Kubernetes Workshop Series

JTC16

Kubernetes Security Basic Concepts

Niklaus Hirt
DevOps Architect / Cloud Architect
nikh@ch.ibm.com

Welcome to the

Kubernetes Workshop Series

Housekeeping

Meeting is being recorded to be shared on Social Media

Meeting Mute All: Unmute to speak

Breaks: every 60mins (interrupt me if I forget ;-)

Questions:

- In Slack # (not in Webex!)
- Adressed at the end of the Module
- Additional questions: unmute to speak

We will monitor the Slack channel during the Labs
→ Feel free to answer other participants questions

Who am I?

Niklaus Hirt

Passionate about tech for over 35 years

- High-school in Berne
- Degree in Computer Science at EPFL
- ELCA
- CAST
- IBM

✉ nikh@ch.ibm.com
@nhirt

Agenda – Kubernetes Security - Basic Concepts

Module 0: Prepare the Labs
Module 1: Security Introduction
Module 2: Securing Workloads
Module 3: DevSecOps
Module 4: Hands-On Lab

Videos, sources and documentation will be available here:

All Workshop Recordings
https://www.youtube.com/channel/UCIS0jmGOQrG2AKKPkTJYj9w/videos

Recording from JTC02
https://www.youtube.com/watch?v=uDX5HZwq18U

https://github.com/niklaushirt/k8s_training_public

https://github.com/niklaushirt/training

Session Quiz & Feedback

We will collect some feedback and run a quiz or two.

Please make sure you can access https://kahoot.it/
either on your PC of Phone.

You will get the Game PIN
later in the training.

QUIZZ!!!

https://kahoot.it/

Kubernetes Workshop Series

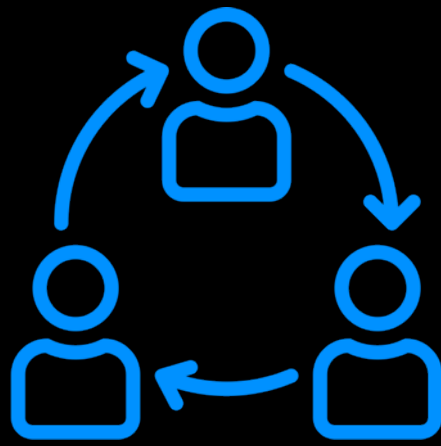
Prepare the Labs

00

Session Objectives

Attendees will run their own ***Personal Training Environment (PTE)*** in the VM.

Following some lectures will be ***hands-on*** work that each participant can to complete.



JTC90 Lab Setup

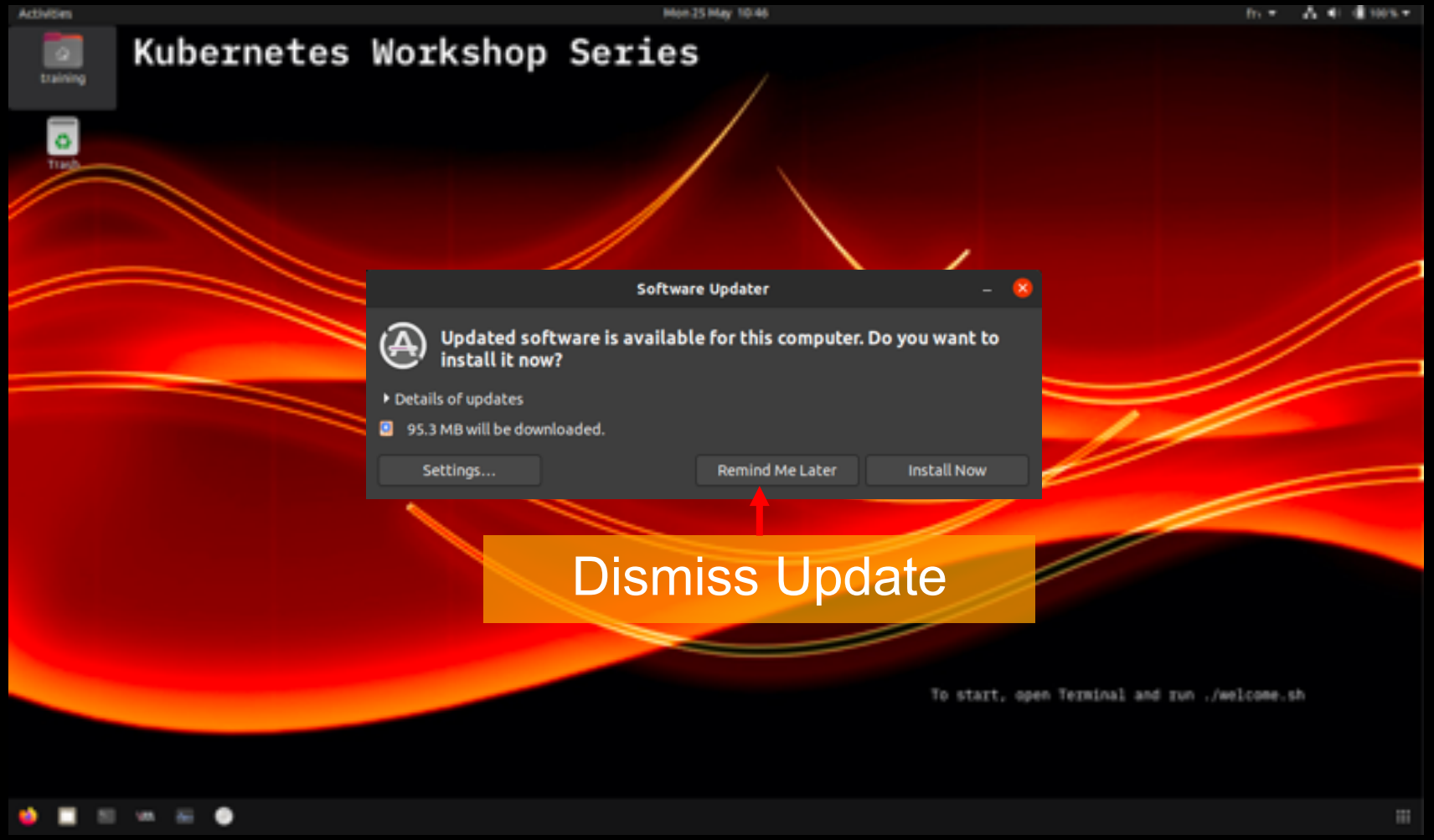
Task 1: Download Training VM

Task 2: Setup VMWare / VirtualBox

Task 3: Start Training VM

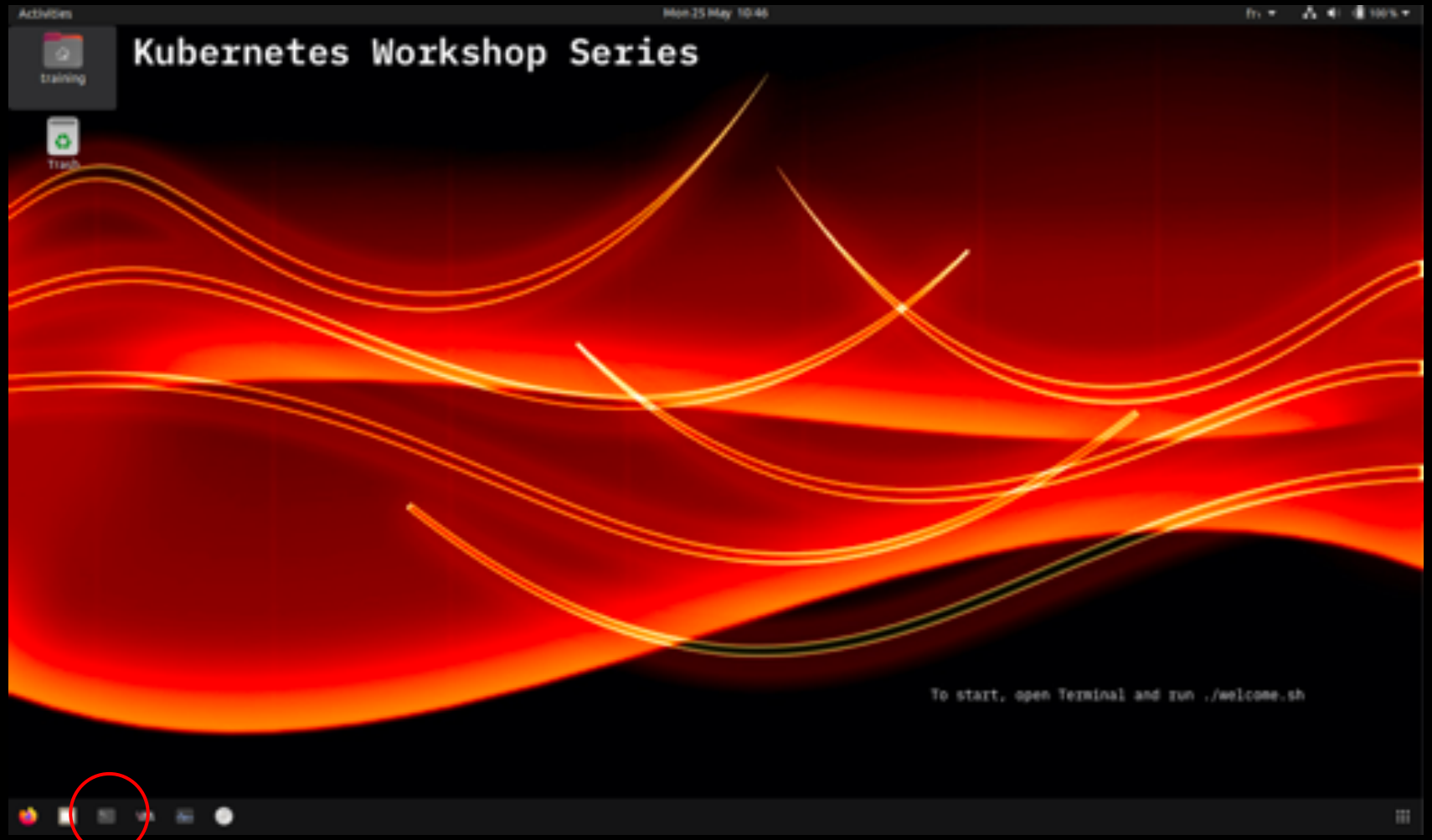
Task 4: Login / Check

Accessing your Personal Training Environment



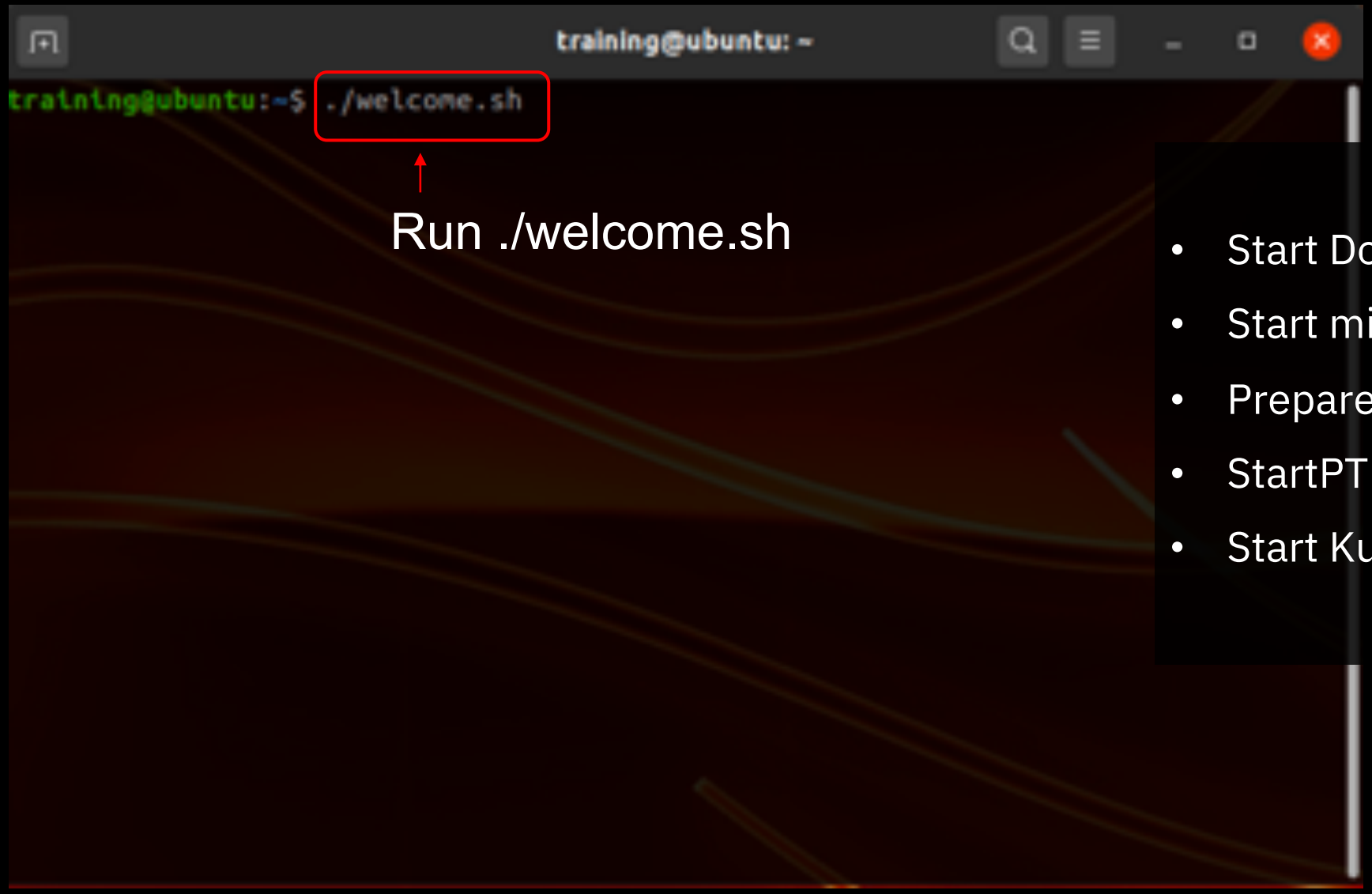
Accessing your Personal Training Environment

Accessing your Personal Training Environment



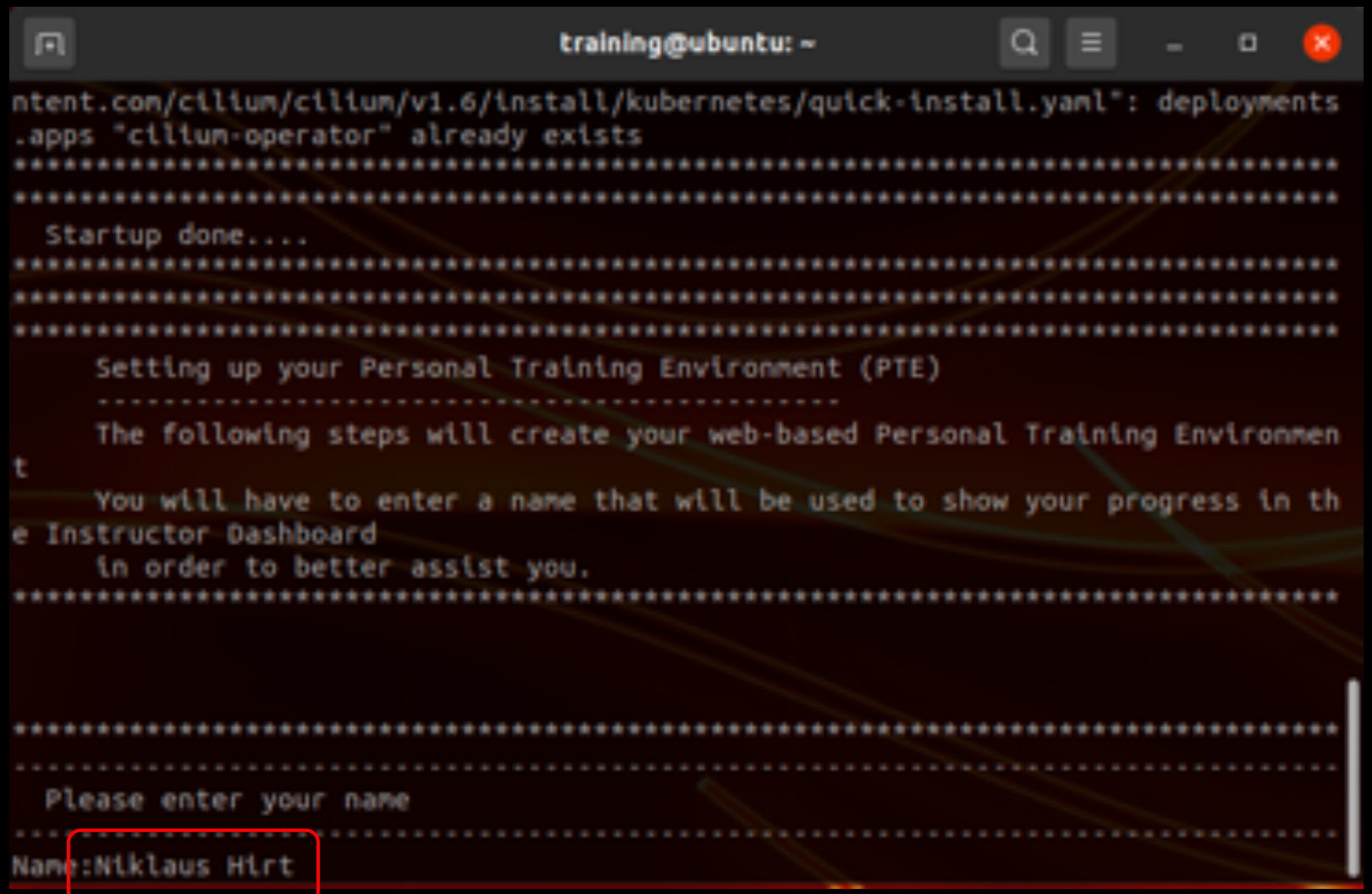
Start Terminal

Accessing your Personal Training Environment



- Start Docker
- Start minikube
- Prepares networking
- StartPTE
- Start Kubernetes Dashboard

Accessing your Personal Training Environment



Enter your name

Name will be used to show your progress in the Instructor Dashboard in order to better assist you

Accessing your Personal Training Environment

Troubleshooting

- If the startup script doesn't work you can run ./resetEnvironment.sh
  (this can take up to 30 minutes as it has to redownload all Docker images)

- If you lose your PTE Webpage just run `minikube service student-ui`

- Windows 10 problems can mostly be fixed by turning off Hyper-V by running (as admin)
  `bcdedit /set hypervisorlaunchtype off`
  and rebooting.
  This disables Hyper-V and allows Virtualbox to support nested virtualisation.

