Kubernetes Workshop Series

JTC17

Kubernetes Security Advanced 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 - Advanced Concepts

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

Videos, sources and documentation will be available here:

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

https://github.com/niklaushirt/k8s_training_public

https://github.com/niklaushirt/training

Session Quiz & Feedback

We will collect some feedback.

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

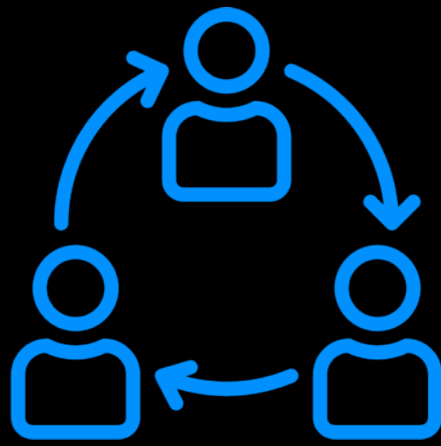
Prepare the Labs

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

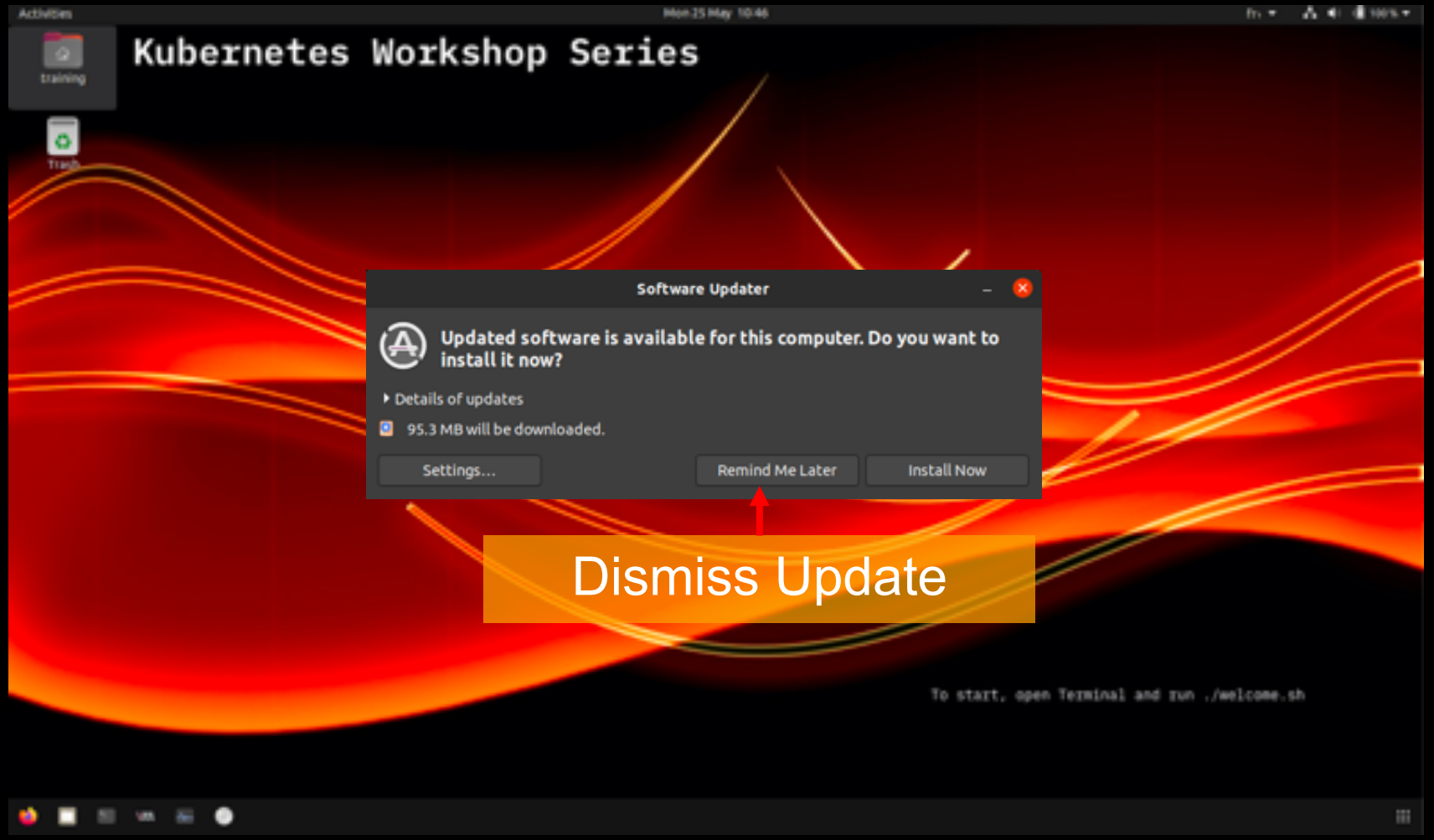
Task 1: Download Training VM

Task 2: Setup VMWare / VirtualBox

Task 3: Start Training VM

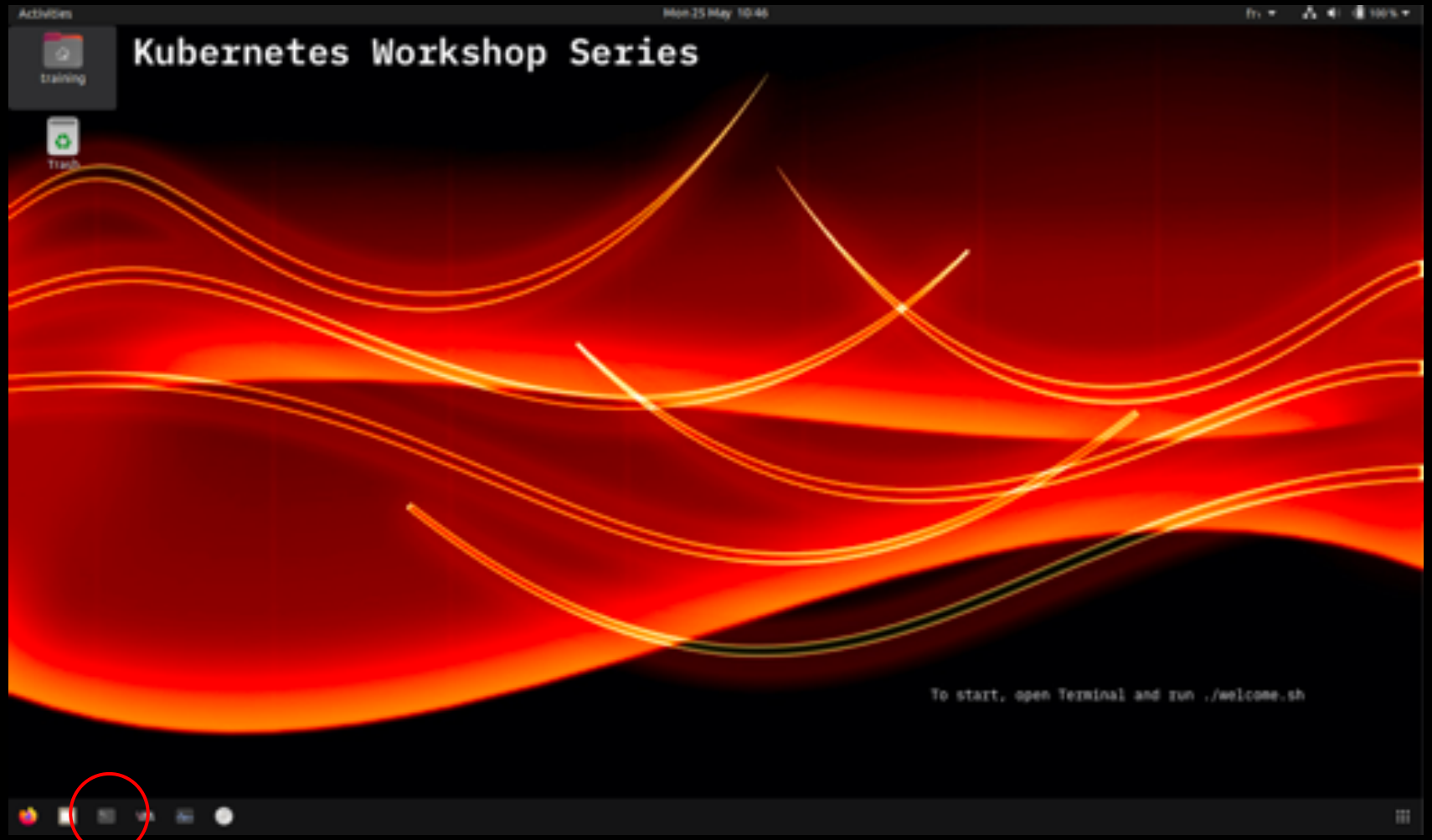
Task 4: Login / Check

Accessing your Personal Training Environment



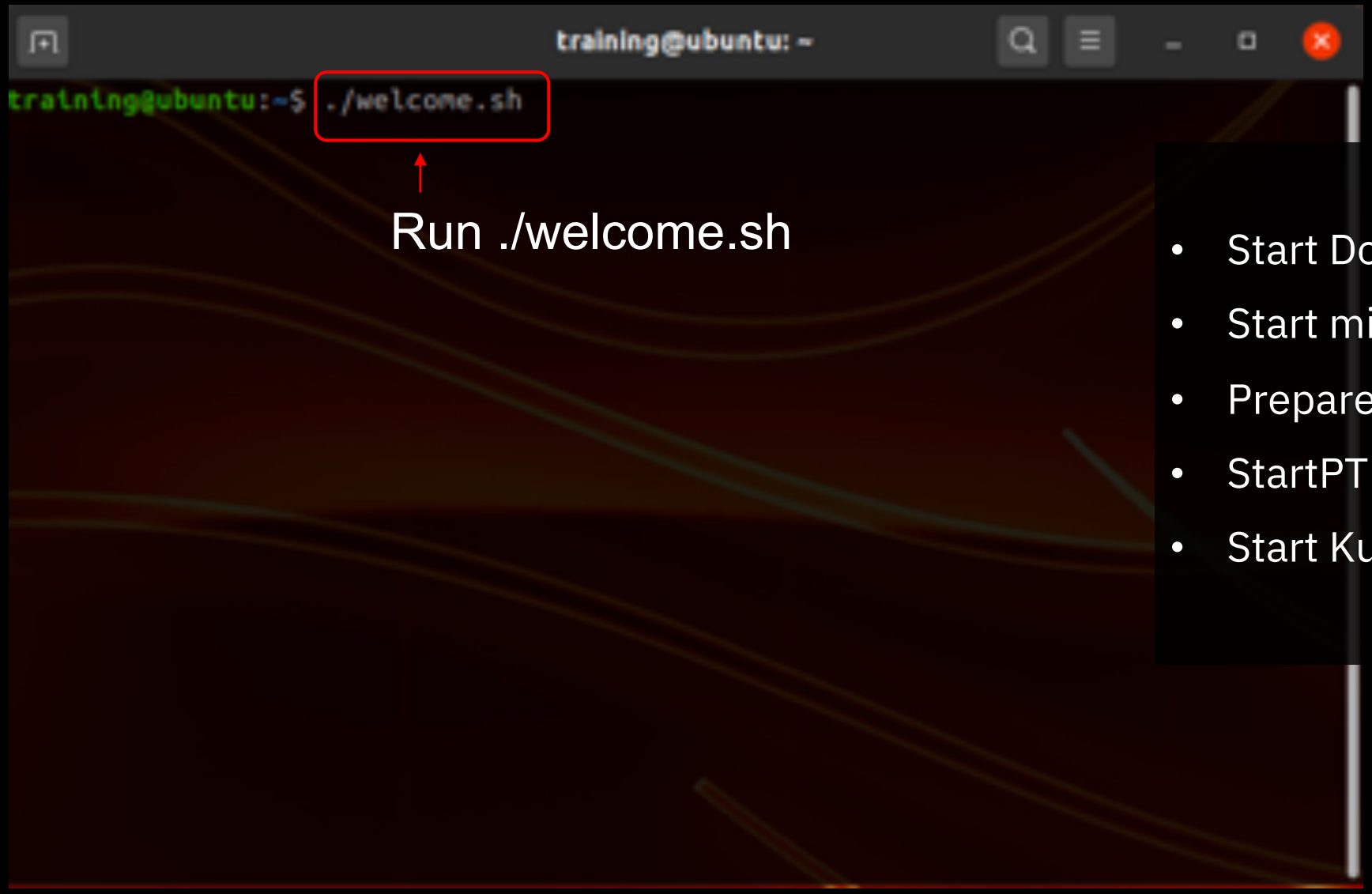
Accessing your Personal Training Environment

