done
E: List directory /var/lib/apt/lists/partial is missing. - Acquire
(30: Read-only file system)
command terminated with exit code 100
```

nginx.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx

    securityContext:
      readOnlyRootFilesystem: true
     privileged: true
    volumeMounts:
      - name: cache-volume
        mountPath: /var/cache/nginx
      - name: runtime-volume
        mountPath: /var/run
   volumes:
   - name: cache-volume
     emptyDir: {}
   - name: runtime-volume
     emptyDir: {}
```

```
kubectl create –f nginx.yaml
```

```
pod/nginx created
```

```
kubectl get pods
```

```
NAME     READY    STATUS     RESTARTS    AGE
nginx    1/1      Running    0           20s
```

```
kubectl exec -ti nginx – apt update
```

```
Reading package lists... Done
E: List directory /var/lib/apt/lists/partial is missing. - Acquire
(30: Read-only file system)
command terminated with exit code 100
```

```
kubectl exec -ti nginx -- cat /proc/sys/vm/swappiness
```

```
60
```

```
kubectl exec -ti nginx -- bash -c "echo '75' > /proc/sys/vm/swappiness"
```

```
kubectl exec -ti nginx -- cat /proc/sys/vm/swappiness
```

```
75
```

node01

```
cat /proc/sys/vm/swappiness
```

```
75
```

nginx.yaml

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx


     securityContext:
       readOnlyRootFilesystem: true
       privileged: true
     volumeMounts:
       - name: cache-volume
         mountPath: /var/cache/nginx
       - name: runtime-volume
         mountPath: /var/run
   volumes:
   - name: cache-volume
     emptyDir: {}
   - name: runtime-volume
     emptyDir: {}
```

readOnlyRootFilesystem: false ✗

Privileged: true ✗

runAsUser: 0 ✗

psp.yaml

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: example
spec:
  privileged: false
  readOnlyRootFilesystem: true
  runAsUser:
    rule: RunAsNonRoot
  seLinux:
    rule: RunAsAny
  supplementalGroups:
    rule: RunAsAny
  runAsUser:
    rule: RunAsNonRoot
  fsGroup:
    rule: RunAsAny
```

Hands-on Labs

cks.kodekloud.com

KODEKLOUD

www.kodekloud.com