- You can turn it back on again with
  `bcdedit /set hypervisorlaunchtype auto`

Accessing your Personal Training Environment

Troubleshooting

I have added a standalone version to the Git repository for participants wishing to run the Labs directly on their PC.

This is **untested** and I cannot guarantee that all the Labs will be working 100%.

You must have the following setup on your PC:

- Minikube
- Docker
- Git

1. Clone the repository to your home directory
   ```
   git clone https://github.com/niklaushirt/training.git
   ```
2. Go to the installation directory
   ```
   cd ~/training/standalone
   ```
3. Run the preparation script
   ```
   ./welcome.sh
   ```

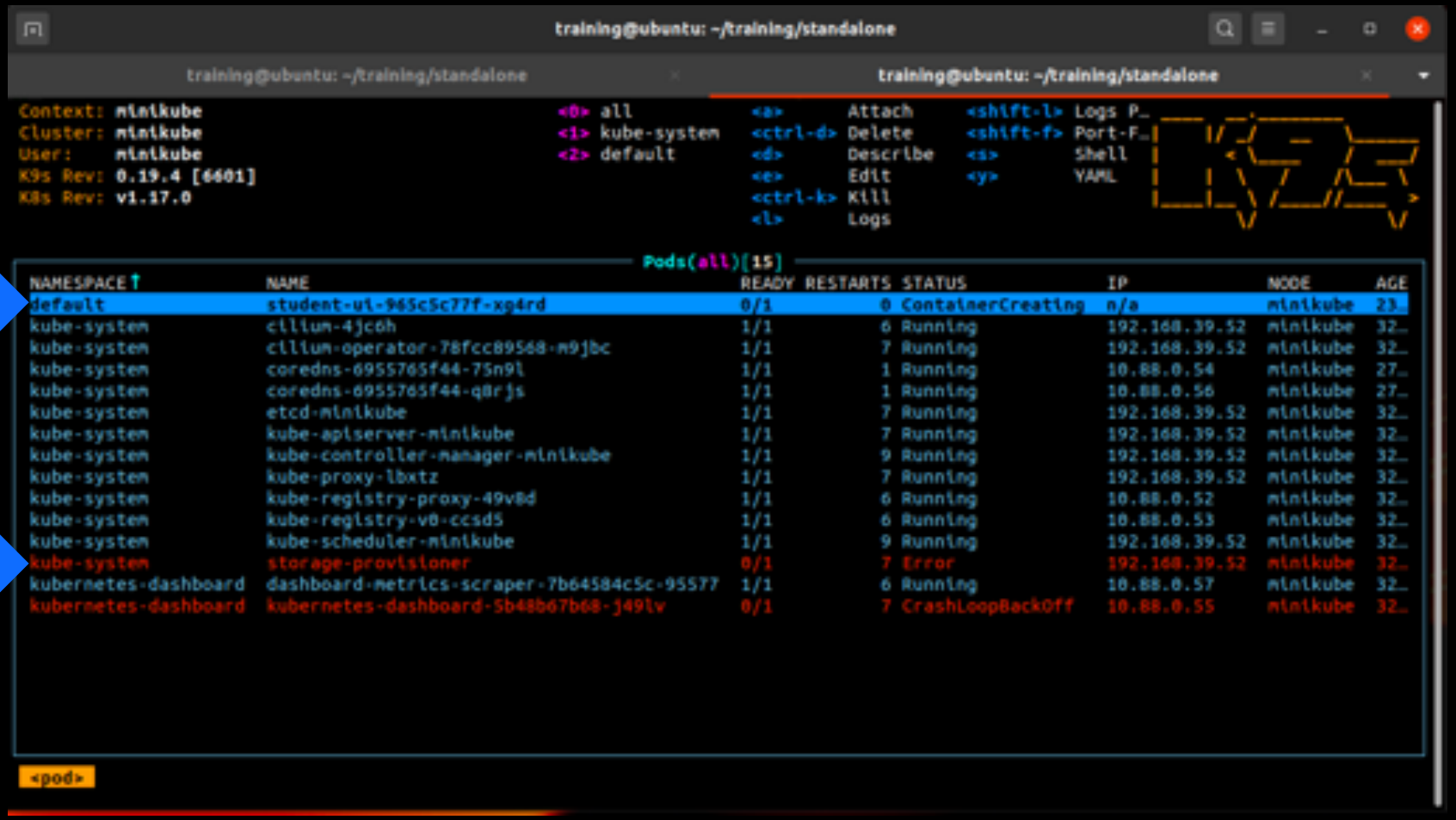
Accessing your Personal Training Environment

Troubleshooting

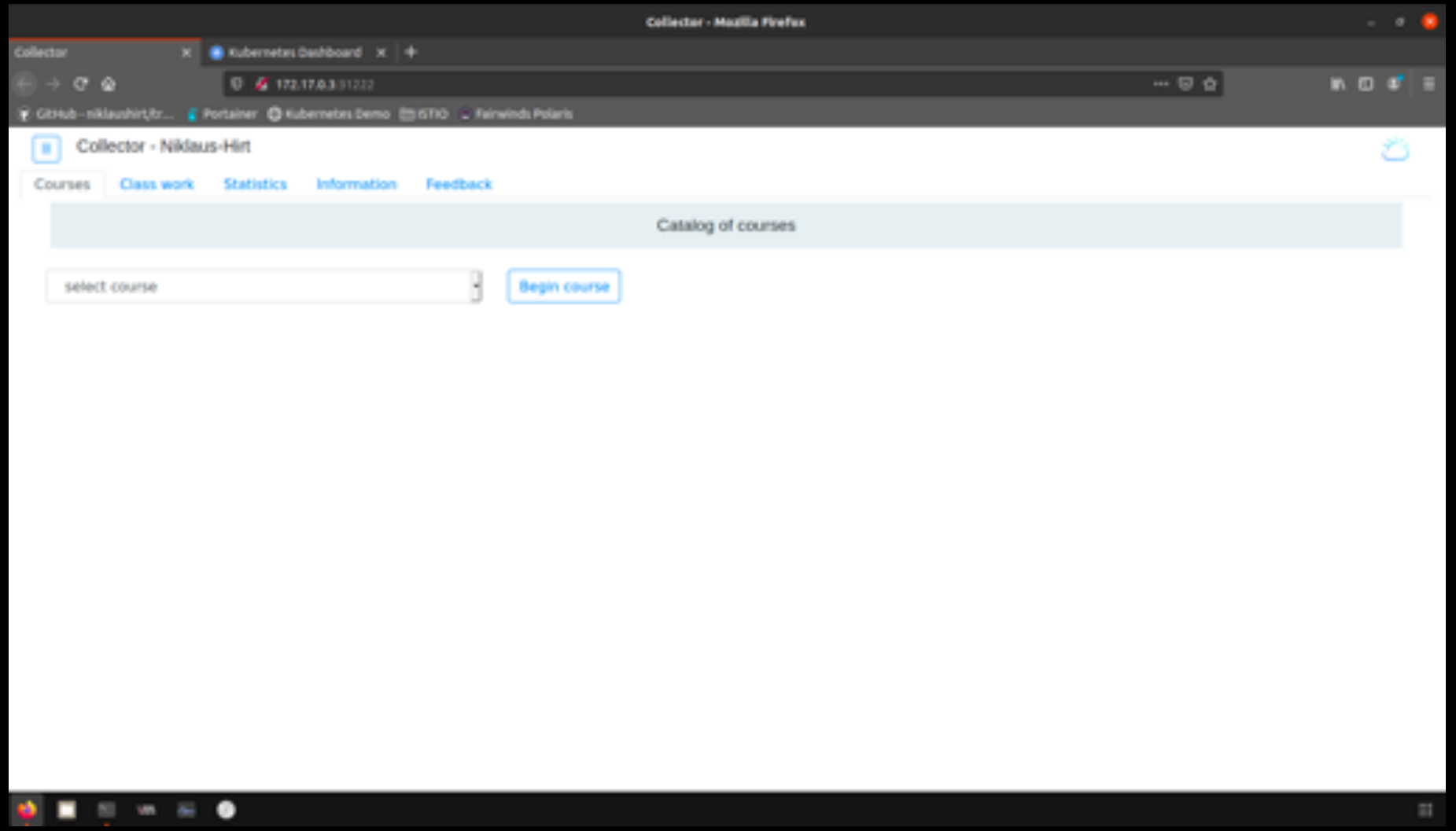
- Run ***k9s*** in the Terminal – wait for all the pods to be Running (blue – 1/1)

Image pulling - wait

Dependencies - wait



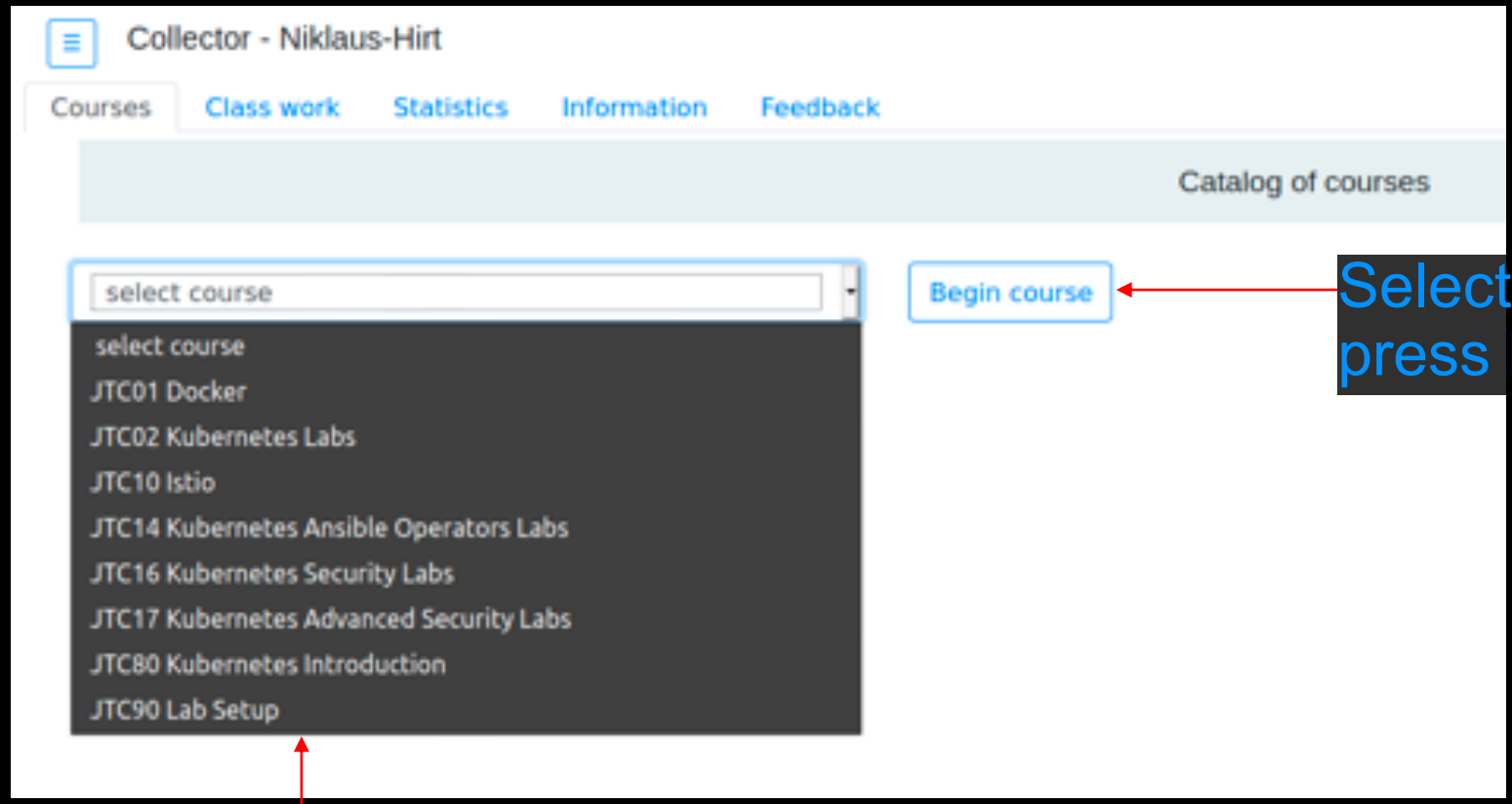
Accessing your Personal Training Environment



When completed, your PTE and Kubernetes Dashboard will open automatically

Accessing your Personal Training Environment

Name will be shown

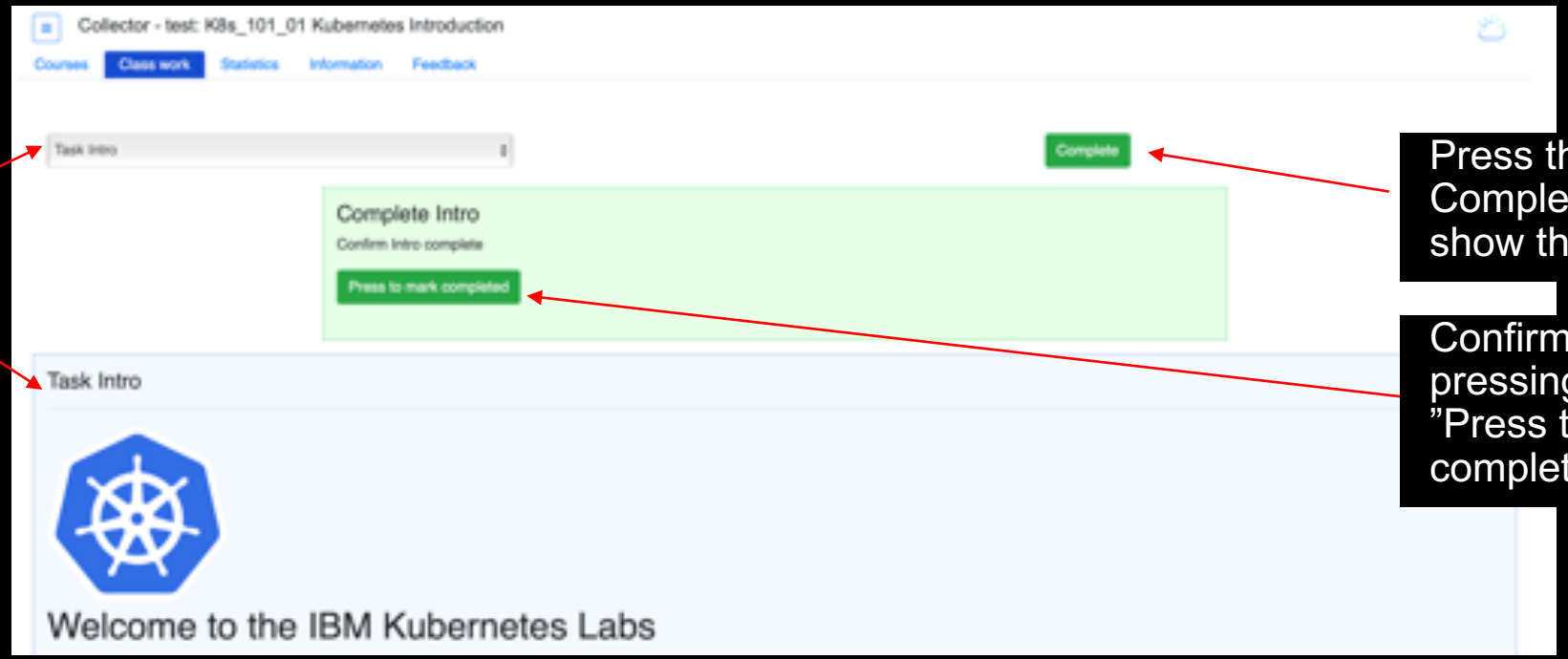


Select course and press button to begin

Current course catalog

Class Work

Select class work and the blue portion of the screen is shown



Press the green Complete button to show the green portion.