Accessing your Personal Training Environment



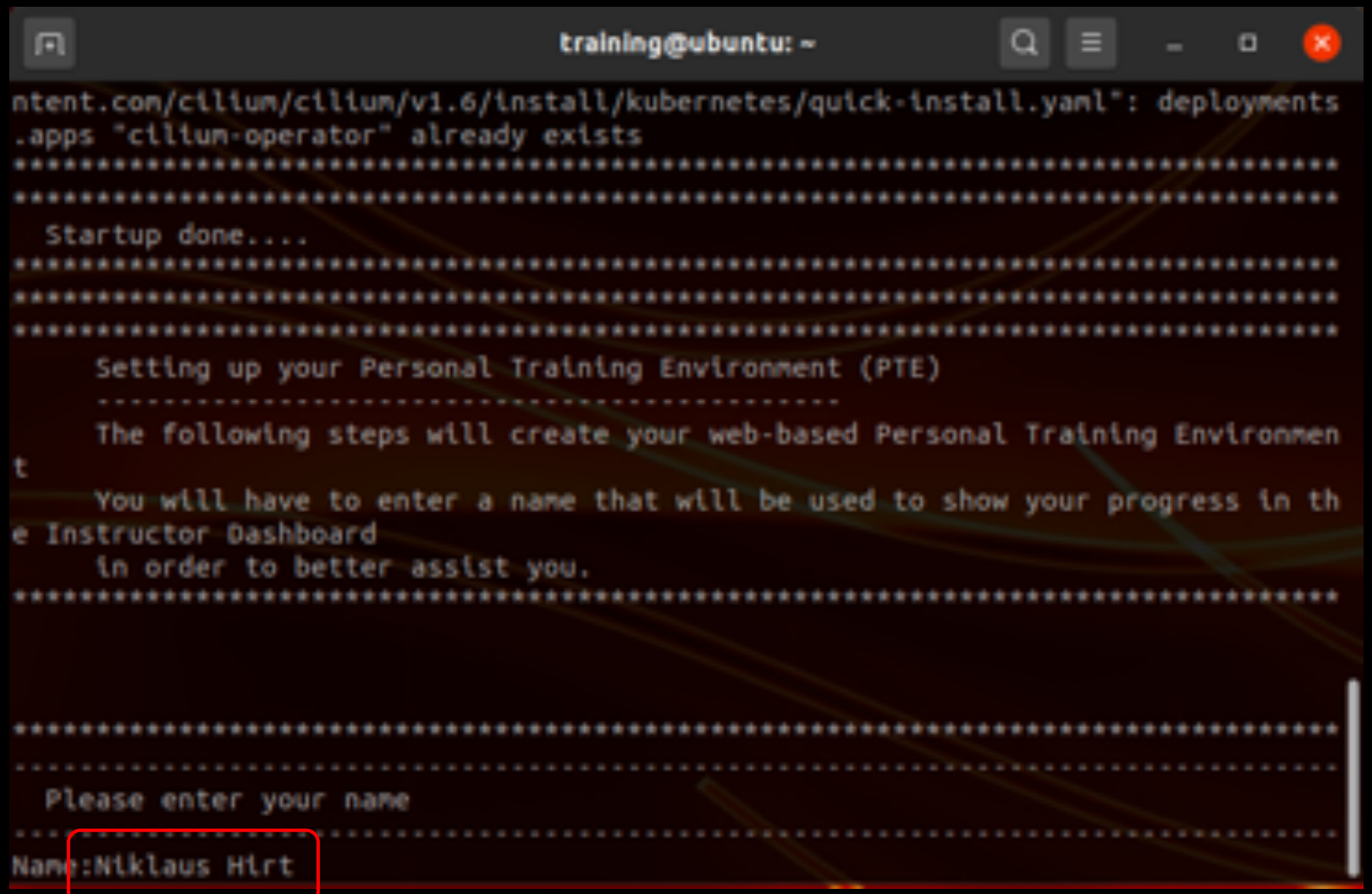
Start Terminal

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
   ```

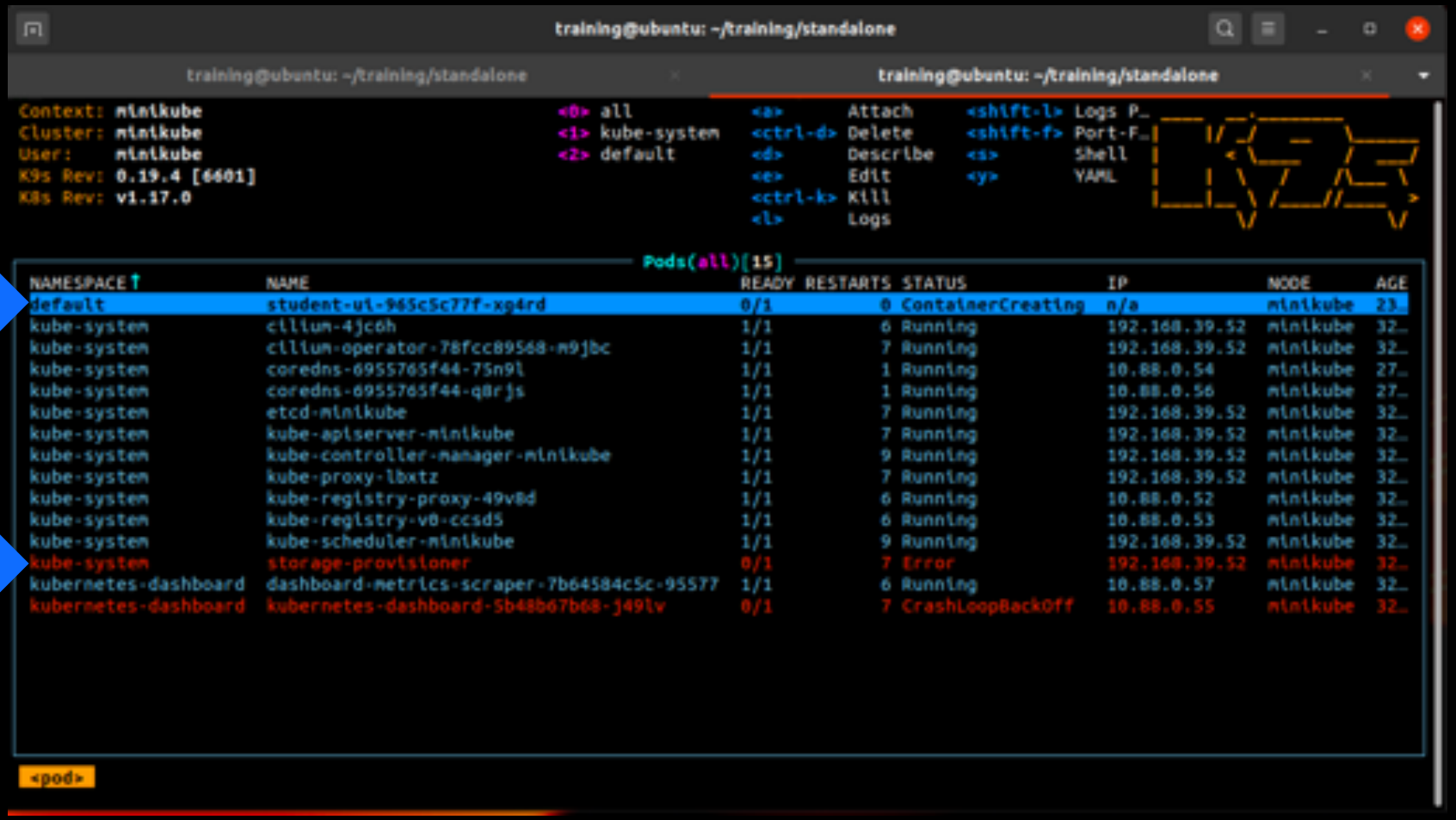
Accessing your Personal Training Environment

Troubleshooting

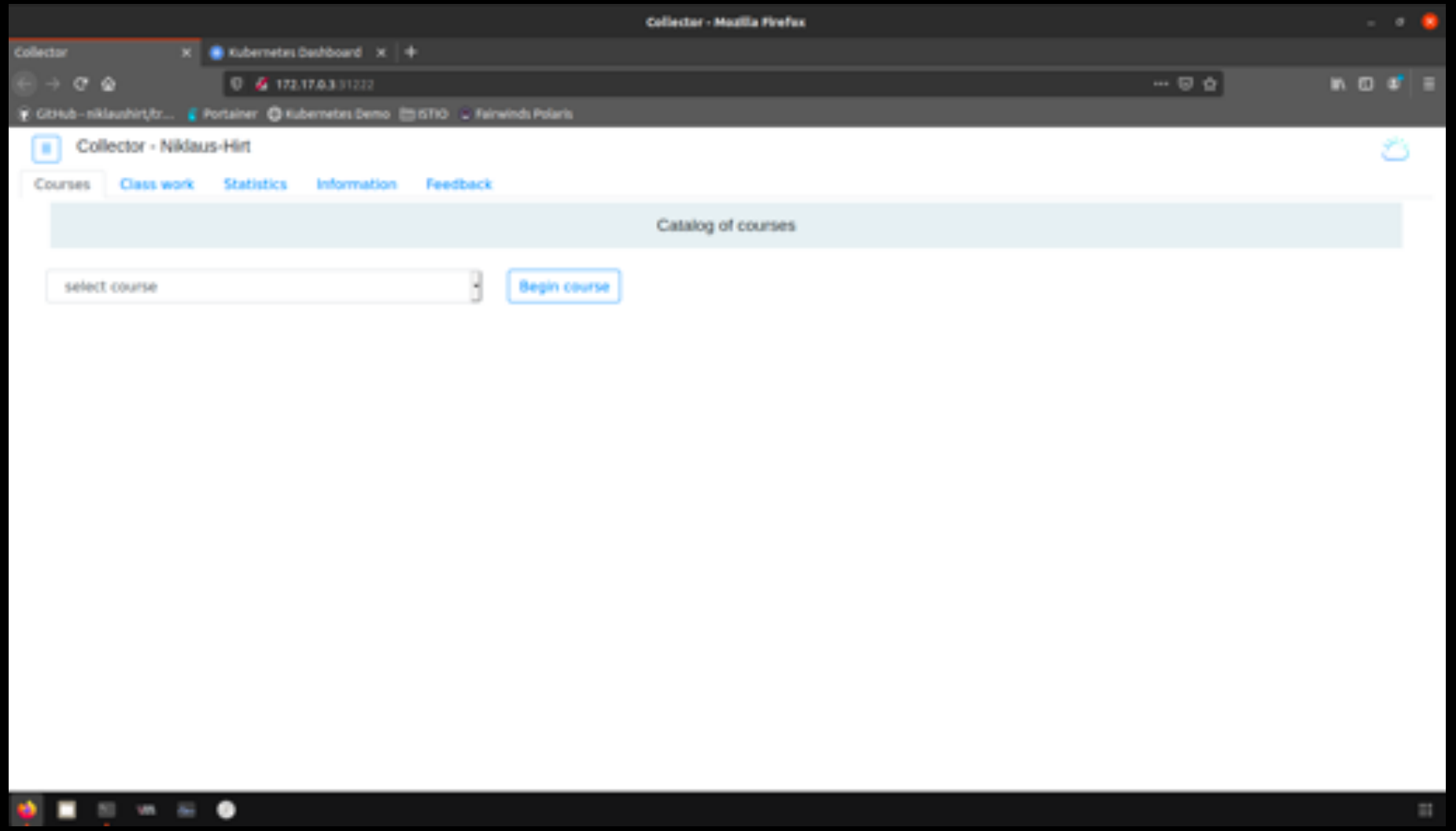
- Run ***k9s*** in the Terminal – wait for all the pods to be Running (blue – 1/1)

Image pulling - wait

Dependencies - wait



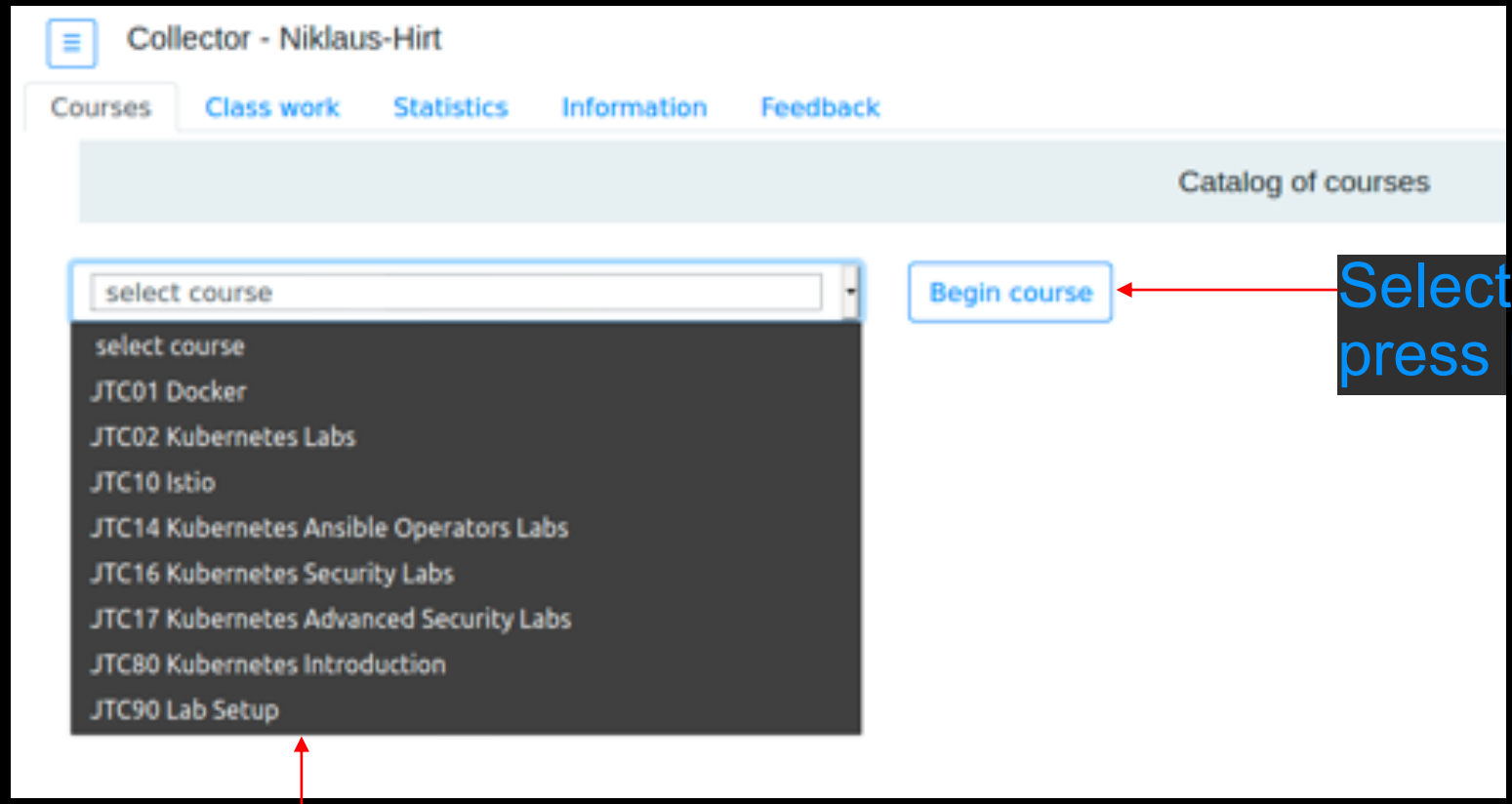
Accessing your Personal Training Environment



When completed, your PTE and Kubernetes Dashboard will open automatically

Accessing your Personal Training Environment

Name will be shown

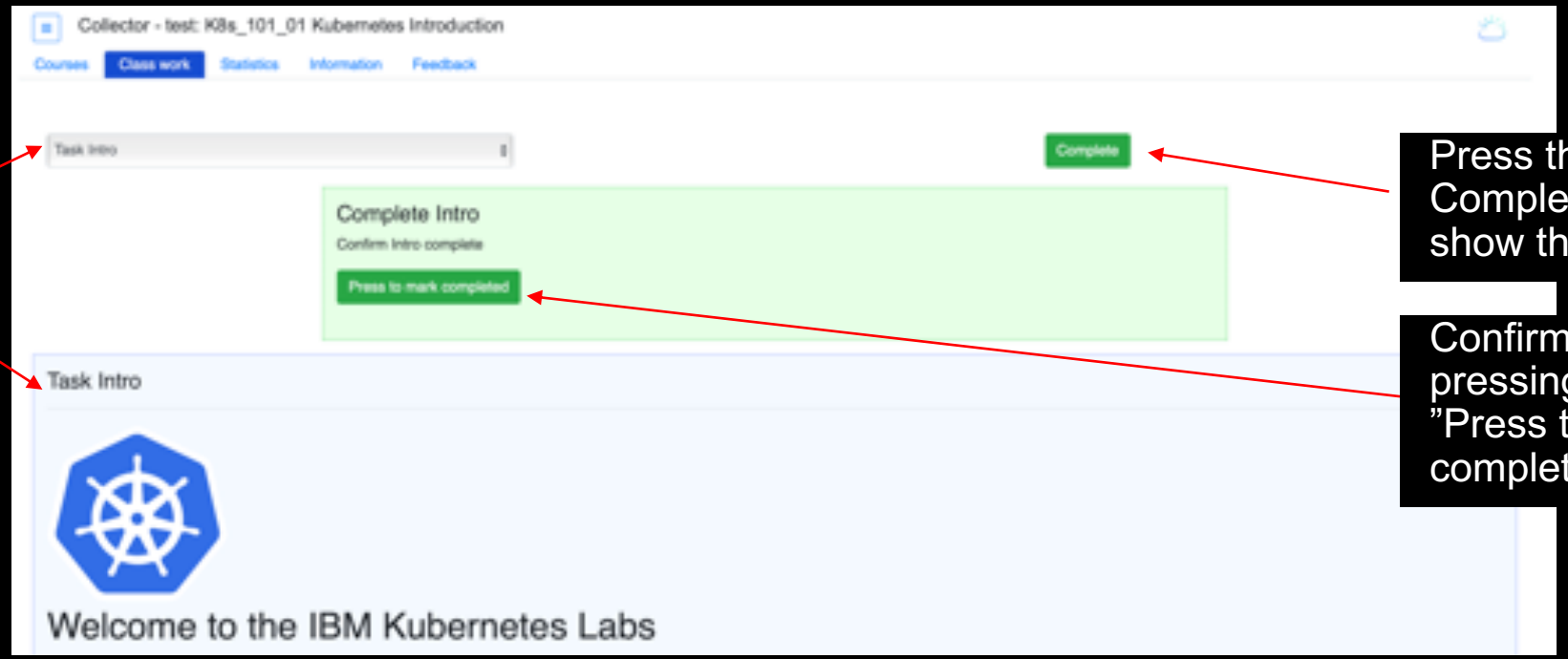


Select course and press button to begin

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

QUESTIONS?

Kubernetes Workshop Series

Security Introduction

01

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

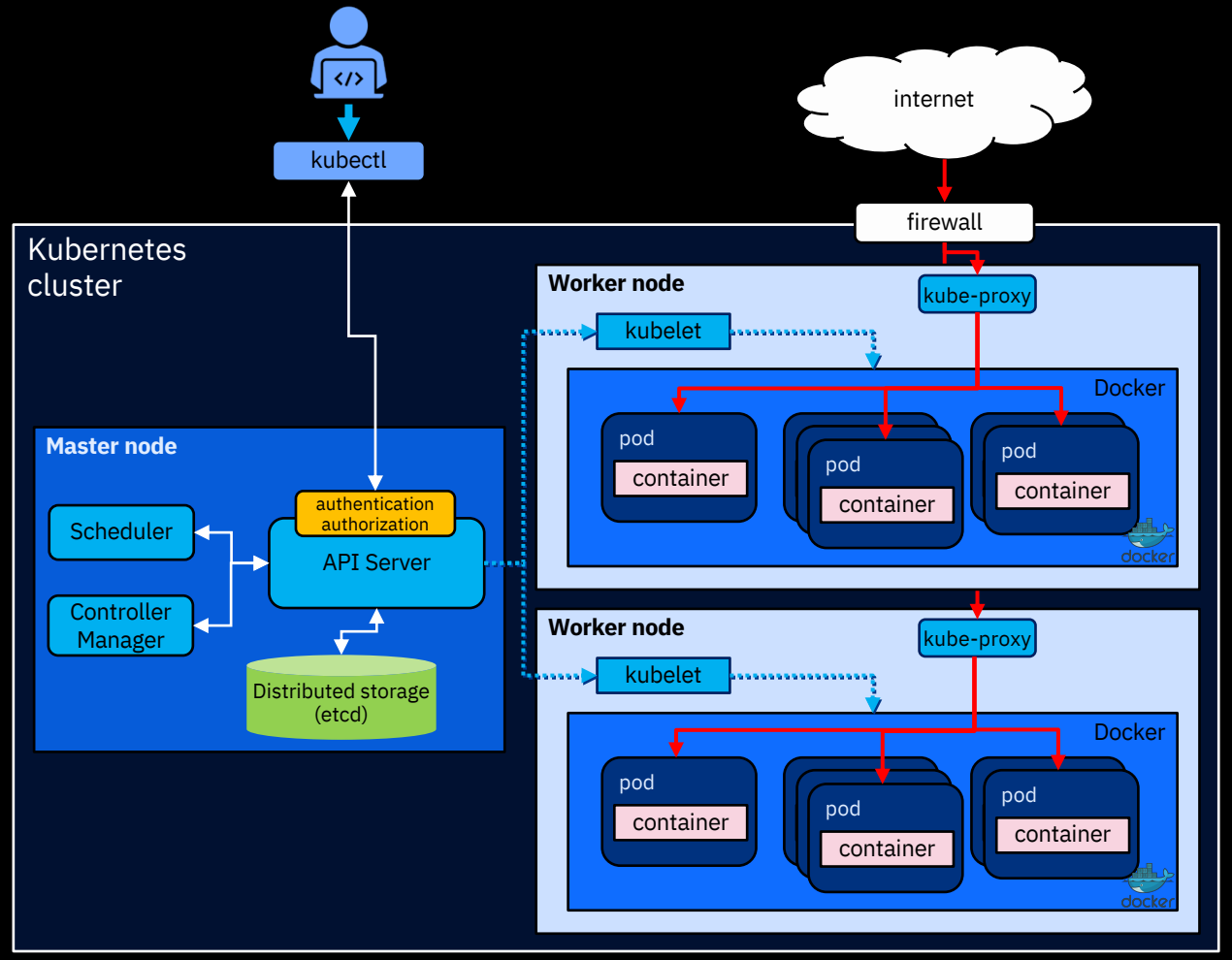
What happened so far...
Kubernetes Cluster Architecture

Master node

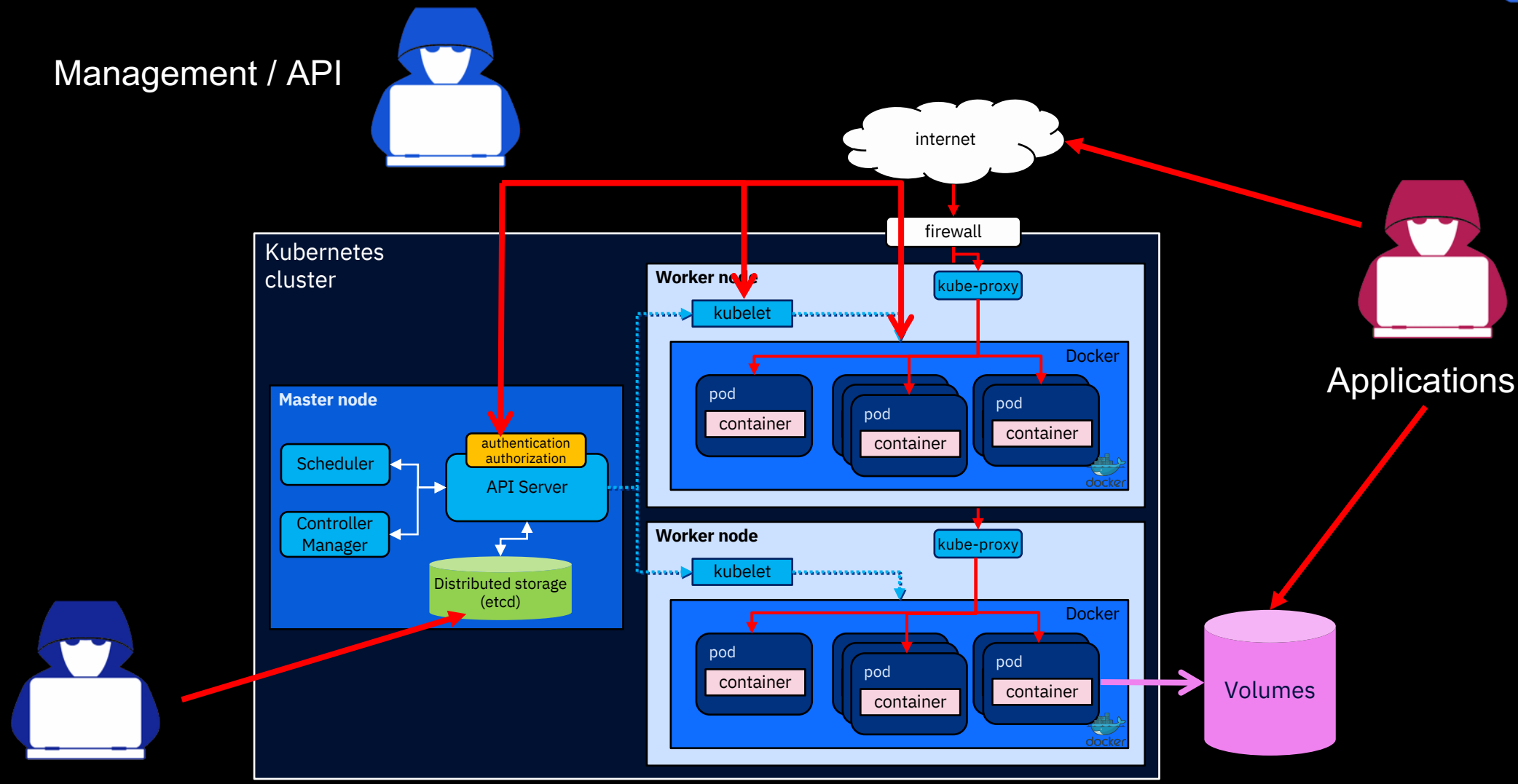
- Node that manages the cluster
- Scheduling, replication & control
- Multiple nodes for HA

Worker nodes

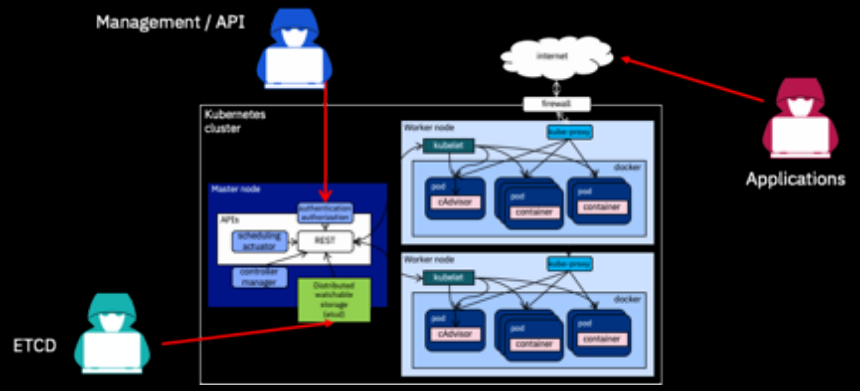
- Node where pods are run
- Docker engine
- kubelet agent accepts & executes commands from the master to manage pods
- kube-proxy – routes inbound or ingress traffic



Kubernetes – **Security – Attack surface**



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
- Use Kubernetes secrets

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?

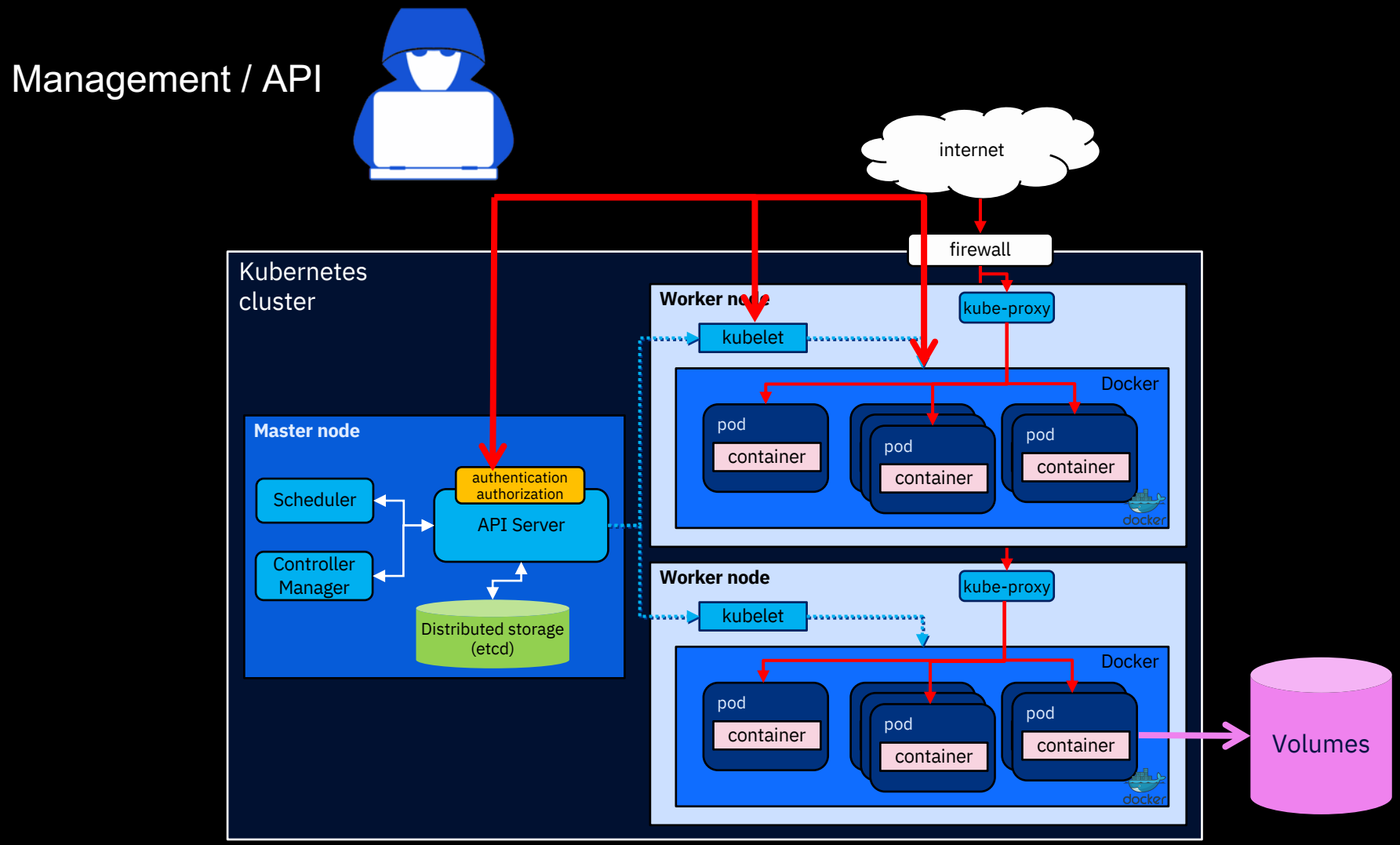
Kubernetes Workshop Series

Security Elements

02

Securing the API

Kubernetes Security – Mitigation – API

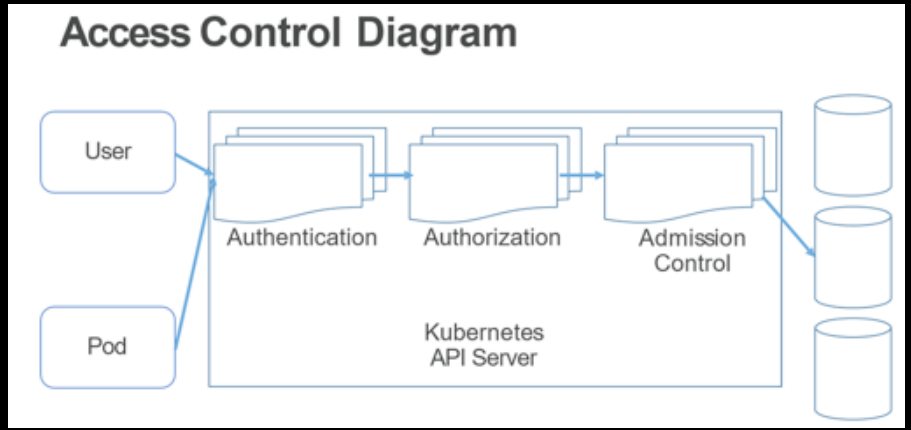


Kubernetes – Core Services - Controlling K8s Access

Authentication – WHO am I - (token, certs, OpenID Connect Provider (OIDC)…)
Developer — create, view, get permission.
Tester — create, delete, update, get permission.
Administrator — All permission

Authorization – CAN I - (RBAC)
Is the user or application authorized or have permissions to access a K8s object?

Admission Control
Intercepts requests to the Kubernetes API server prior to persistence of the object, but after the request is authenticated and authorized

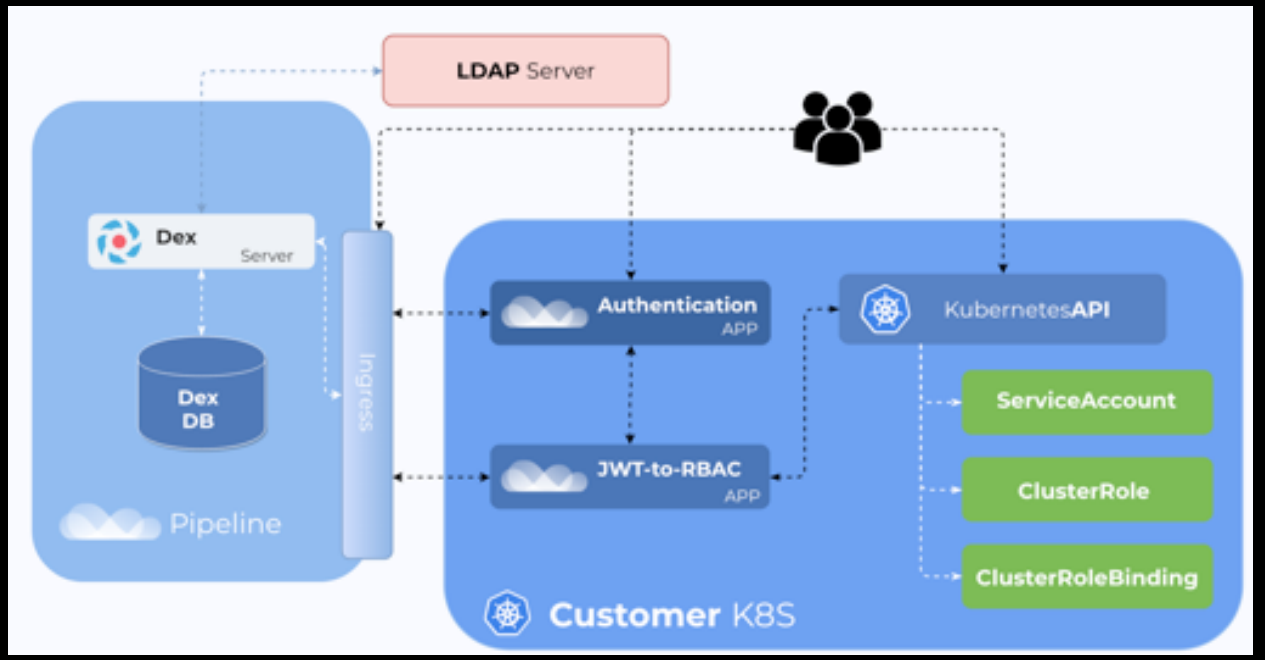


Source: https://kubernetes.io/docs/reference/access-authn-authz/controlling-access/

Integrating with LDAP/AD

OpenID Connect
ID Tokens are JSON Web Tokens (JWTs)

dex
Dex acts as a portal to other identity providers through connectors
dex defer authentication to LDAP servers, SAML providers, or established identity providers like GitHub, Google, and Active Directory.

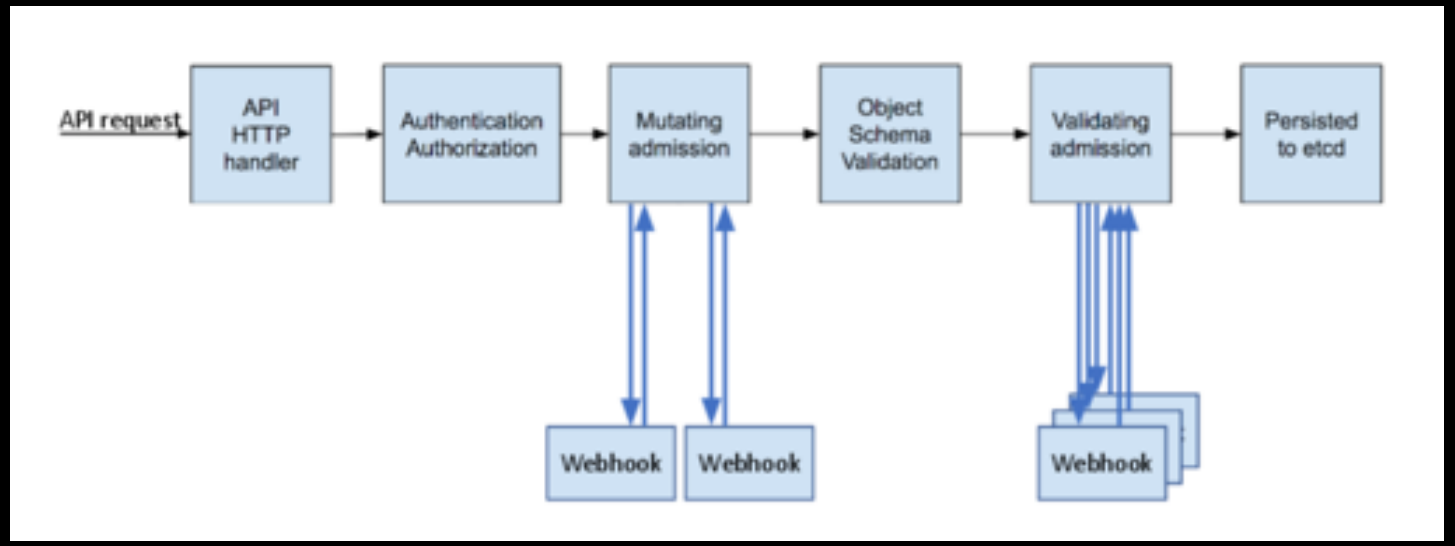


Source: https://github.com/krishnapmv/k8s-ldap and https://github.com/dexidp/dex