Confirm completion by pressing the green ”Press to mark completed” button.

! The Complete Button might not show instantly depending on the course settings

Following your progress

Course title

The number of items tracked will change based on the current course selected.

Green checkmark - item is completed
Red circle - item is waiting to be completed

Instructor Dashboard

Remaining Time for the Lab

What happened so far...
Everybody Loves Containers

A **standard way to package an application and all its dependencies** so that it can be moved between environments and run without changes

Containers work by **isolating the differences between applications** inside the container so that everything outside the container can be standardized

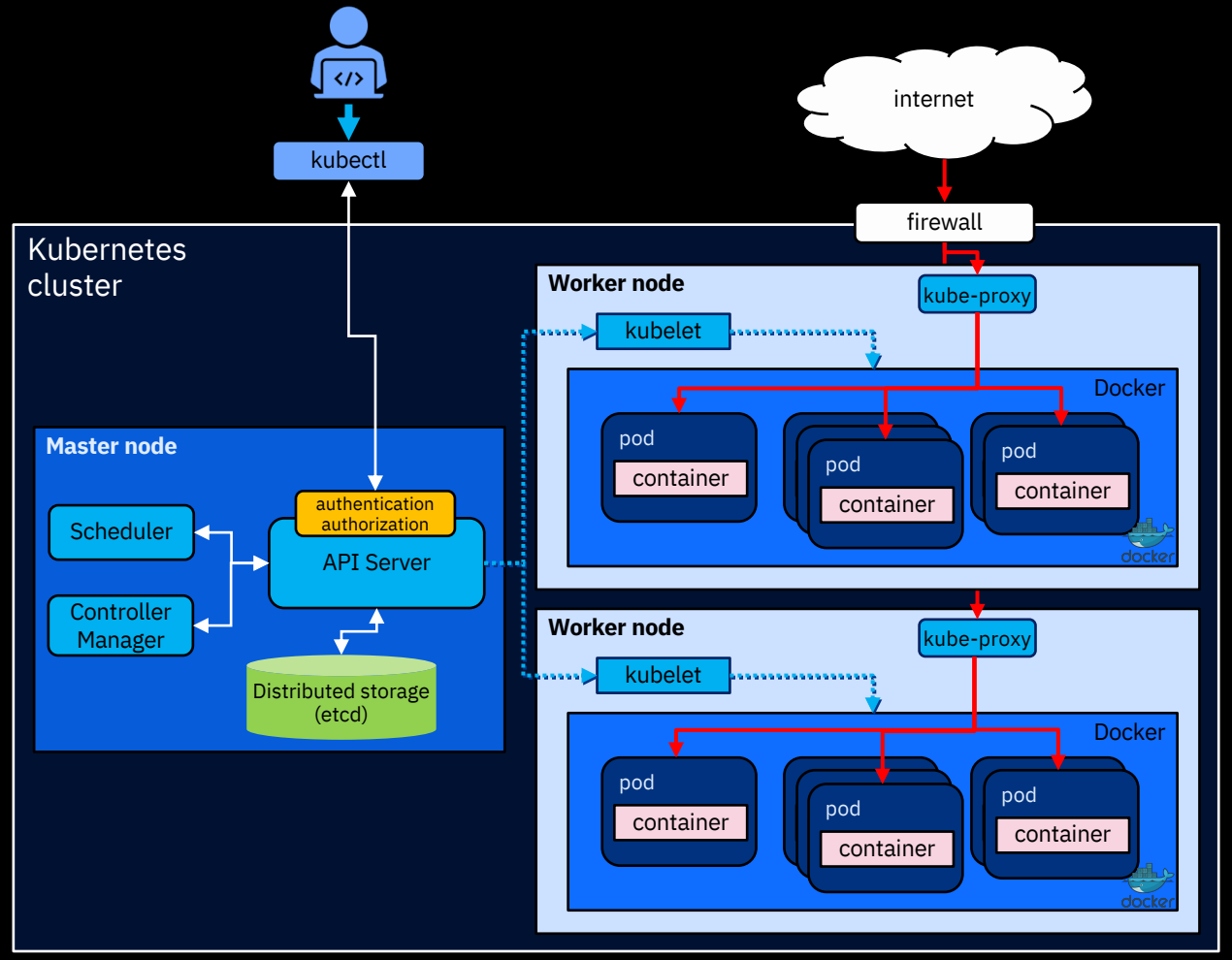
What happened so far...
Kubernetes Cluster Architecture

Master node

- Node that manages the cluster
- Scheduling, replication & control
- Multiple nodes for HA

Worker nodes

- Node where pods are run
- Docker engine
- kubelet agent accepts & executes commands from the master to manage pods
- kube-proxy – routes inbound or ingress traffic



What happened so far...
Some hints

- Cluster Sizing
  - https://itnext.io/architecting-kubernetes-clusters-choosing-a-cluster-size-92f6feaa2908#f738

Few large clusters:

- 🟢 Ease of management
- 🟢 Cost-efficient
- 🟢 Optimal resource utilisation
- 🔴 Single point of failure
- 🔴 Lack of isolation
- 🔴 Management of non-namespaced resources
- 🔴 Cluster size is limited

Many small clusters:

- 🟢 Strong isolation
- 🟢 Reduced blast radius
- 🟢 Granular access control
- 🟢 Clusters can be customised
- 🔴 Costly
- 🔴 Inefficient resource usage
- 🔴 Operational overhead

What happened so far…
and some corrections…

- K8s-demo App
  - Limits were too low for certain configs

- If you have already done welcome.sh
  - Please run:

```
cd ~/training
gitrefresh
cd
```

```
kind: Deployment
metadata:
  name: access-frontend-backend
  namespace: k8sdemo
spec:
  …
    spec:
      containers:
      - name: k8sdemo
          image: niklaushirt/k8sdemo:1.0.0
          …
          resources:
            requests:
              cpu: "150m"
              memory: "150Mi"
            limits:
              cpu: "500m"
              memory: "500Mi"
```

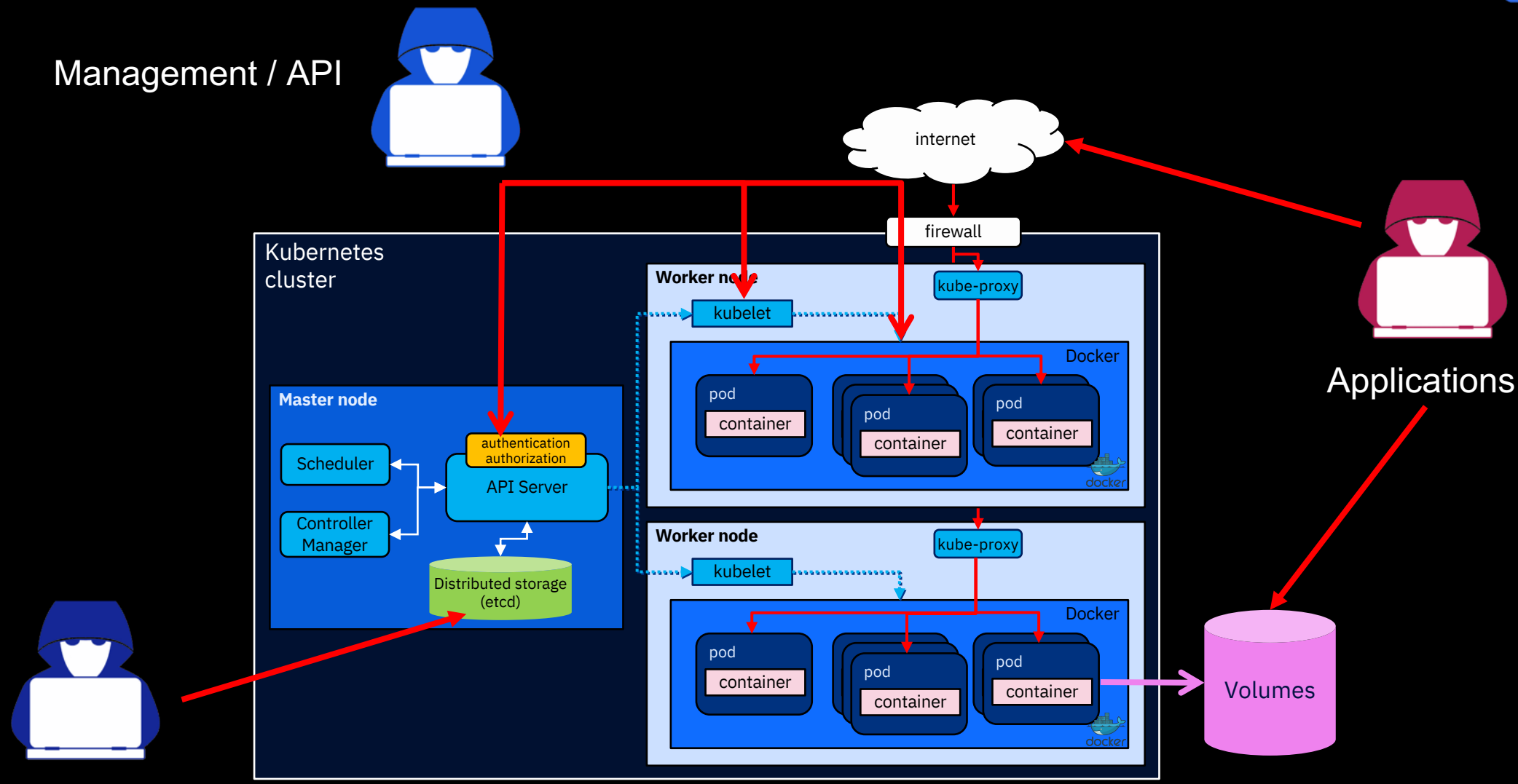
QUESTIONS?

Kubernetes Workshop Series

Security Introduction

01

Kubernetes – **Security – Attack surface**



Kubernetes – **Security Basic Topics**

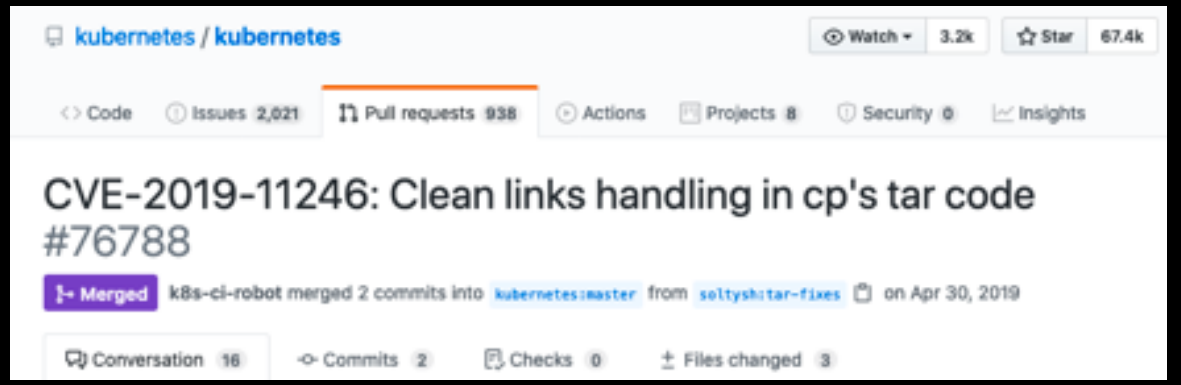
The **Billion Laughs** attack is a particular type of denial of service (DoS) attack which is aimed specifically at XML document parsers. This attack is also referred to as an XML bomb or an exponential entity expansion attack.

It exploits the fact that nested references to nodes can grow very large when expanded. Because the **kube-apiserver** doesn't perform validation on the manifest, it doesn't detect if those nested references will cause a problem. If the nesting references grow too large, excessive CPU and RAM usage can render the apiserver unresponsive to connections … hence the Denial of Service.

Kubernetes – **Security Basic Topics**

CVE-2019-11246: Another kubectl Path Traversal Vulnerability Disclosed

CVE registration 17.04.2019
Git Merge on 30.04.2019



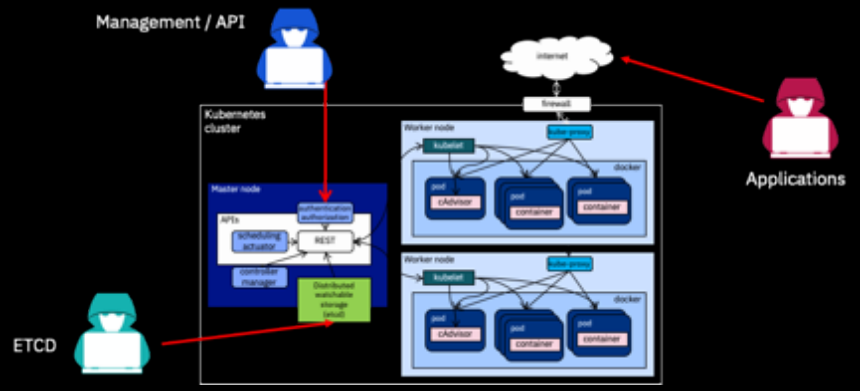
Kinsing Malware Attacks Targeting Container Environments

CVE-2019-5021: Alpine Docker Image ‘null root password’ Vulnerability

Kubernetes Pod Escape Using Log Mounts

Source: https://blog.aquasec.com/topic/security-threats

Kubernetes – **Security Basic Topics**



Reduce Kubernetes Attack Surfaces

etcd

Management/API

Applications

- Secure access to etcd
- Controlling access to the Kubernetes API
- Controlling access to the Kubelet
- Enforce resource usage limits for workloads
- Rotate infrastructure credentials frequently
- Enable audit logging
- Use Linux security features
- Controlling what privileges containers run with
- Enforcing Network Policies
- Image Scanning of your containers
- Volume Security

Source: https://kubernetes.io/docs/tasks/administer-cluster/securing-a-cluster
https://kubernetes.io/blog/2018/07/18/11-ways-not-to-get-hacked/

QUESTIONS?

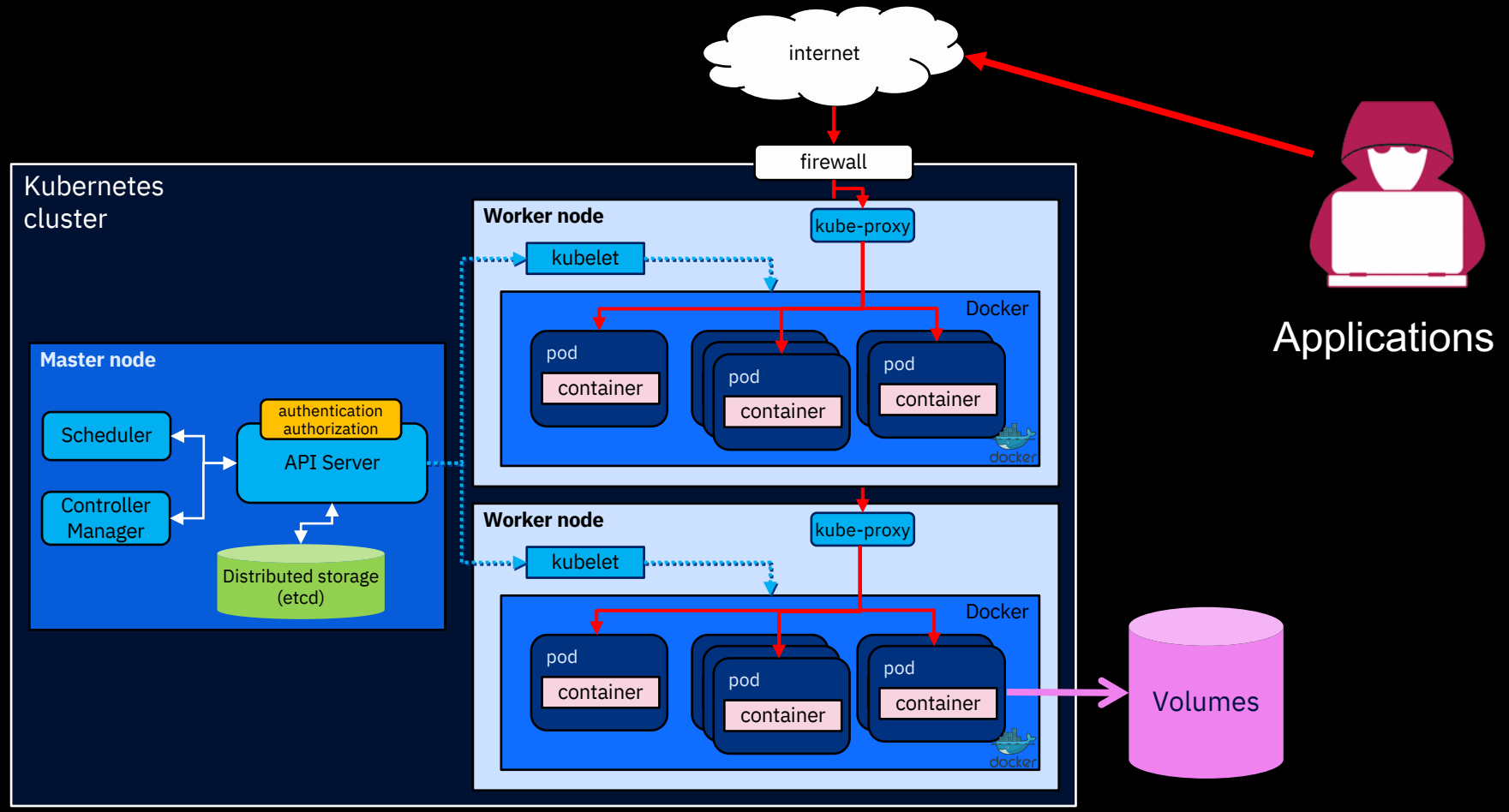
Kubernetes Workshop Series

Security Elements

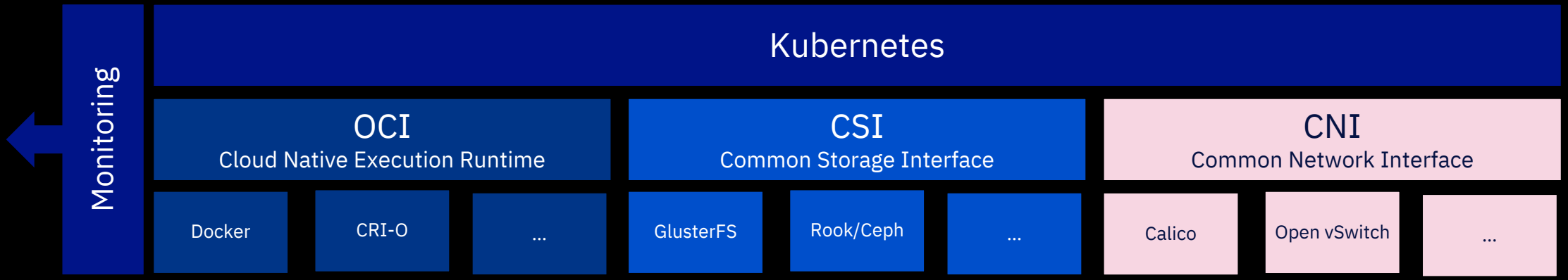
02

Securing Workloads

Kubernetes Security – Mitigation – Network



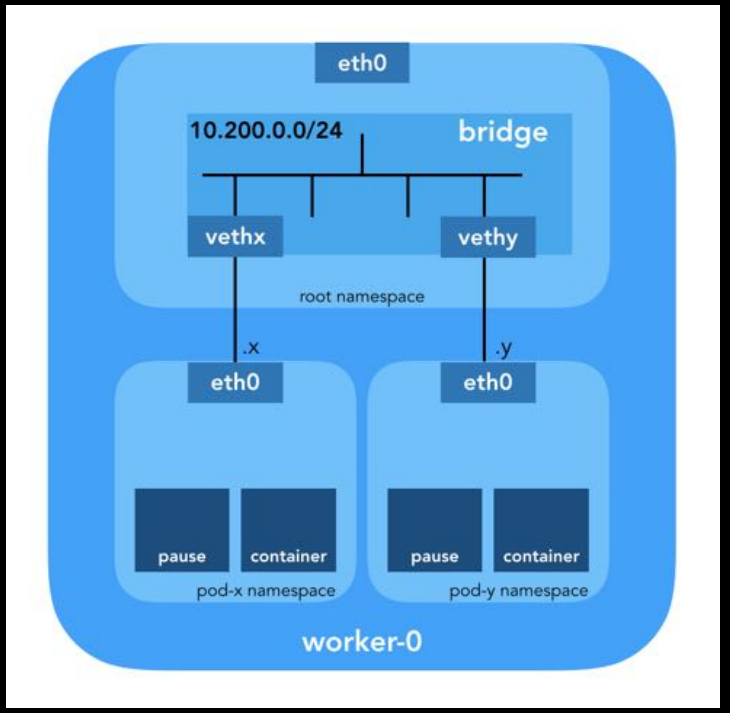
Kubernetes Security – Mitigation – Network



CNI – Common Network Interface

Provides a rich set of security enforcement capabilities running on top of a highly scalable and efficient virtual network

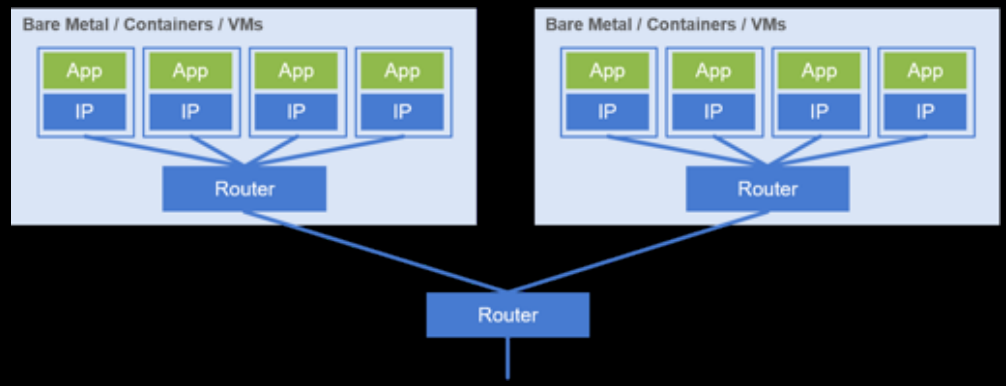
- **Calico**
  Virtual networking and network security for containers, VMs, and bare metal services.
- **Open vSwitch**
  Production quality, multilayer virtual switch
- ...



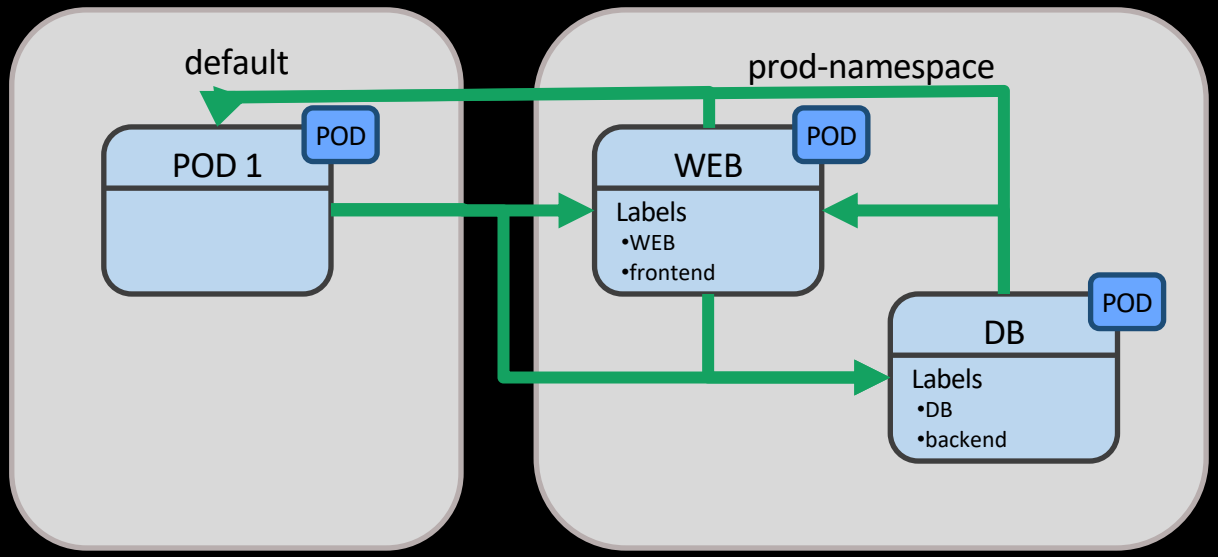
Kubernetes Security – Mitigation – Network

Operates at **Layer 3**, which is the network layer

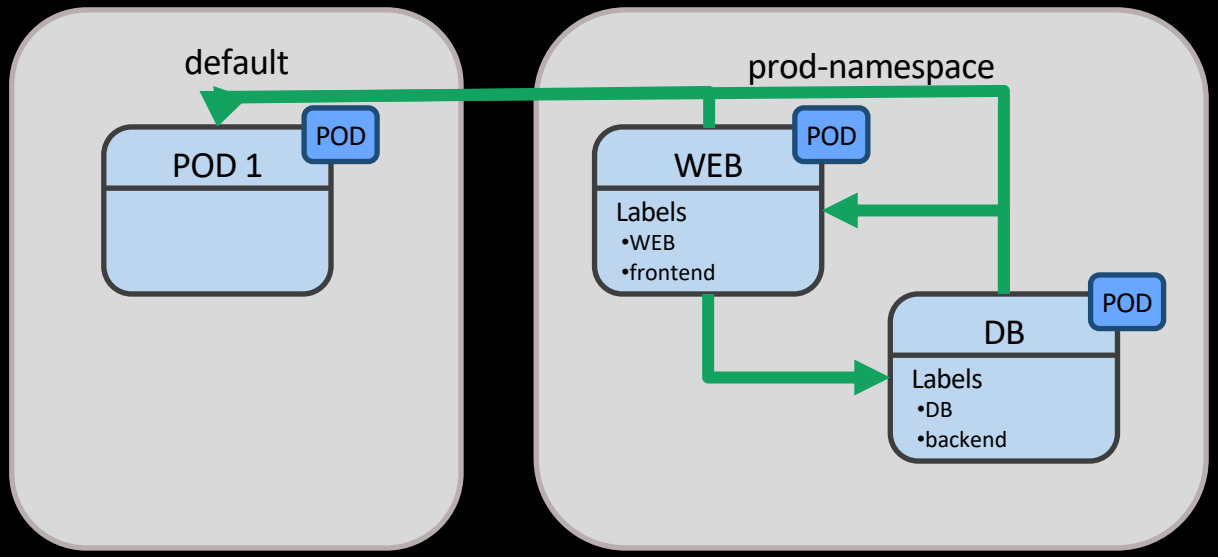
- Has the advantage of being **universal** (DNS, SQL, real-time streaming, …)
- Can extend beyond the service mesh (including to **bare metal or VM** endpoints not under the control of Kubernetes).
- Calico's policy is enforced at the host node, outside the network namespace of the guest pods.
- Based on **iptables**, which are packet filters implemented in the standard Linux kernel, it is extremely fast.



Kubernetes Security – Mitigation – Network

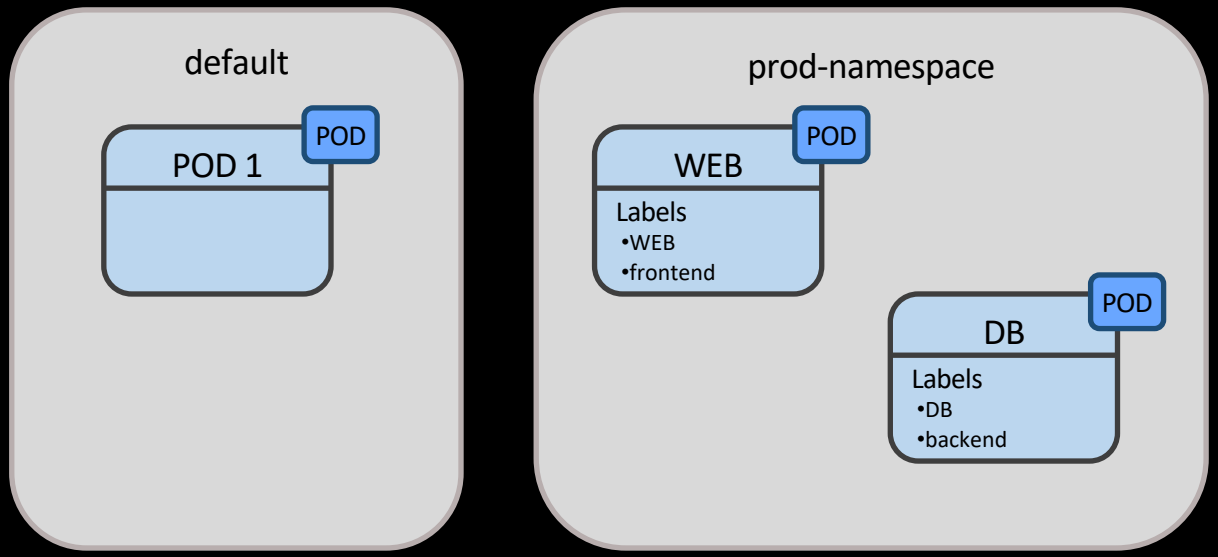


Kubernetes Security – Mitigation – Network



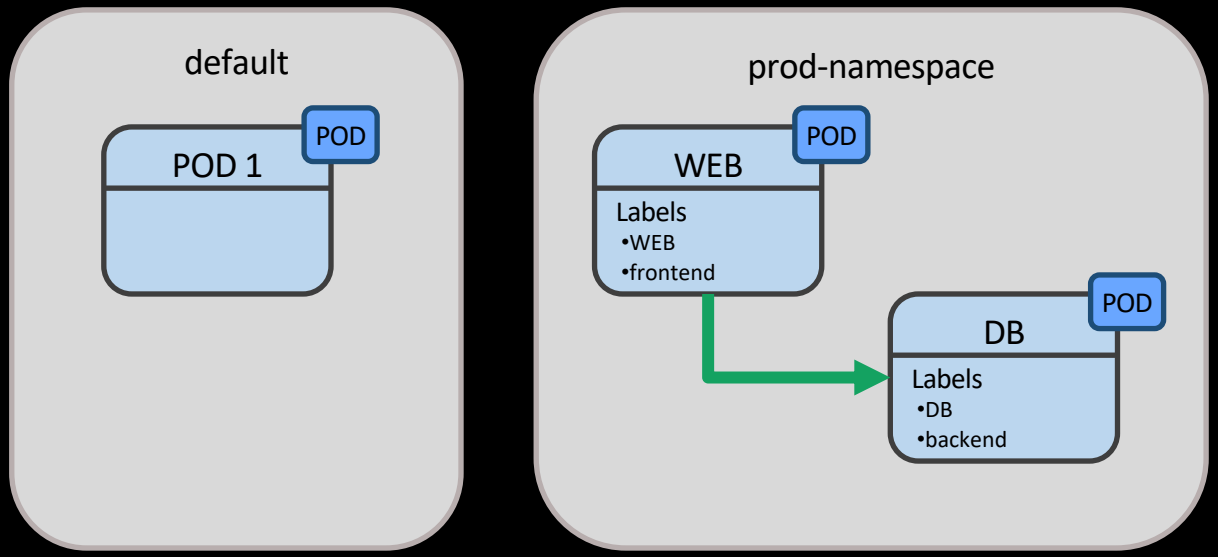

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny
  namespace: demo-namespace
spec:
  podSelector:
    matchLabels: {}
```

Kubernetes Security – Mitigation – Network



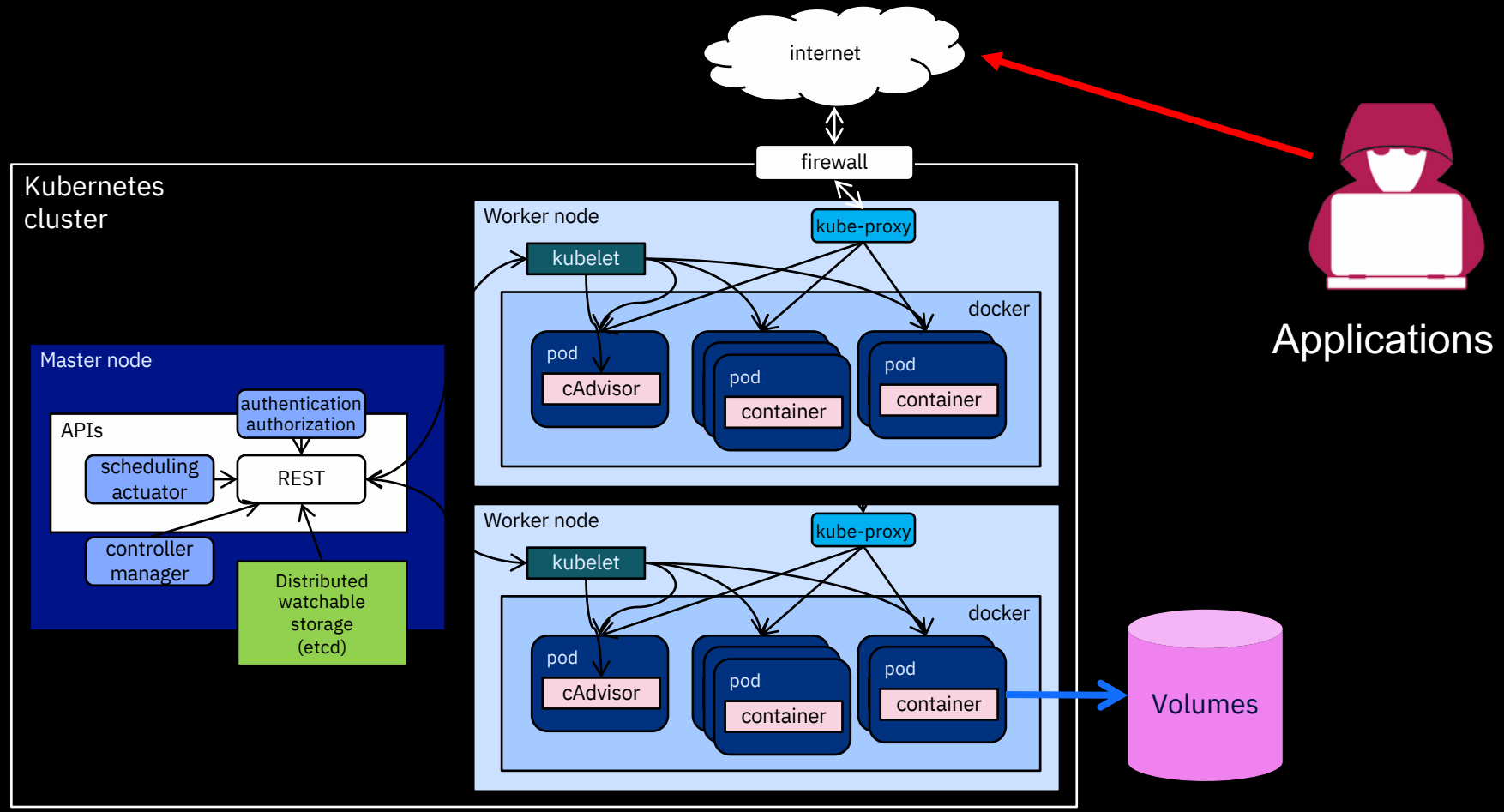

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny-all
spec:
  podSelector:
    matchLabels: {}
```