Kubernetes WebHooks

WebHooks Type

- **mutating**: to dynamically change incoming deployments on the fly (think automatic Istio sidecar injection), and
- **validating**: to accept or reject those same deployments based on the rules in your callback.
- → Image Scanner



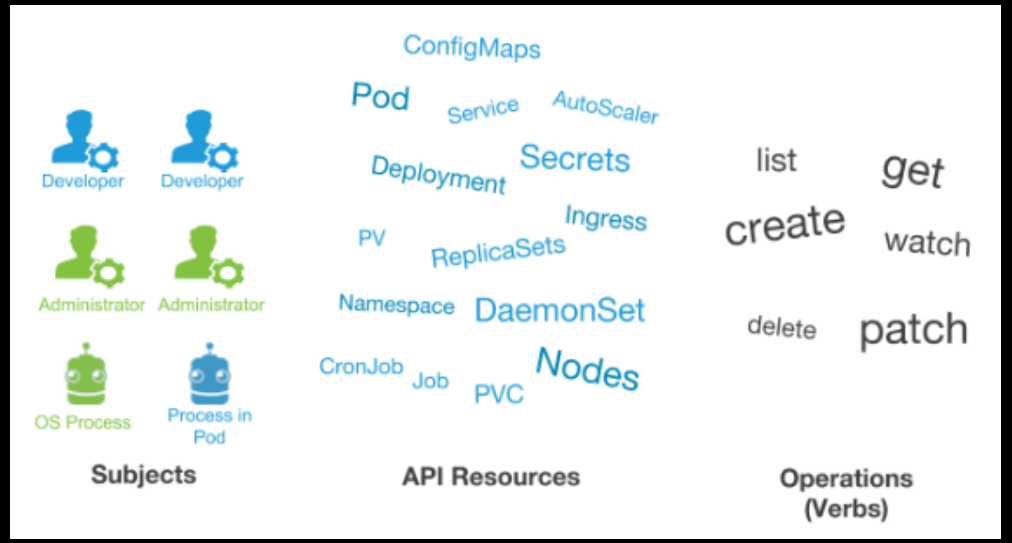
Source: https://github.com/krishnapmv/k8s-ldap and https://github.com/dexidp/dex

RBAC Basic Elements

RBAC Building Blocks

Objects
- Pods
- PersistentVolumes
- ConfigMaps
- Deployments
- Nodes
- Secrets
- Namespaces

Verbs
- create
- get
- delete
- list
- update
- edit
- watch
- exec



Source: https://kubernetes.io/docs/reference/access-authn-authz/rbac/

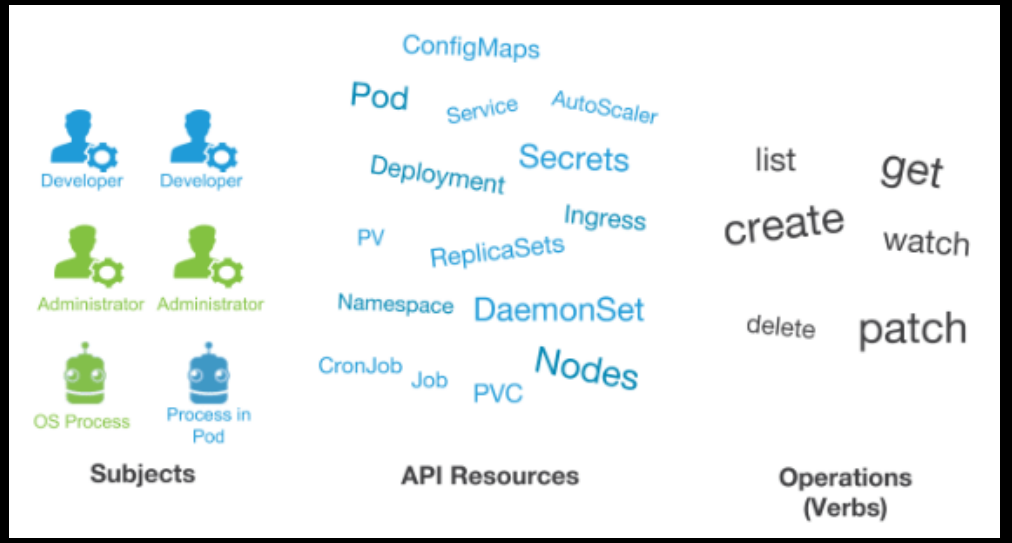
RBAC Basic Elements

Rules
Operations (verbs) that can be carried out on a group of resources which belong to different API Groups.

Roles and **ClusterRoles**
Both consist of rules.
- Role: applicable to a single namespace
- ClusterRole: is cluster-wide

Subjects
Entity that attempts an operation in the cluster
- User Accounts: Humans or processes living outside the cluster.
- Service Accounts: Namespaced account.
- Groups: Reference multiple accounts. (default groups like cluster-admin)

RoleBindings and **ClusterRoleBindings**
Bind subjects to roles



Source: https://kubernetes.io/docs/reference/access-authn-authz/rbac/

RBAC Example