Kubernetes Security – Mitigation – Network




```
kind: NetworkPolicy
metadata:
  name: access-frontend-backend
  namespace: prod-namespace
spec:
  podSelector:
    matchLabels:
      run: DB
  ingress:
    - from:
      - podSelector:
          matchLabels:
            run: WEB
```

Kubernetes Security – Containers

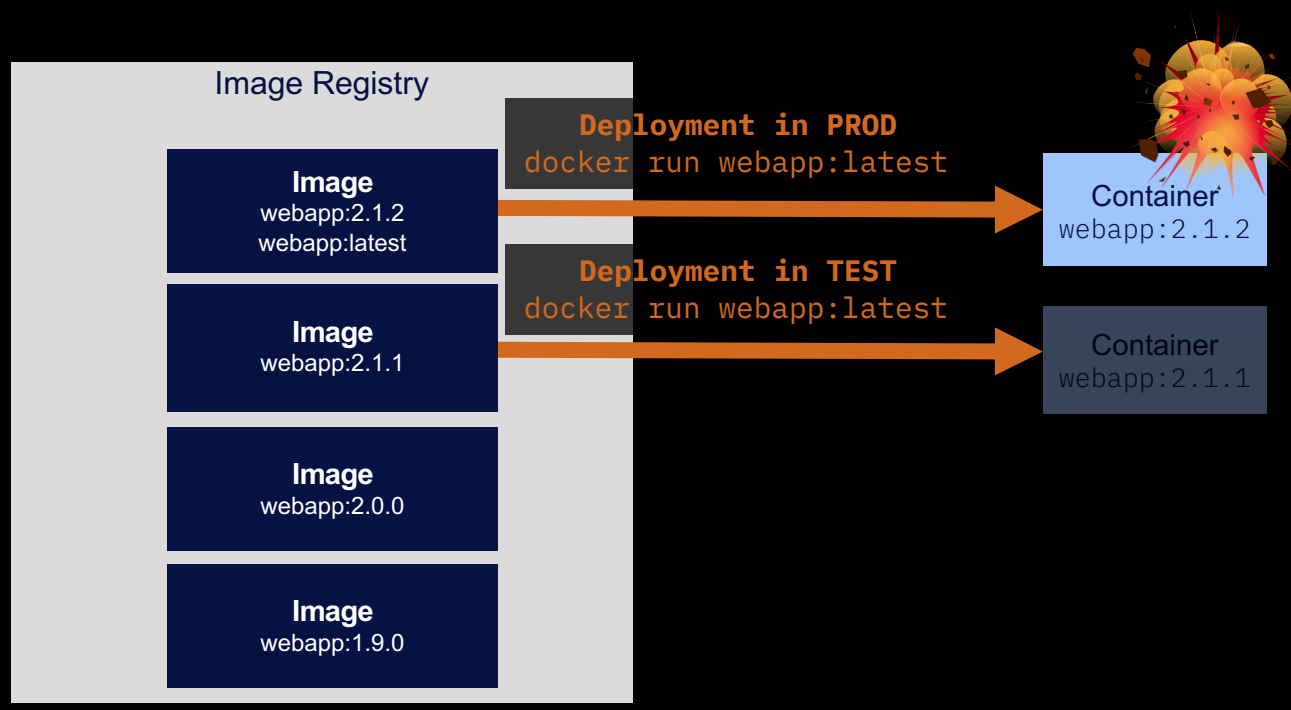


Kubernetes Security – Rolling Tags

Avoid Rolling Tags at Any Cost

Avoid commands like *docker pull myregistry/myimage:latest.*
This “latest” is an example of a rolling tag (i.e. a tag that will point to different images over time).

If you want your deployments to be secure and maintainable, make sure that you use immutable images (for example: “*myregistry/myimage:1.1.2 "*).
With this approach, every time you deploy or scale, you know what image you are using and you will have the guarantee that the deployed image has been tested and validated.



Kubernetes Security – Multiple processes per container

Don’t run multiple processes per container

Docker general best practices suggest a single process per container simply for usability and size reasons.

But it also keeps your attack surface small and limits the number of potential vulnerabilities.

Source https://medium.com/uptime-99/kubernetes-203-security-basics-6af5aee0367d

Kubernetes Security – Root User

Don’t run containers as the root user

By default, processes run as the root user inside the container. This is easy to avoid using the pod specification to set a high-numbered UID.

```
spec:
  securityContext:
    runAsUser: 10324
```

Source https://medium.com/uptime-99/kubernetes-203-security-basics-6af5aee0367d

Kubernetes Security – Privileged Containers

Don’t run containers as privileged

Running a container or pod as privileged gives it the ability to make modifications to the host.
This is a large security issue that is easy to mitigate by just not doing it.

Source https://medium.com/uptime-99/kubernetes-203-security-basics-6af5aee0367d

Kubernetes Security – Linux Capabilities

Don’t use the default list of capabilities

Docker runs containers with a significant set of Linux capabilities by default, many of which your app might not require.

The following config will drop all Linux capabilities by default, allowing you to add only the specific capabilities your app actually needs:

```
spec:
  containers:
  - name: foo
    securityContext:
      capabilities:
        drop:
        - ALL
```

Source https://medium.com/uptime-99/kubernetes-203-security-basics-6af5aee0367d

Kubernetes Security – Mitigation - Containers

Open Policy Agent

Scan the manifests for vulnerabilities

conftest

Conftest is a utility to help you write tests against structured configuration data.
For example you could write tests for your Kubernetes configurations, Dockerfiles, or Tekton pipeline definitions, Terraform code, Serverless configs or any other structured data.

```
training@ubuntu:~$ conftest test -p ~/training/conftest/src/examples/kubernetes/policy ~/training/deployment/demoapp.yaml
FAIL - /home/training/training/deployment/demoapp.yaml - Containers must not run as root in Deployment k8sdemo
FAIL - /home/training/training/deployment/demoapp.yaml - Deployment k8sdemo must provide app/release labels for pod selectors
FAIL - /home/training/training/deployment/demoapp.yaml - k8sdemo must include Kubernetes recommended labels: https://kubernetes.io/docs/cond
FAIL - /home/training/training/deployment/demoapp.yaml - Found deployment k8sdemo but deployments are not allowed
--------------------------------------------------------------------------------
PASS: 1/5
WARN: 0/5
FAIL: 4/5
```

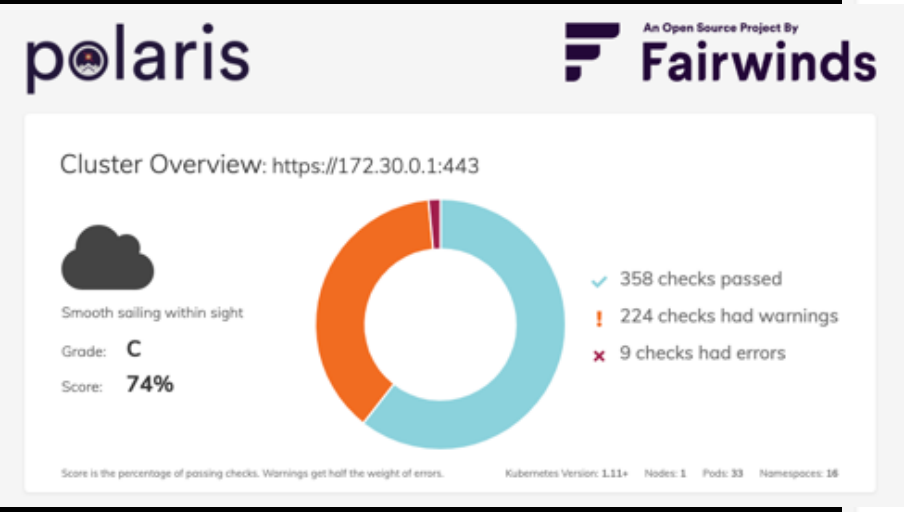
```
training@ubuntu:~$ conftest test -p ~/training/conftest/src/examples/docker/policy --output=json ~/training/conftest/src/examples/docker/Dockerfile
[
        {
                "filename": "/home/training/training/conftest/src/examples/docker/Dockerfile",
                "warnings": [],
                "failures": [
                        {
                                "msg": "unallowed image found [\"openjdk:8-jdk-alpine\"]"
                        }
                ],
                "successes": []
        }
]
```

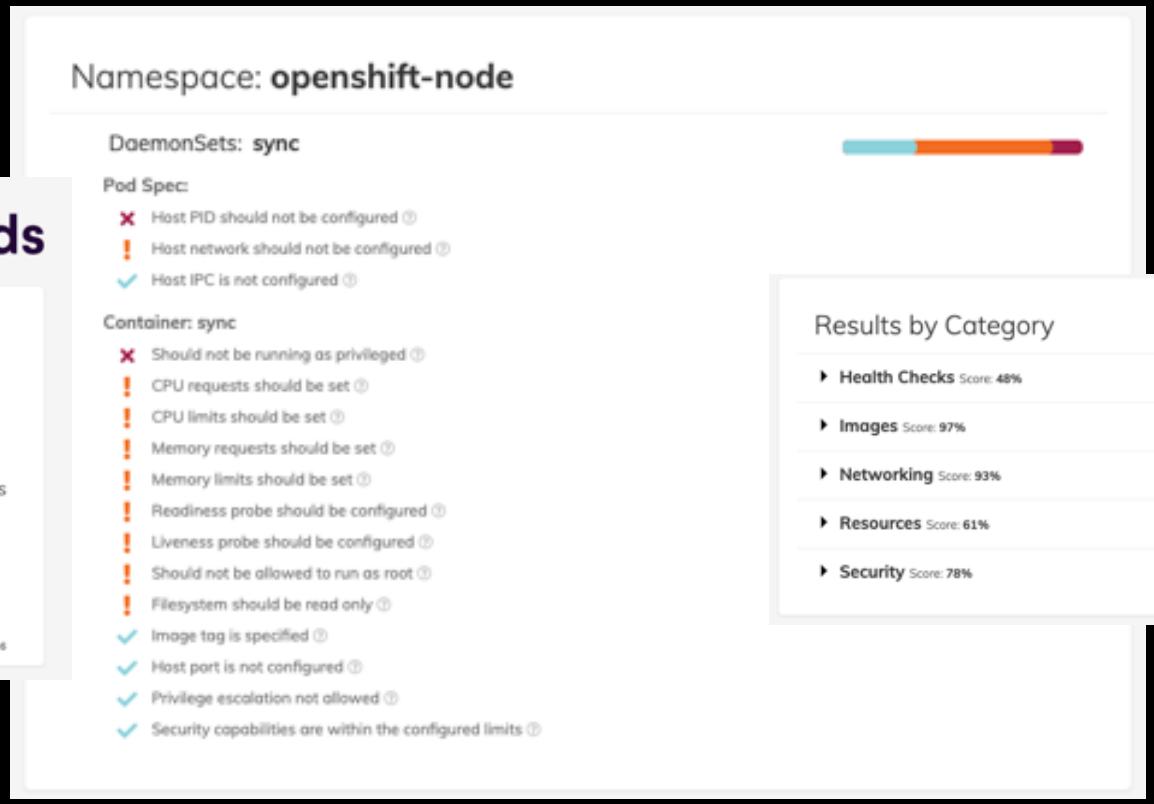
Source: https://github.com/open-policy-agent/conftest

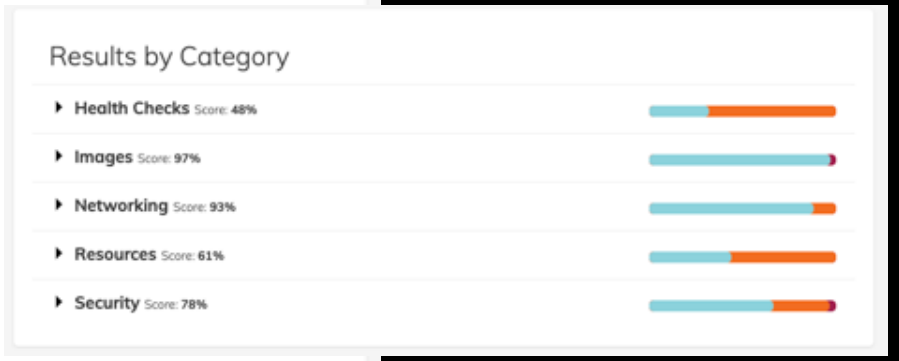
Kubernetes Security – Mitigation – Containers

Scan for K8s Best Practices

Polaris – Validation of best practices

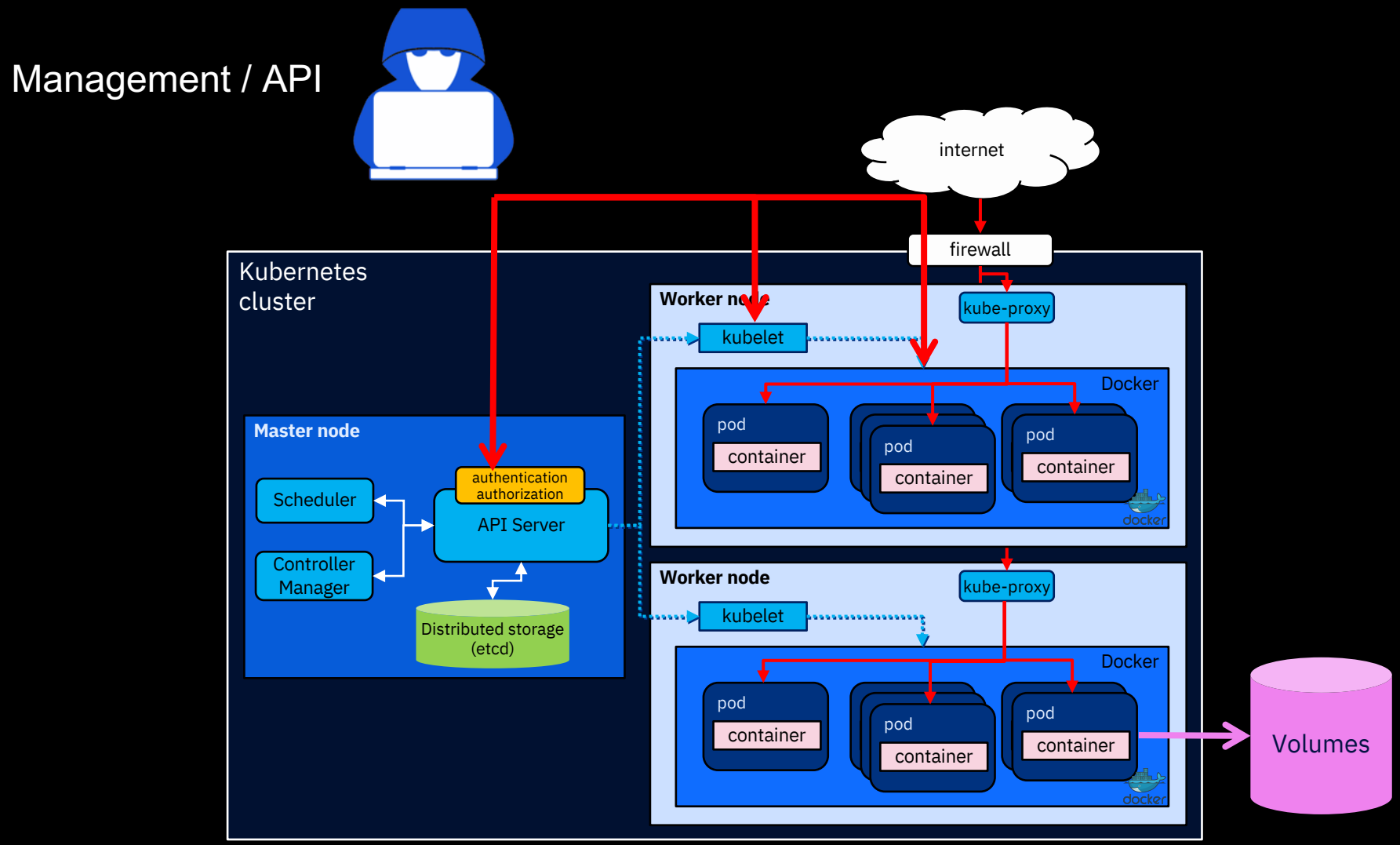






Source: https://github.com/FairwindsOps/polaris

Kubernetes Security – API / Cluster Security



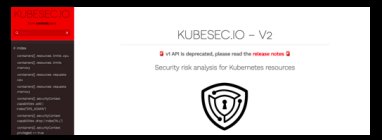
Kubernetes Security – Namespaces

Use Namespaces Liberally

“Namespaces are cheap.”
Use them to separate things like infrastructure tooling and applications.
This allows you to restrict access easily using **RBAC** and to limit the scope of applications. It will also make your **network policy** creation easier when you decide to do it.

Kubernetes Security – Mitigation - Cluster Security

Scan the for K8s Best Practices



kubesec

Performs security risk analysis for Kubernetes resources and tells you what you should change in order to improve the security of those pods.
It gives a score that you can use to create a minimum standard. The score incorporates a great number of Kubernetes best practices.


```
{
  "object": "Deployment/k8sdemo.default",
  "valid": true,
  "message": "Passed with a score of 4 points",
  "score": 4,
  "scoring": {
    "advise": [
      {
        "selector": ".spec .serviceAccountName",
        "reason": "Service accounts restrict Kubernetes API access and should be configured with least privilege",
        "points": 3
      },
      {
        "selector": ".metadata .annotations .\"container.seccomp.security.alpha.kubernetes.io/pod\"",
        "reason": "Seccomp profiles set minimum privilege and secure against unknown threats",
        "points": 1
      },
      {
        "selector": "containers[] .securityContext .runAsUser -gt 10000",
        "reason": "Run as a high-UID user to avoid conflicts with the host's user table",
        "points": 1
      }
    ]
  }
}
```


Source: https://kubesec.io/

Kubernetes Security – Mitigation - Cluster Security

BlackBox Scan of the cluster

kubehunter

Kube-hunter hunts for security weaknesses in Kubernetes clusters. The tool was developed to increase awareness and visibility for security issues in Kubernetes environments.

You should NOT run kube-hunter on a Kubernetes cluster that you don't own!

```
~/kube-hunter/kube-hunter.py  --remote $(minikube ip) --active

> ~ Started
> ~ Discovering Open Kubernetes Services...
> |
> | Etcd:
> |   type: open service
> |   service: Etcd
> |_  location: localhost:2379
> |
> | Kubelet API (readonly):
> |   type: open service
> |   service: Kubelet API (readonly)
> |_  location: localhost:10255
...
```

LOCATION	CATEGORY	VULNERABILITY	DESCRIPTION	EVIDENCE
localhost:10250	Remote Code Execution	Anonymous Authentication	The kubelet is misconfigured, potentially allowing secure access to all requests on the kubelet, without the need to authenticate	

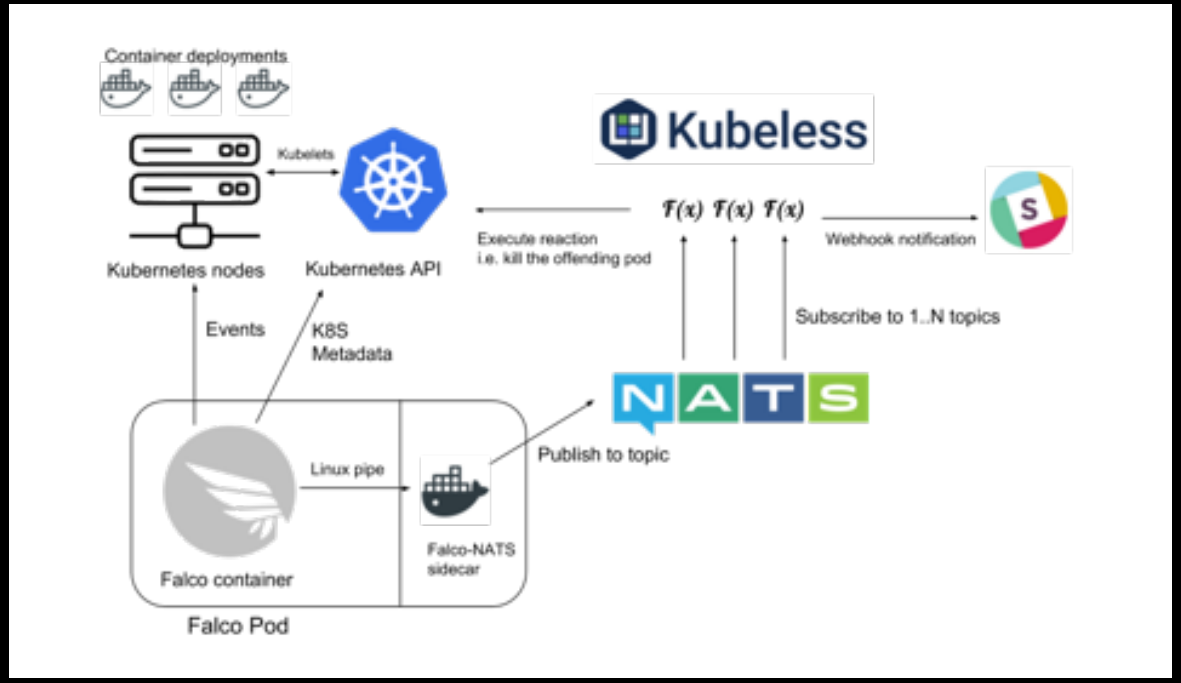
Source: https://github.com/aquasecurity/kube-hunter

Parsing system calls

Falco

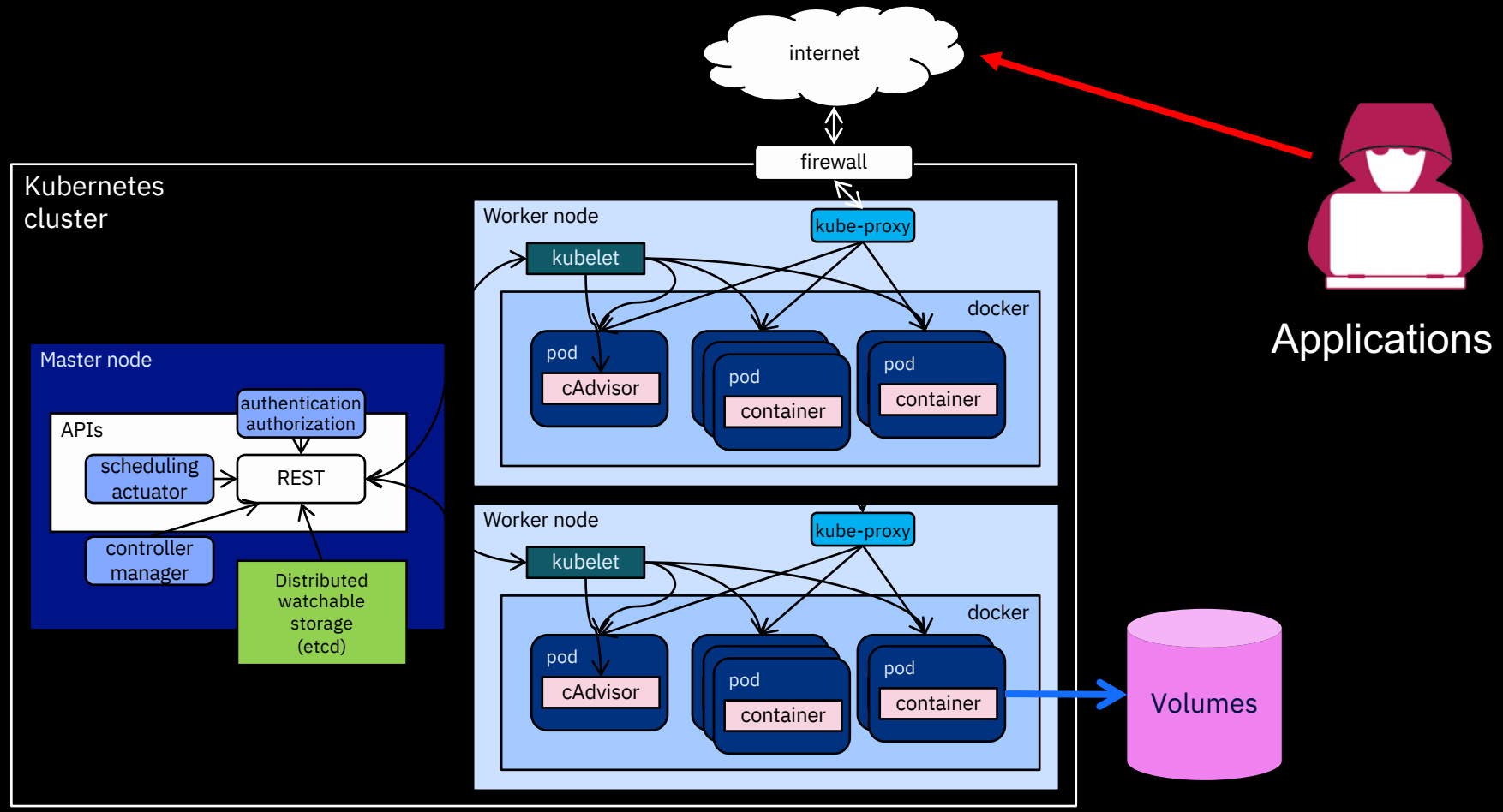
The Falco Project is an open source runtime security tool originally built by Sysdig, Inc. Falco was donated to the CNCF and is now a CNCF incubating project.

Falco parses Linux system calls from the kernel at runtime, and asserts against a rules engine.
If a rule is violated a Falco alert is triggered.



Source: https://falco.org/

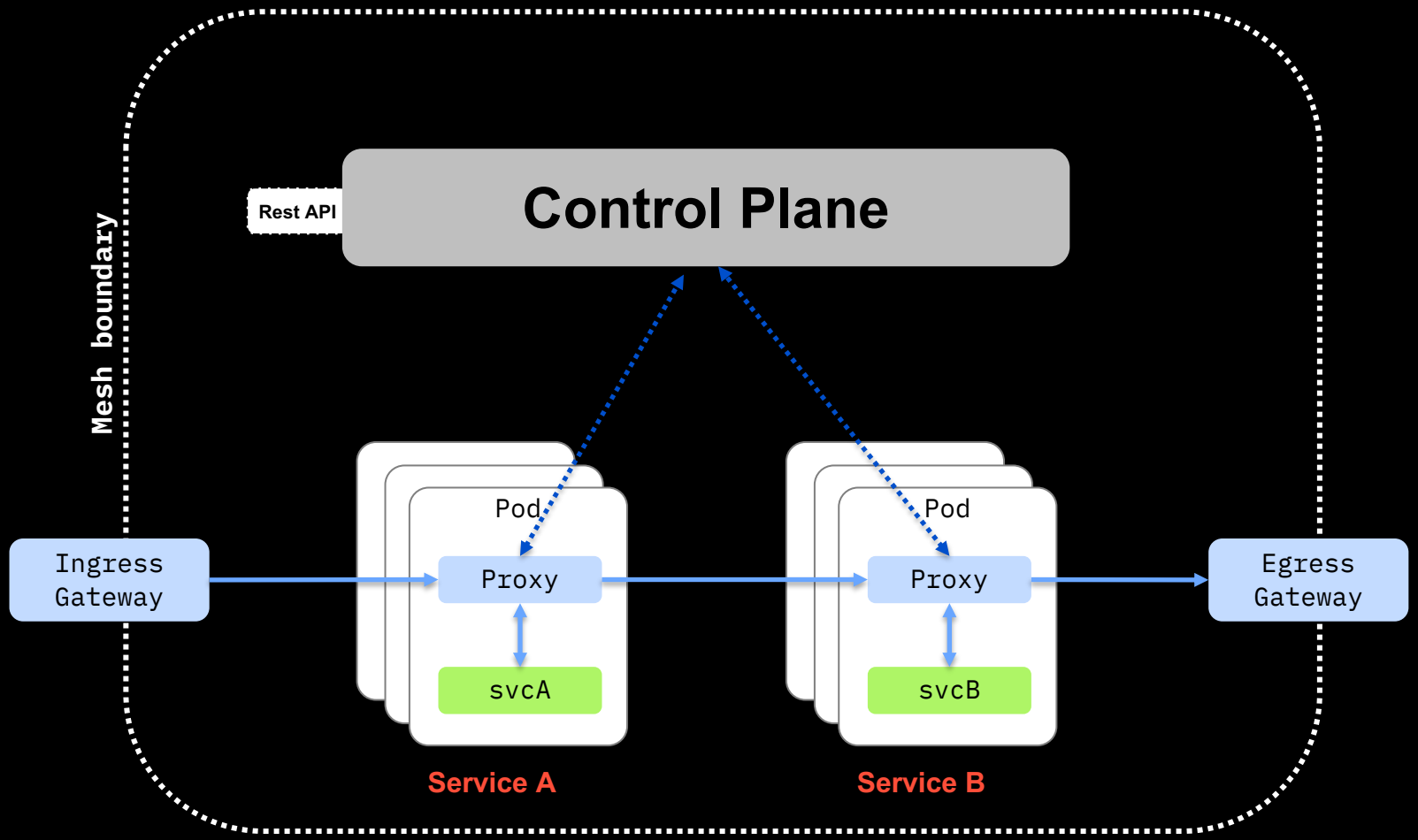
Kubernetes Security – Mesh Network



Service Mesh

Dedicated infrastructure layer
to make
service-to-service communication
fast, safe and reliable

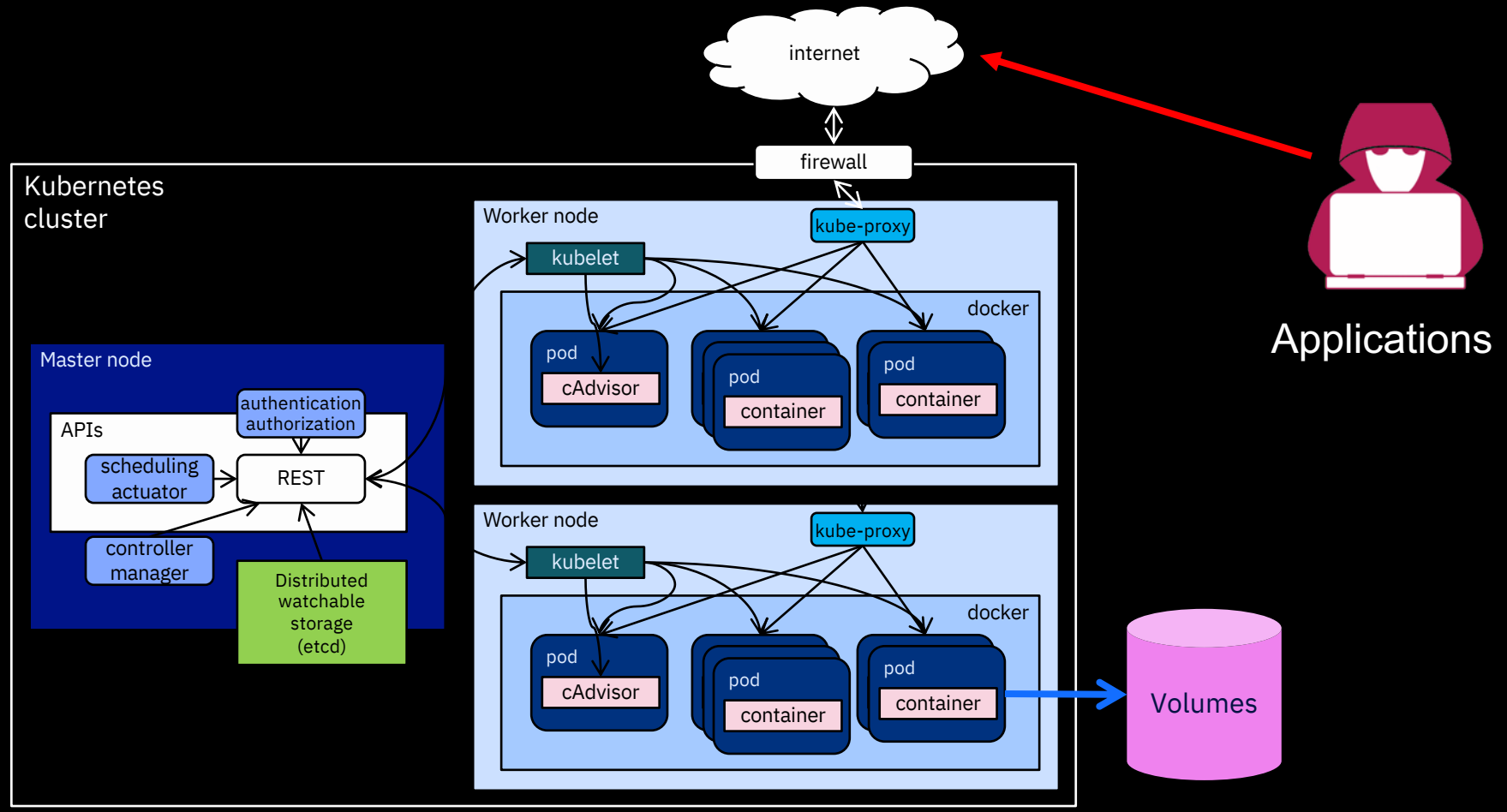
ISTIO - Architecture



Addressing DevOps Challenges

ROLL OUT NEW VERSION WITHOUT DOWNTIME OR CHANGING CODE	**TRAFFIC CONTROL**
HOW TO DO **CANARY TESTING**	**TRAFFIC SPLITTING**
HOW TO DO **A/B TESTING**	**TRAFFIC STEERING**
THINGS DON'T ALWAYS GO CORRECTLY **IN PRODUCTION**...	**TRAFFIC MIRRORING** **RESILIENCY** **RESILIENCY TESTING**
HOW CAN I **LIMIT RATE** FOR SOME OF MY SERVICES?	**RATE LIMITING**
I NEED TO **VIEW AND MONITOR** WHAT IS GOING ON WHEN CRISIS ARISES	**TELEMETRY**
HOW CAN I **SECURE MY SERVICES**?	**AUTHENTICATION** **AUTHORIZATION**