```
kind: Role
metadata:
  name: pod-reader
rules:
- apiGroups: [“”]
  resources: [“pods”]
  verbs: [”get”,”list”]
```

```
kind: RoleBinding
metadata:
  name: pod-reader
subjects:
- kind: User
  name: John
roleRef:
  kind: Role
  name: pod-reader
```

Source: https://kubernetes.io/docs/reference/access-authn-authz/rbac/

Common RBAC Pitfalls

- **Cluster Administrator Role Granted Unnecessarily**
  The built-in **cluster-admin** role grants effectively unlimited access to the cluster.
  Should be granted only to the specific users that need it.

- **Duplicated Role Grant**
  Role definitions may overlap with each other, making access revocation more difficult.

- **Unused Role**
  Roles that are created but not granted to any subject can increase the complexity of RBAC management.

- **Grant of Missing Roles**
  Role bindings can reference roles that do not exist.
  If the role name is reused those role bindings can unexpectedly become active.

Source: https://www.cncf.io/blog/2019/09/16/5-kubernetes-rbac-mistakes-you-must-avoid/

Always adapt least privileged access practices

Audit RBAC – rbac-lookup

Some tools to help you see more clearly

```
~ rbac-lookup openshift
SUBJECT                                                 SCOPE                                   ROLE
system:serviceaccounts:openshift                        openshift                               ClusterRole/system:image-puller
system:serviceaccounts:openshift-console                openshift-console                       ClusterRole/system:image-puller
system:serviceaccounts:openshift-console                openshift-console                       ClusterRole/system:image-puller
system:serviceaccounts:openshift-infra                  openshift-infra                         ClusterRole/system:image-puller
system:serviceaccounts:openshift-logging                openshift-logging                       ClusterRole/system:image-puller
system:serviceaccounts:openshift-logging                openshift-logging                       ClusterRole/system:image-puller
system:serviceaccounts:openshift-metrics-server         openshift-metrics-server                ClusterRole/system:image-puller
system:serviceaccounts:openshift-metrics-server         openshift-metrics-server                ClusterRole/system:image-puller
system:serviceaccounts:openshift-monitoring             openshift-monitoring                    ClusterRole/system:image-puller
system:serviceaccounts:openshift-monitoring             openshift-monitoring                    ClusterRole/system:image-puller
system:serviceaccounts:openshift-node                   openshift-node                          ClusterRole/system:image-puller
system:serviceaccounts:openshift-sdn                    openshift-sdn                           ClusterRole/system:image-puller
system:serviceaccounts:openshift-sdn                    openshift-sdn                           ClusterRole/system:image-puller
system:serviceaccounts:openshift-template-service-broker openshift-template-service-broker     ClusterRole/system:image-puller
system:serviceaccounts:openshift-template-service-broker openshift-template-service-broker     ClusterRole/system:image-puller
system:serviceaccounts:openshift-web-console            openshift-web-console                   ClusterRole/system:image-puller
system:serviceaccounts:openshift-web-console            openshift-web-console                   ClusterRole/system:image-puller
```

```
~ rbac-lookup root
SUBJECT   SCOPE          ROLE
root      cluster-wide   ClusterRole/cluster-admin
```

Example Usage

- rbac-lookup root
- rbac-lookup openshift

Source: https://github.com/FairwindsOps/rbac-lookup

Audit RBAC - rakkess

Some tools to help you see more clearly