Kubernetes Security – Network Encryption



Kubernetes Security – Mitigation – Network Encryption

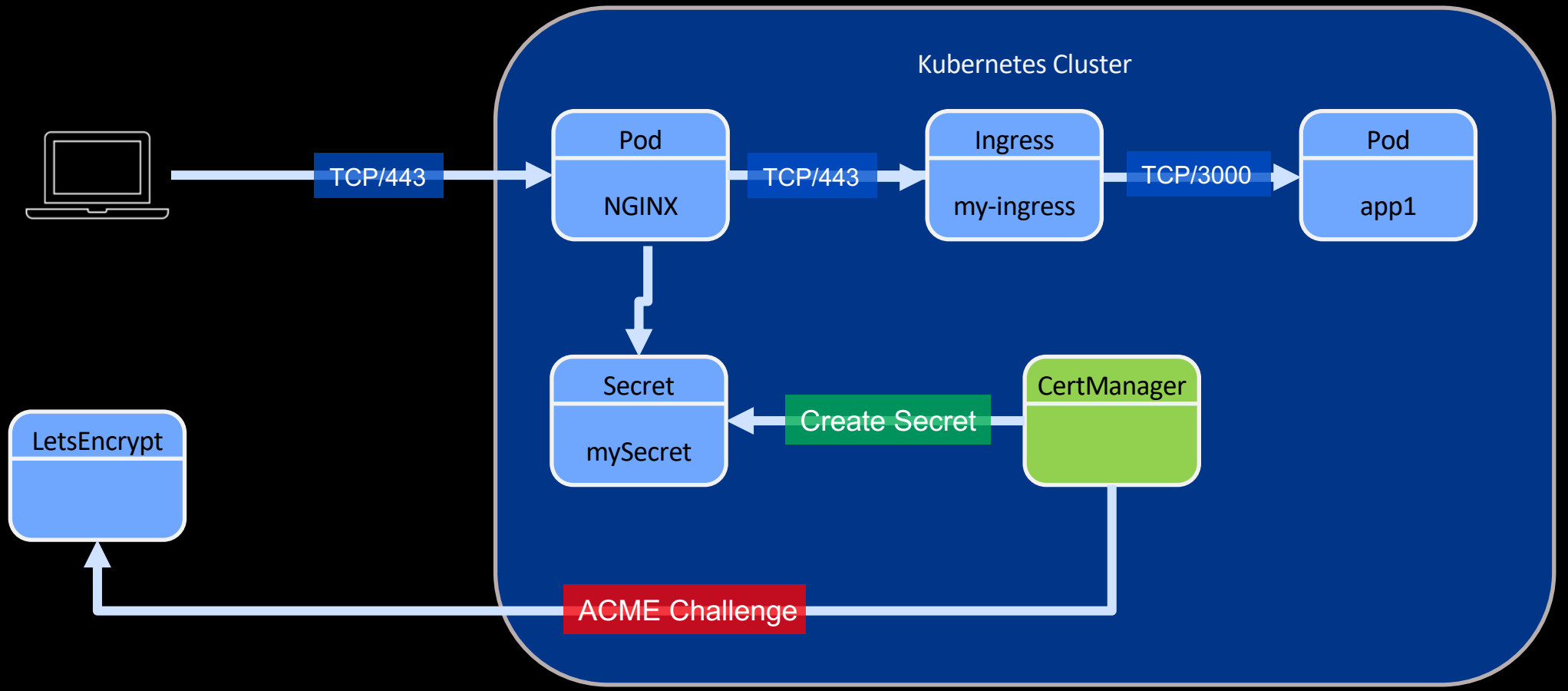
Certificate Management for TLS

cert-manager is a native Kubernetes certificate management controller.
It can help with issuing certificates from a variety of sources, such as Let's Encrypt, HashiCorp Vault, Venafi, a simple signing keypair, or self signed.

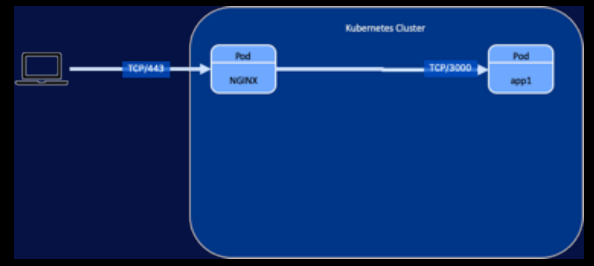
It will ensure certificates are valid and up to date, and attempt to renew certificates at a configured time before expiry.

Tutorial: https://docs.cert-manager.io/en/latest/tutorials/acme/quick-start/index.html#

Kubernetes Security – Mitigation – Network Encryption

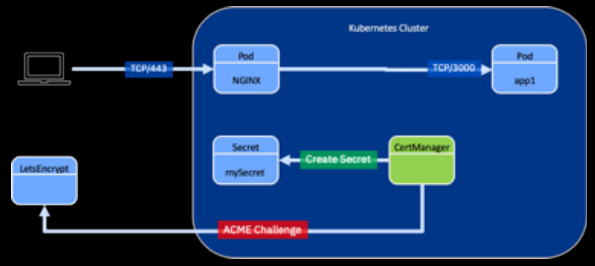


Kubernetes Security – Mitigation – Network Encryption




```
apiVersion: v1
kind: Service
metadata:
  name: app1
  labels:
    app: app1
    tier: frontend
spec:
  type: ClusterIP
  ports:
  - port: 80
    targetPort: 3000
  selector:
    app: app1
    tier: frontend
```

Kubernetes Security – Mitigation – Network Encryption




```yaml
apiVersion: v1
kind: Service
metadata:
  name: app1
  labels:
    app: app1
    tier: frontend
spec:
  type: ClusterIP
  ports:
  - port: 80
    targetPort: 3000
  selector:
    app: app1
    tier: frontend
```

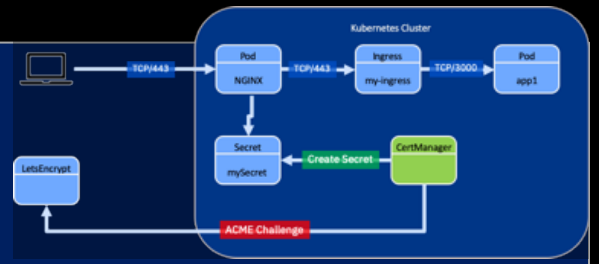
```yaml
apiVersion: certmanager.k8s.io/v1alpha1
kind: ClusterIssuer
metadata:
  name: letsencrypt-staging
spec:
  acme:
    server: https://acme-staging-v02.api.letsencrypt.org/directory
    email: <me@example.com>
    privateKeySecretRef:
      name: mySecret
    http01: {}
```

Kubernetes Security – Mitigation – Network Encryption

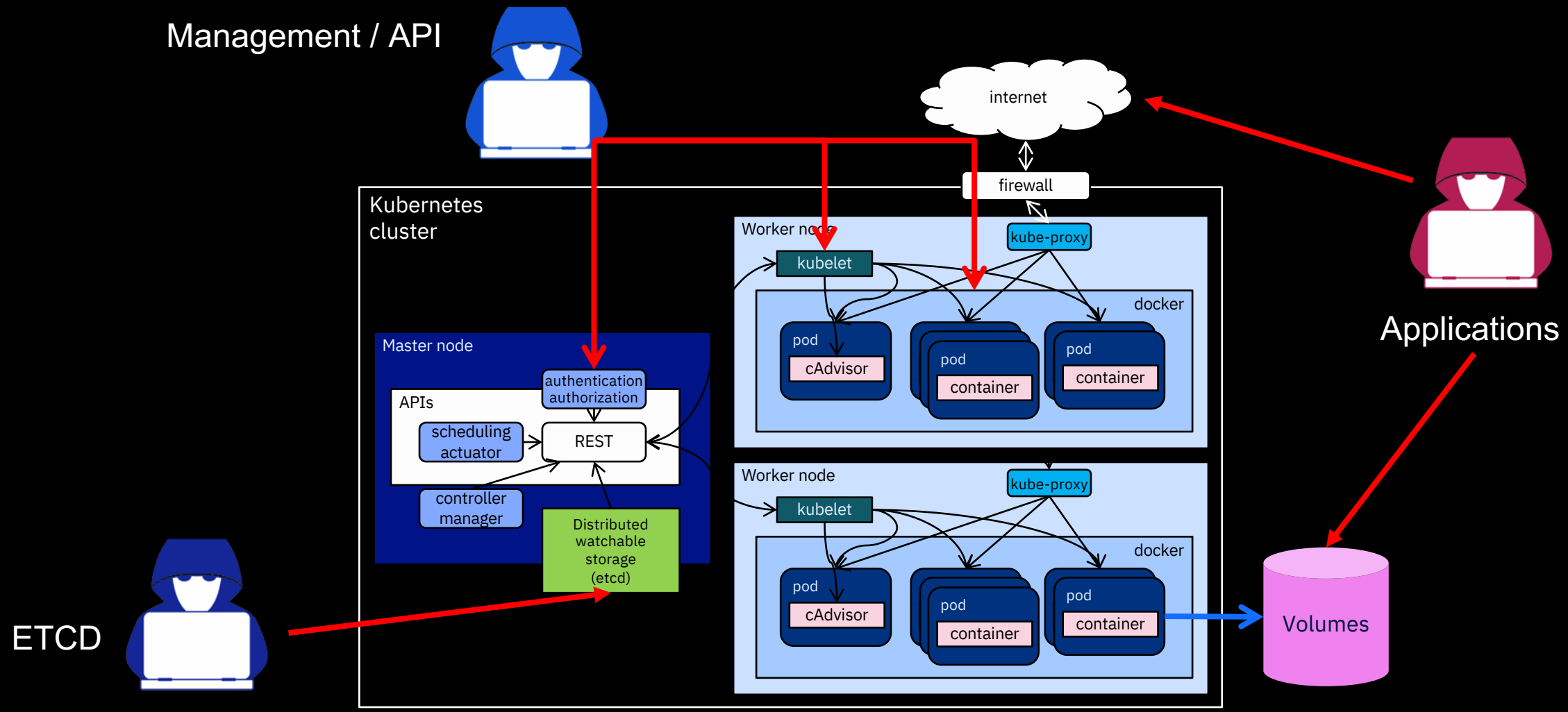
```yaml
apiVersion: certmanager.k8s.io/v1alpha1
kind: ClusterIssuer
metadata:
  name: letsencrypt-staging
spec:
  acme:
    server: https://acme-staging-v02.api.
    email: <me@example.com>
    privateKeySecretRef:
      name: mySecret
    http01: {}
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: app1
  labels:
    app: app1
    tier: frontend
spec:
  type: ClusterIP
  ports:
  - port: 80
    targetPort: 3000
  selector:
    app: app1
    tier: frontend
```

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: my-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
    certmanager.k8s.io/cluster-issuer: letsencrypt-staging
    kubernetes.io/tls-acme: "true"
spec:
  rules:
  - host: app.example.com
    http:
      paths:
      - path: /
        backend:
          serviceName: app1
          servicePort: 80
  tls:
  - secretName: mySecret
    hosts:
    - app.example.com
```



Kubernetes Security – Logging



Logging components

The easiest and most embraced logging method for containerized applications is to write to standard out and standard error

Filebeat: A log data shipper for local files. Filebeat monitors the log directories or specific log files, tails the files, and forwards them either to Elasticsearch and/or Logstash for indexing.

Elasticsearch: An open source full-text search engine based on Lucene. It provides HTTP web interface and schema-free JSON documents.

Logstash: A open source tool for collecting, parsing, and storing logs for future use.

Heapster: The Kubernetes network proxy runs on each node.

Kibana: An open source data visualization plugin for Elasticsearch. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data.

QUESTIONS?

Kubernetes Workshop Series

DevSecOps

03

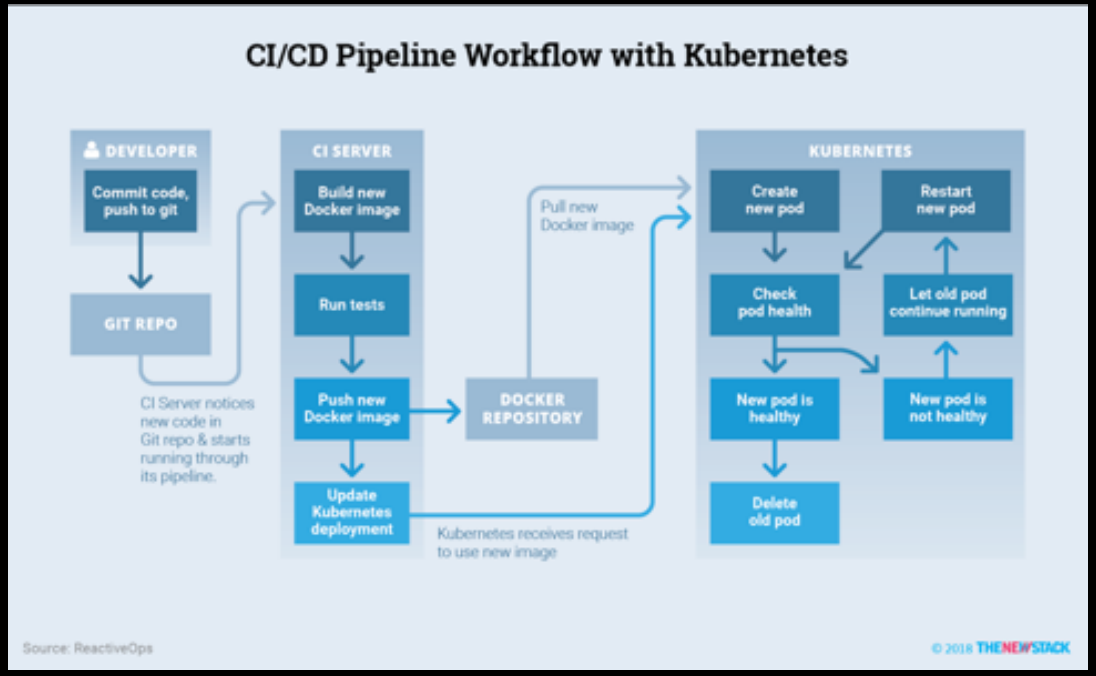
Where is the Sec in DevSecOps

What is DevSecOps?

“bringing operations and development together with security functions”

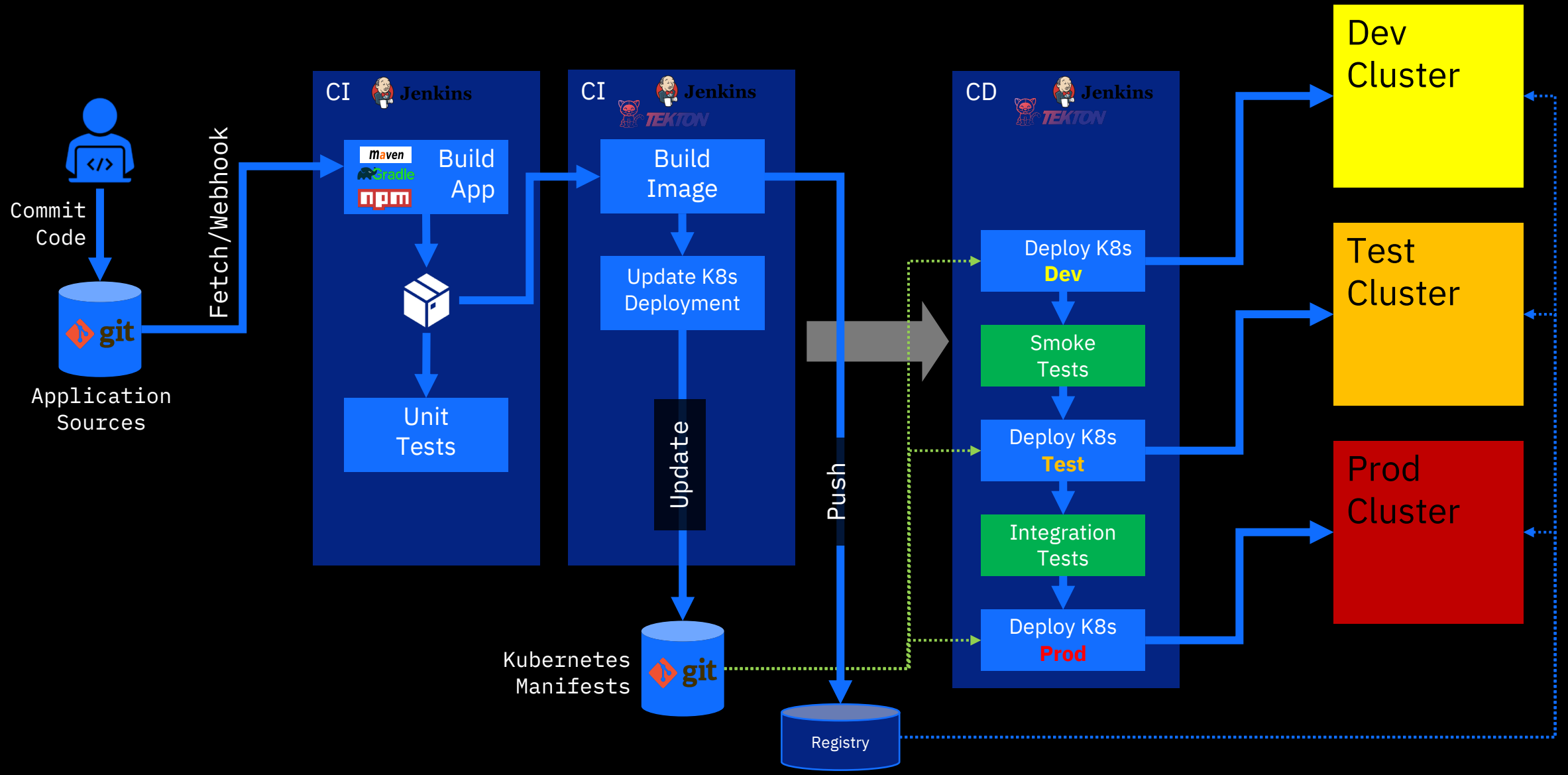
Automate security tasks by embedding security controls early in the DevOps workflow.

Insecure by default

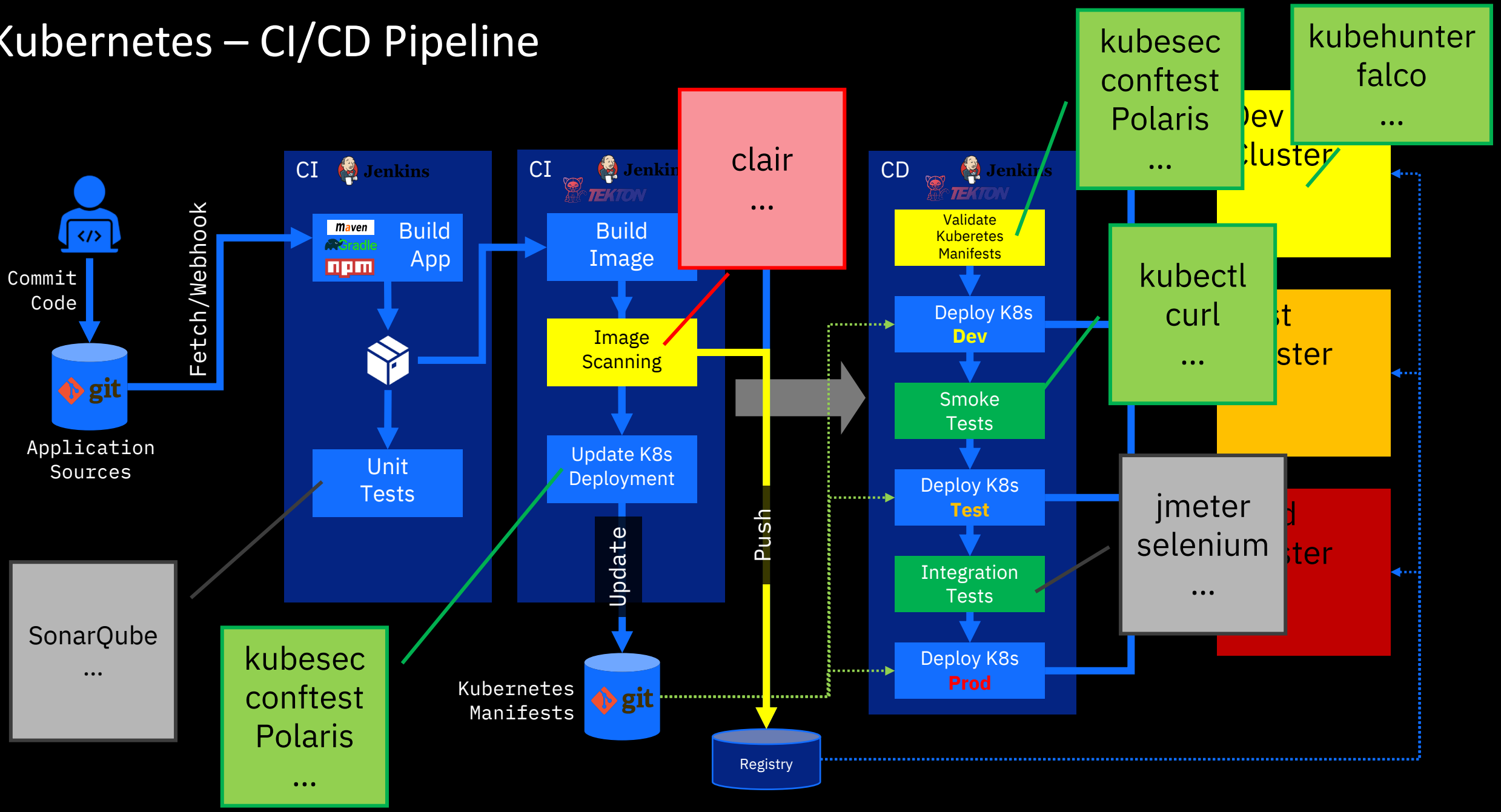


https://theresnomon.co/insecure-by-default-kubernetes-cicd-reference-diagrams-bce31523d506

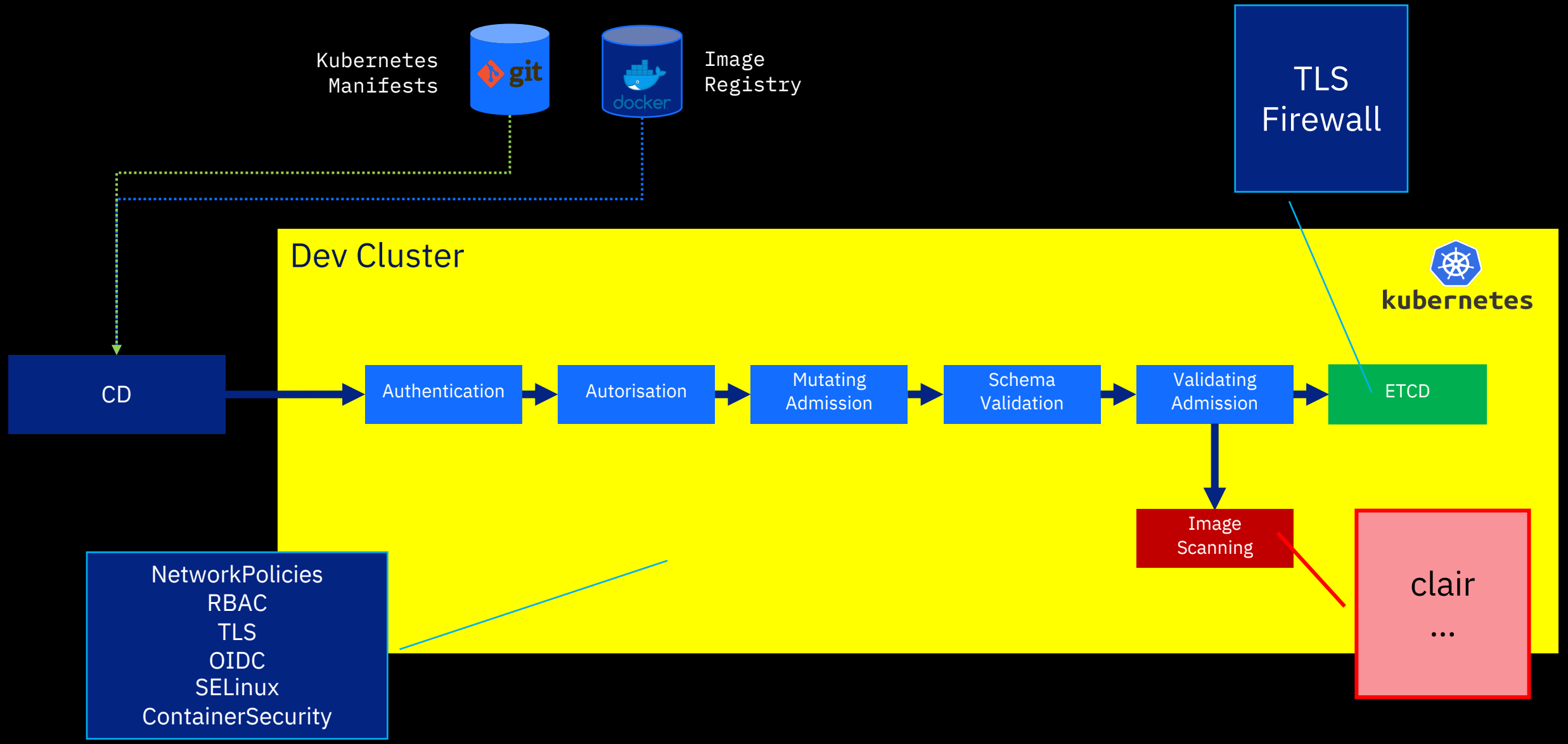
Kubernetes – CI/CD Pipeline - GitOps



Kubernetes – CI/CD Pipeline



Kubernetes – Secure CI/CD Pipeline



QUESTIONS?

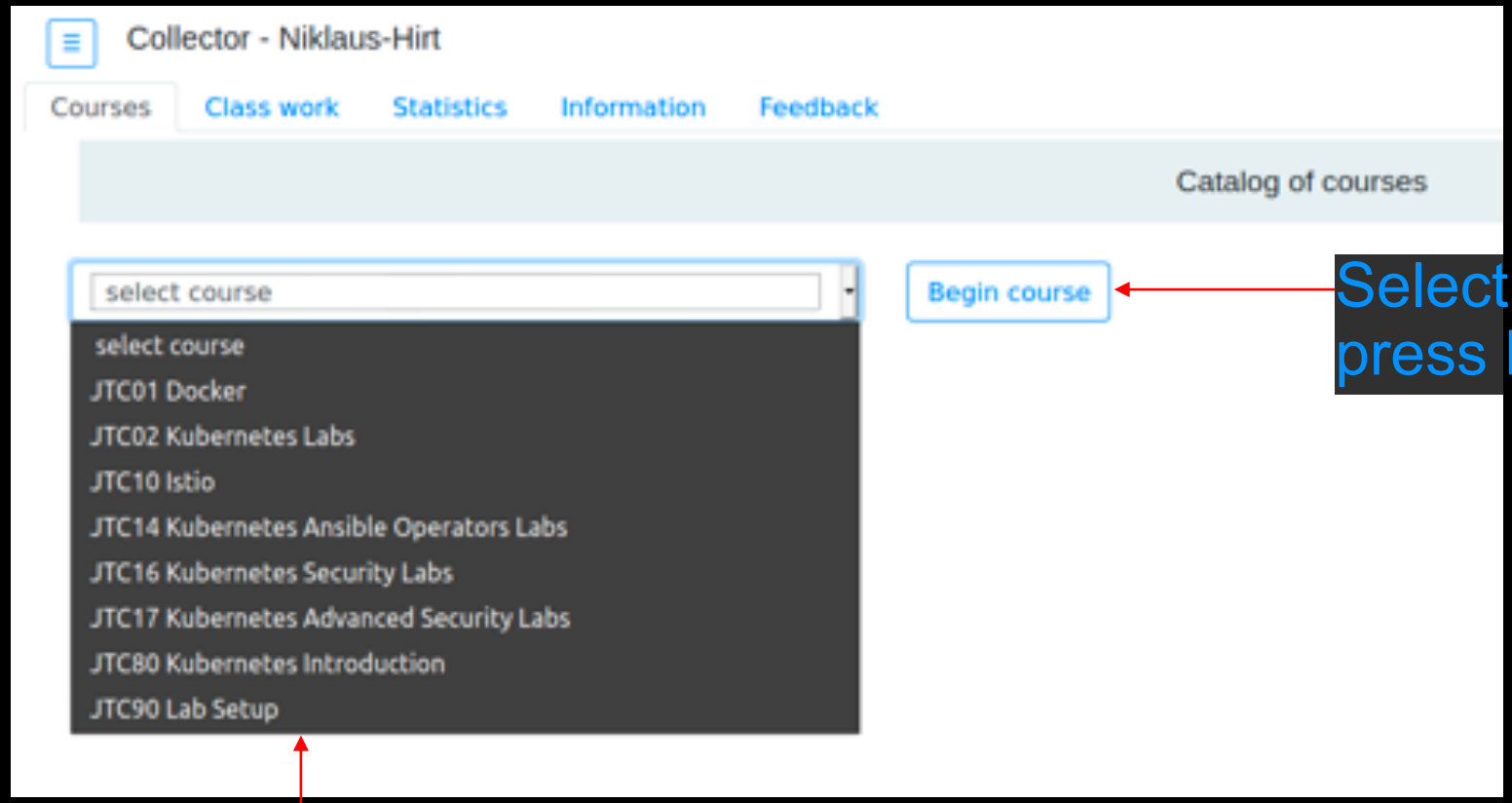
Kubernetes Workshop Series

K8s Security - *Hands-On*

04

Starting Course JTC16 Kubernetes Security

Name will be shown



Select course and press button to begin

Current course catalog

JTC16 Kubernetes Security Labs

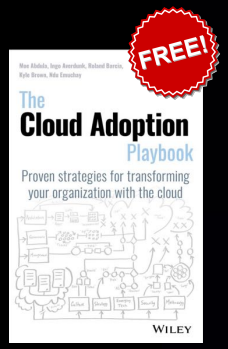
Lab 1 : NetworkPolicies

Lab 2 : Security Tooling

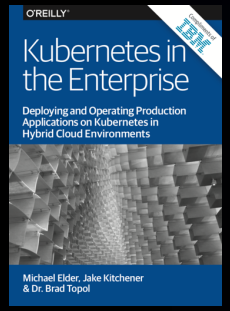
Kubernetes Security – Some Reading Tips

https://kubernetespodcast.com/episode/065-attacking-and-defending-kubernetes/
https://en.wikipedia.org/wiki/Red_team
https://blog.ropnop.com/attacking-default-installs-of-helm-on-kubernetes/
https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/
https://kubernetes.io/blog/2019/03/21/a-guide-to-kubernetes-admission-controllers/
https://kubernetes.io/docs/tutorials/clusters/apparmor/
https://kubernetes.io/docs/concepts/policy/pod-security-policy/
https://kubernetes.io/docs/concepts/storage/volumes/#hostpath

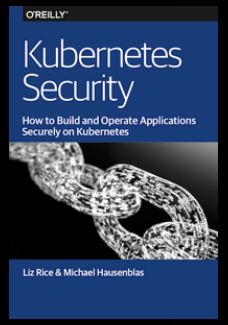
Kubernetes – Some Reading Tips



The de facto guide to improving your enterprise with the cloud, created by distinguished members of our Solution Engineering team
http://ibm.biz/playbook



Deploying and Operating Production Applications on Kubernetes in Hybrid Cloud Environments
https://ibm.co/2LQketN (excerpt)



https://kubernetes-security.info/

Sources and documentation will be available here:

https://github.com/niklaushirt/k8s_training_public

https://github.com/niklaushirt/training

See you next week!

- **Same place**
- **Same time**

Kubernetes Workshop Series

Kubernetes Security Advanced

READY SET GO!!!!

Duration: 60 mins

QUESTIONS?

Niklaus Hirt

nikh@ch.ibm.com

@nhirt

THANK YOU!!!!