```
# rakkess --namespace default
NAME                                            LIST  CREATE  UPDATE  DELETE
bindings                                              ✔
configmaps                                      ✔     ✔       ✔       ✔
controllerrevisions.apps                        ✔     ✔       ✔       ✔
cronjobs.batch                                  ✔     ✔       ✔       ✔
daemonsets.apps                                 ✔     ✔       ✔       ✔
daemonsets.extensions                           ✔     ✔       ✔       ✔
deployments.apps                                ✔     ✔       ✔       ✔
deployments.extensions                          ✔     ✔       ✔       ✔
endpoints                                       ✔     ✔       ✔       ✔
events                                          ✔     ✔       ✔       ✔
events.events.k8s.io                            ✔     ✔       ✔       ✔
horizontalpodautoscalers.autoscaling            ✔     ✔       ✔       ✔
ingresses.extensions                            ✔     ✔       ✔       ✔
jobs.batch                                      ✔     ✔       ✔       ✔
leases.coordination.k8s.io                      ✔     ✔       ✔       ✔
limitranges                                     ✔     ✔       ✔       ✔
localsubjectaccessreviews.authorization.k8s.io        ✔
networkpolicies.extensions                      ✔     ✔       ✔       ✔
```

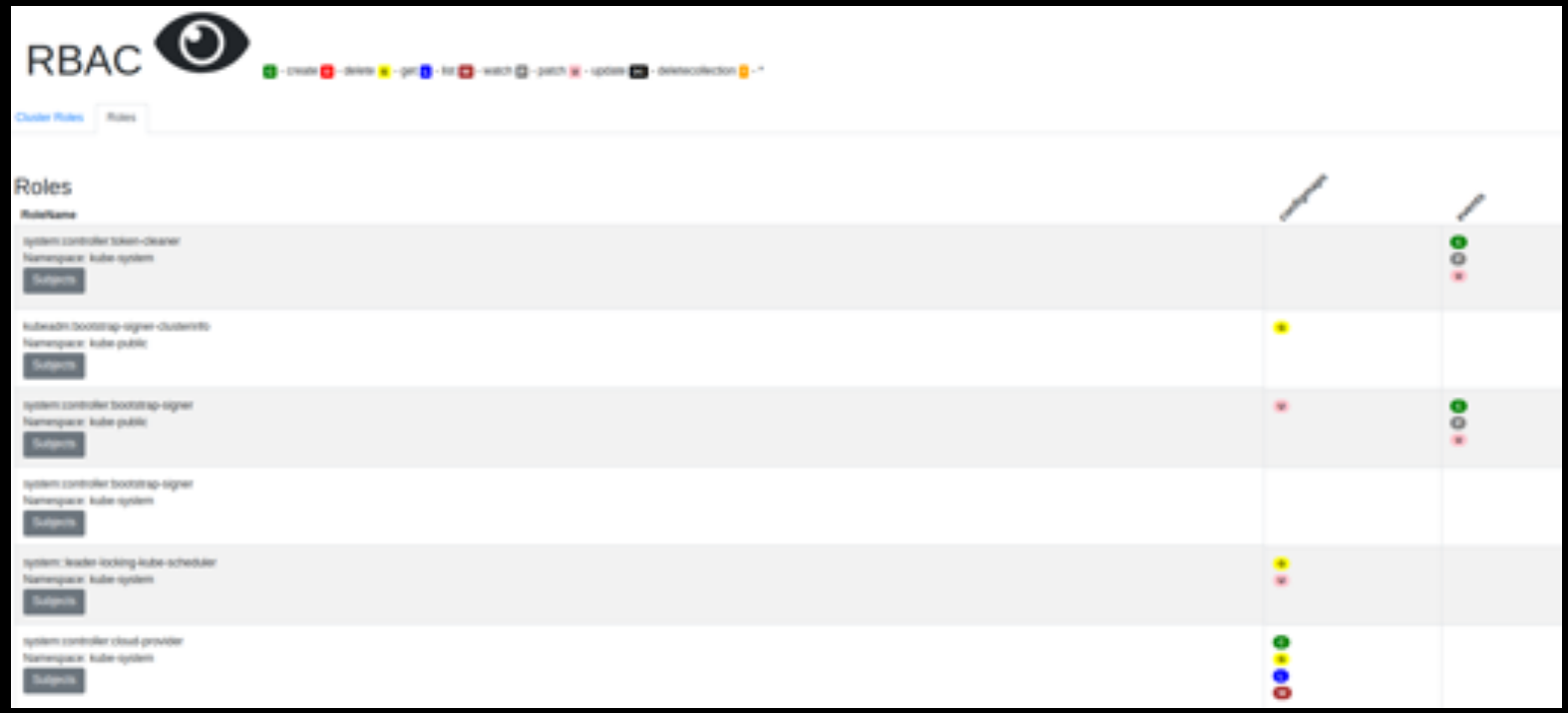
Example Usage

- rakkess --namespace default
- rakkess --verbs get,delete,watch,patch

Source: https://github.com/corneliusweig/rakkess

Audit RBAC – rbac-view

Some tools to help you see more clearly



Example Usage

./rbac-view --render html (default)

./rbac-view --render json

Source: https://github.com/jasonrichardsmith/rbac-view

Audit RBAC – rback

Some tools to help you see more clearly

Example Usage

./rbac-view --render html (default)

./rbac-view --render json

Source: https://github.com/team-soteria/rback

Kubernetes Security – Anonymous Access

Disable Anonymous access

By default, requests to the K8s API and kubelet's HTTPS endpoint that are not rejected by other configured authentication methods are treated as anonymous requests, and given a username of `system:anonymous` and a group of `system:unauthenticated`.

Anonymous authentication provides full access to the *kubelet* API, the only requirement being network access to the service.

To disable anonymous access pass `--anonymous-auth=false` to the API server and start the kubelet with the `--anonymous-auth=false` flag

If RBAC is enabled this risk is mitigated.

Kubernetes Security – Kubelet

Secure kubelet

```
Set --anonymous-auth=false
Set --authorization-mode to something other than AlwaysAllow
Set --read-only-port=0
```

Kubernetes Security – etcd

Secure etcd

```
Set --cert-file and --key-file to enable HTTPS connections
Set --client-cert-auth=true
```

Kubernetes Security – API

Secure API

The Kubernetes API server can serve requests on two ports:

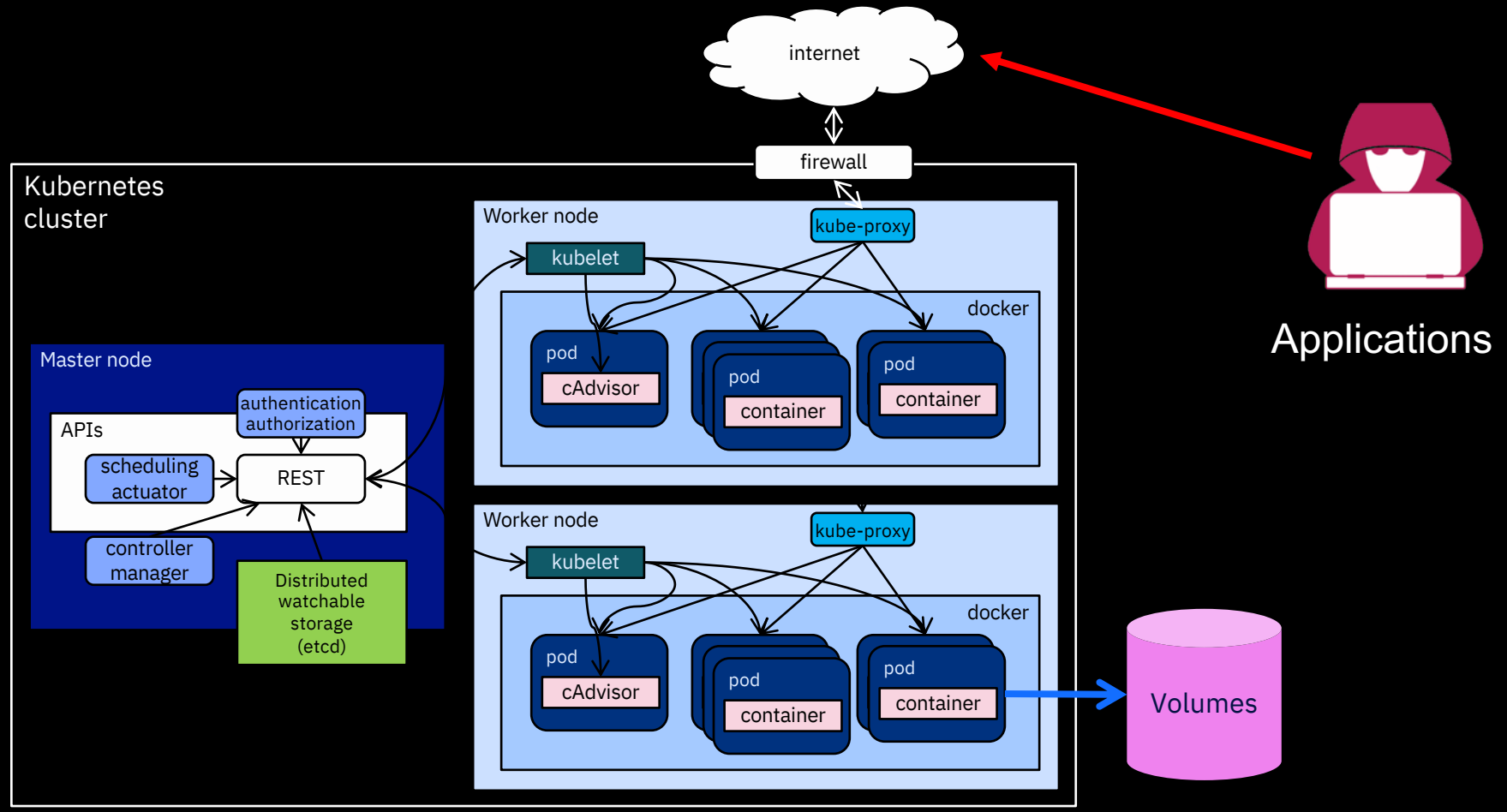
- `localhost` port (8080) and
- secure port

The `localhost` port is intended for testing purposes and for other master components to talk to the API. Requests to the Localhost port bypass authentication and authorization modules.

Best practice is to set `--insecure-port flag=0` and remove the `--insecure-bind-address` flag from the API server manifest.
And remove the `--secure-port` flag from the API server spec to ensure that all requests to the secure port are authenticated and authorized.

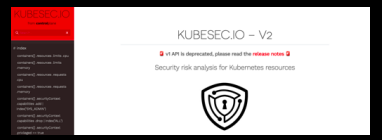
Node: Has been deprecated in Kubernetes 1.10 and should be removed in the future

Kubernetes Security – Mitigation - Containers



Kubernetes – Cluster Security – Secure Deployments

Scan for K8s Best Practices



kubesec – Validation of security best practices

Performs security risk analysis for Kubernetes resources and tells you what you should change in order to improve the security of those pods. It also gives you 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
},
```

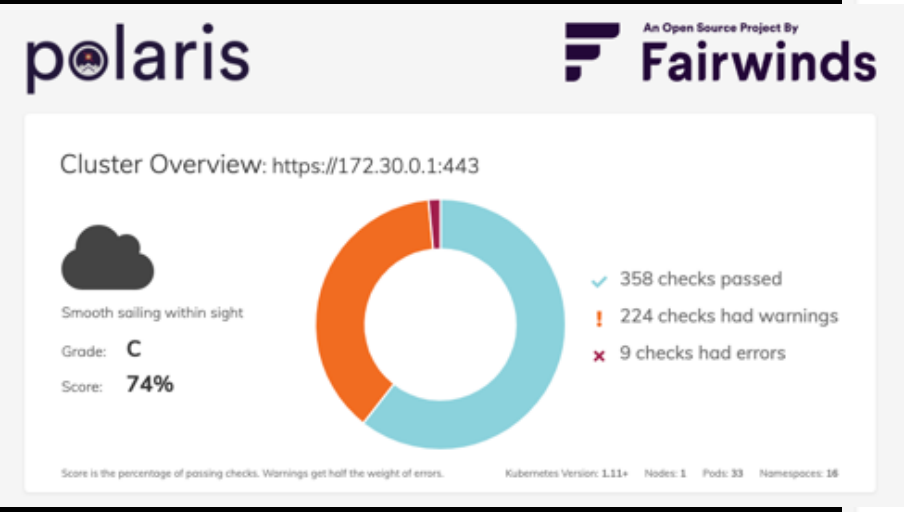

Seen in JTC16

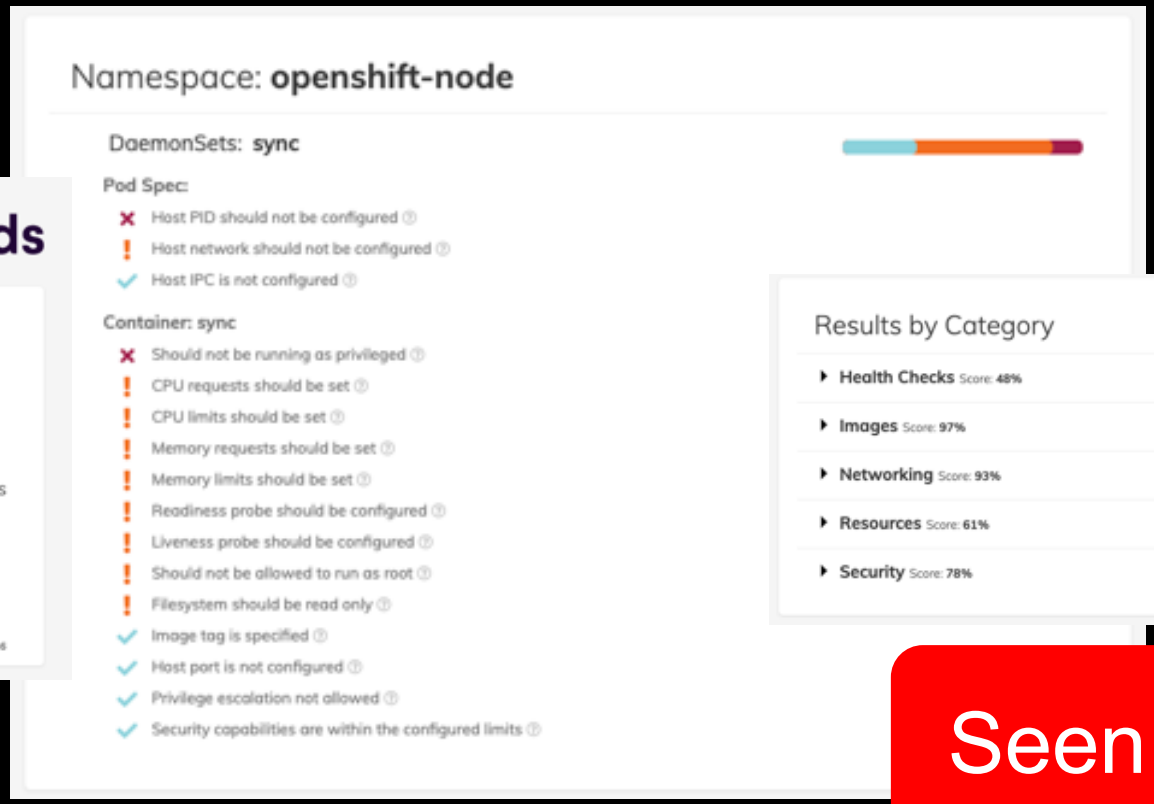
Source: https://kubesec.io/

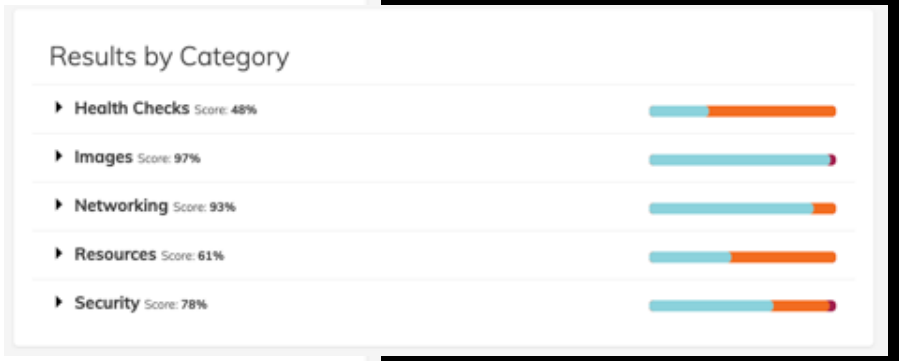
Kubernetes – Cluster Security – Secure Deployments

Scan for K8s Best Practices

Polaris – Validation of best practices







Seen in JTC16

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

Scan images for vulerabilities

One of the best-known Image scanning tools is Clair.
It is an open source project for the static analysis of vulnerabilities in application containers (currently including appc and docker).

```
clair timeout 1m0s
docker timeout: 1m0s
no whitelist file
Analysing 14 layers
Got results from Clair API v1
Found 734 vulnerabilities
Unknown: 223
Negligible: 345
Low: 164
Medium: 2

+----------+----------------+-------------+-----------------+---------+-------------------------------+-------------------------------------------------+
| SEVERITY |      NAME      | FEATURENAME | FEATUREVERSION | FIXEDBY |          DESCRIPTION          |                      LINK                       |
+----------+----------------+-------------+-----------------+---------+-------------------------------+-------------------------------------------------+
| Medium   | CVE-2009-3546  | libwmf      | 0.2.8.4-10.6    |         | The _gdGetColors function     | https://security-tracker.debian.org/tracker/CVE-2009-3546 |
|          |                |             |                 |         | in gd_gd.c in PHP 5.2.11 and  |                                                 |
|          |                |             |                 |         | 5.3.x before 5.3.1, and the   |                                                 |
|          |                |             |                 |         | GD Graphics Library 2.x, does |                                                 |
|          |                |             |                 |         | not properly verify a certain |                                                 |
|          |                |             |                 |         | colorsTotal structure member, |                                                 |
|          |                |             |                 |         | which might allow remote      |                                                 |
|          |                |             |                 |         | attackers to conduct buffer   |                                                 |
|          |                |             |                 |         | overflow or buffer over-read  |                                                 |
|          |                |             |                 |         | attacks via a crafted GD file,|                                                 |
|          |                |             |                 |         | a different vulnerability than|                                                 |
|          |                |             |                 |         | CVE-2009-3293. NOTE: some of  |                                                 |
|          |                |             |                 |         | these details are obtained    |                                                 |
|          |                |             |                 |         | from third party information. |                                                 |
+----------+----------------+-------------+-----------------+---------+-------------------------------+-------------------------------------------------+
| Medium   | CVE-2007-3996  | libwmf      | 0.2.8.4-10.6    |         | Multiple integer overflows    | https://security-tracker.debian.org/tracker/CVE-2007-3996 |
|          |                |             |                 |         | in libgd in PHP before 5.2.4  |                                                 |
|          |                |             |                 |         | allow remote attackers to     |                                                 |
|          |                |             |                 |         | cause a denial of service     |                                                 |
|          |                |             |                 |         | (application crash) and       |                                                 |
|          |                |             |                 |         | possibly execute arbitrary    |                                                 |
|          |                |             |                 |         | code via a large (1) srcW     |                                                 |
|          |                |             |                 |         | or (2) srcH value to the (a)  |                                                 |
|          |                |             |                 |         | gdImageCopyResized function,  |                                                 |
|          |                |             |                 |         | or a large (3) sy (height)    |                                                 |
|          |                |             |                 |         | or (4) sx (width) value to    |                                                 |
|          |                |             |                 |         | the (b) gdImageCreate or the  |                                                 |
|          |                |             |                 |         | (c) gdImageCreateTrueColor    |                                                 |
|          |                |             |                 |         | function.                     |                                                 |
+----------+----------------+-------------+-----------------+---------+-------------------------------+-------------------------------------------------+
```

Source: https://github.com/quay/clair

Kubernetes Security – Namespaces (from last week)

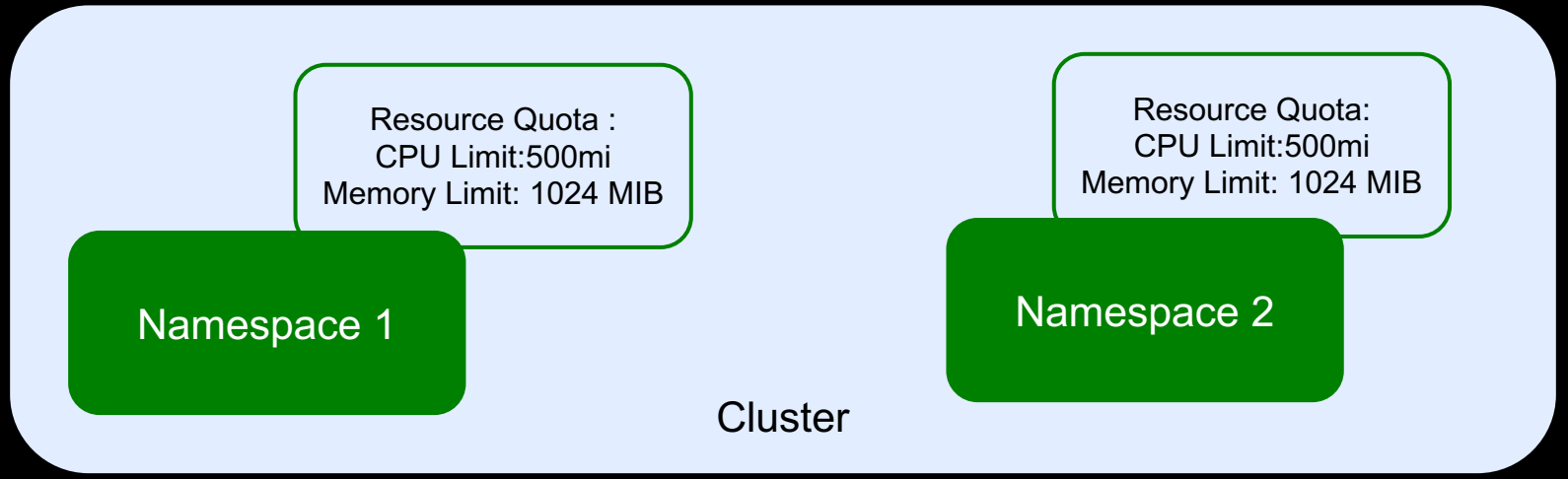
Use Namespaces Liberally

“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 – ResourceQuota

Set resource quota

Once quota in a namespace for compute resources set, the users are forced to set requests or limits for those values. Avoids starving of Cluster.



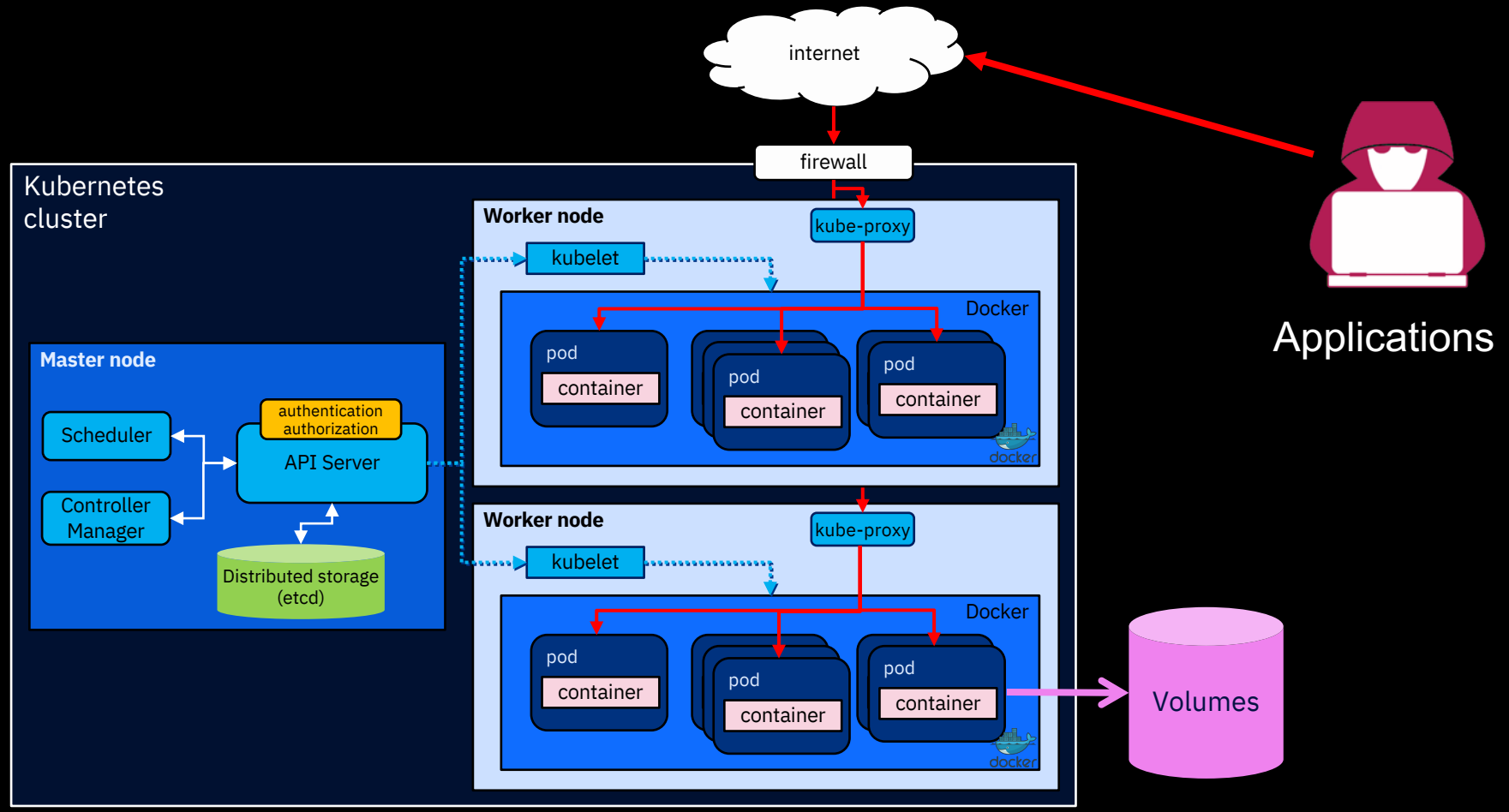
Kubernetes Security – Images

Scan your containers for vulnerabilities

Known vulnerabilities account for a large portion of breaches. Use a tool to scan your containers for them and then mitigate them. Anchore, Clair, and Quay are among my favorite tools, but there are others out there.

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

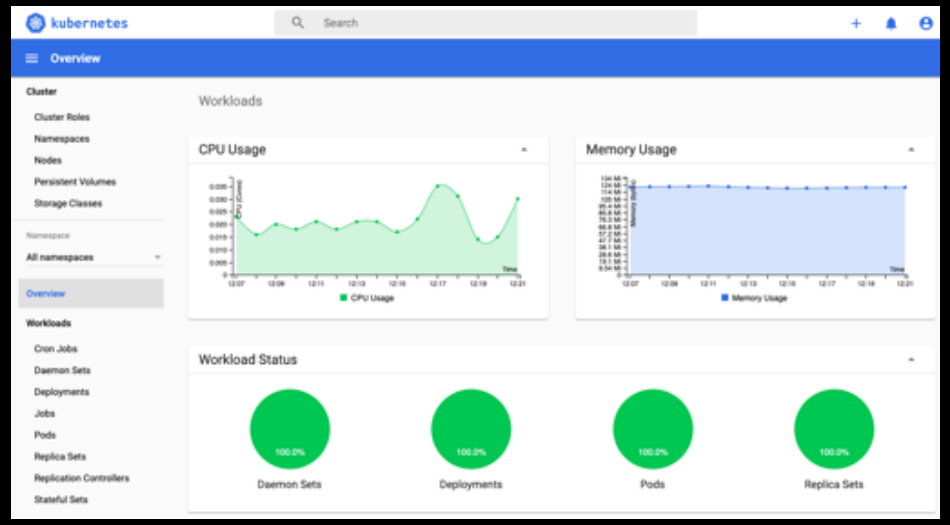
Kubernetes Security – Mitigation – Network



Kubernetes Security – Kubernetes Dashboard

Secure Kubernetes Dashboard

The Kubernetes Dashboard has often been used for attacks.
Easy to deploy with high privileges.



Best practice is to:

- Allow only authenticated access
- Use RBAC
- Ensure Dashboard service account has limited access
- Don’t expose to the internet (use kubectl proxy for local access)

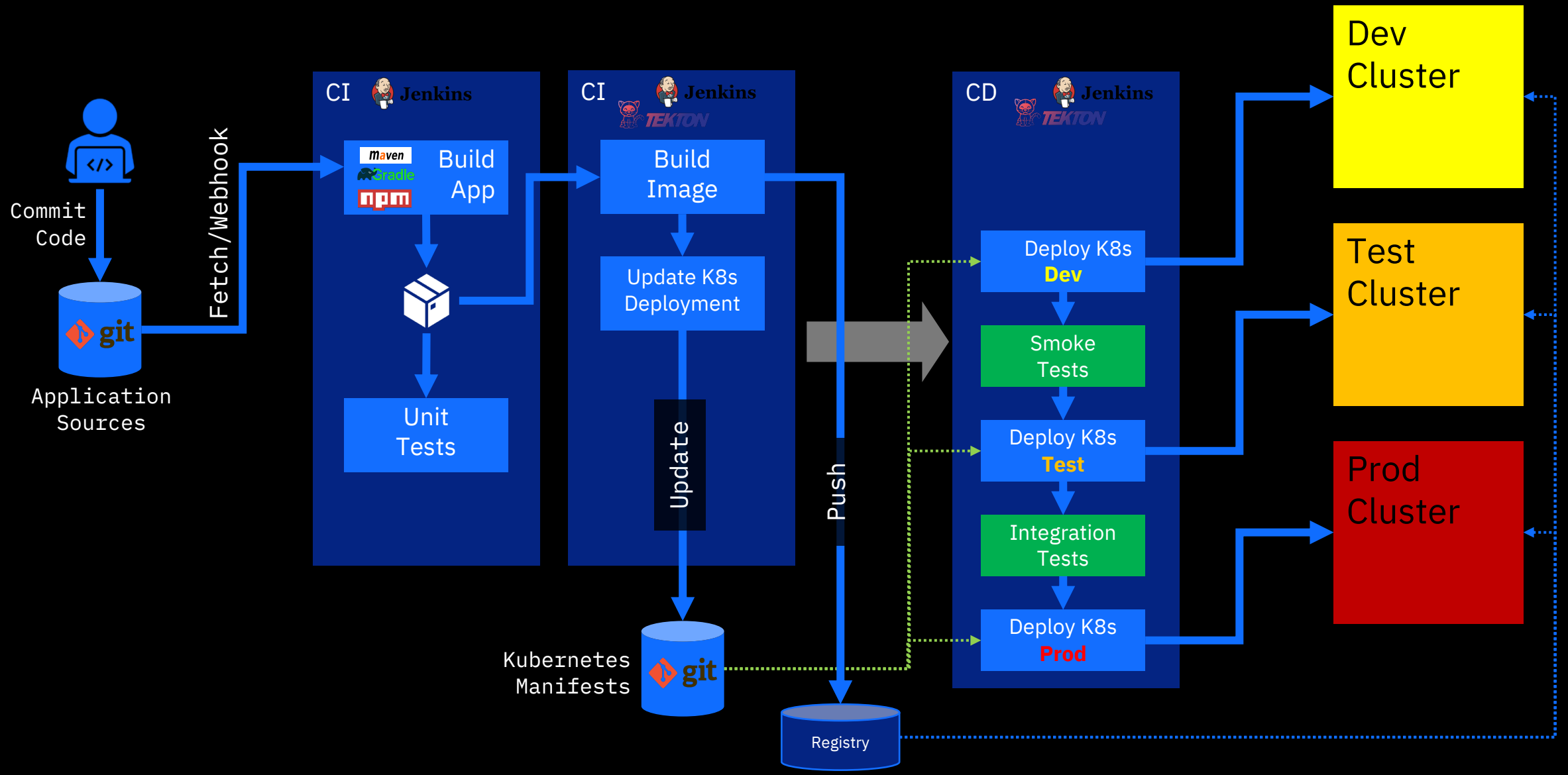
QUESTIONS?

Kubernetes Workshop Series

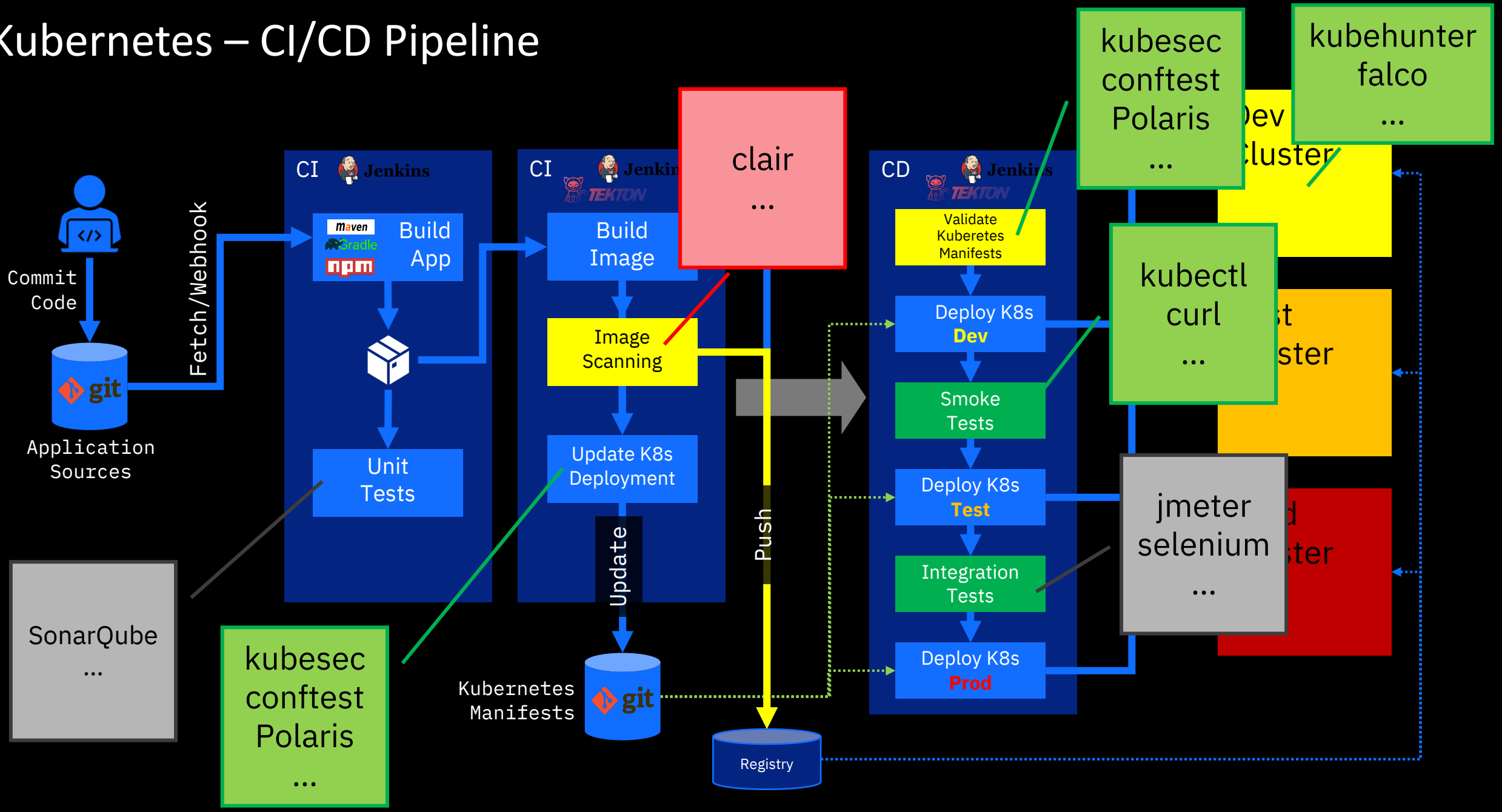
Security Checklist

03

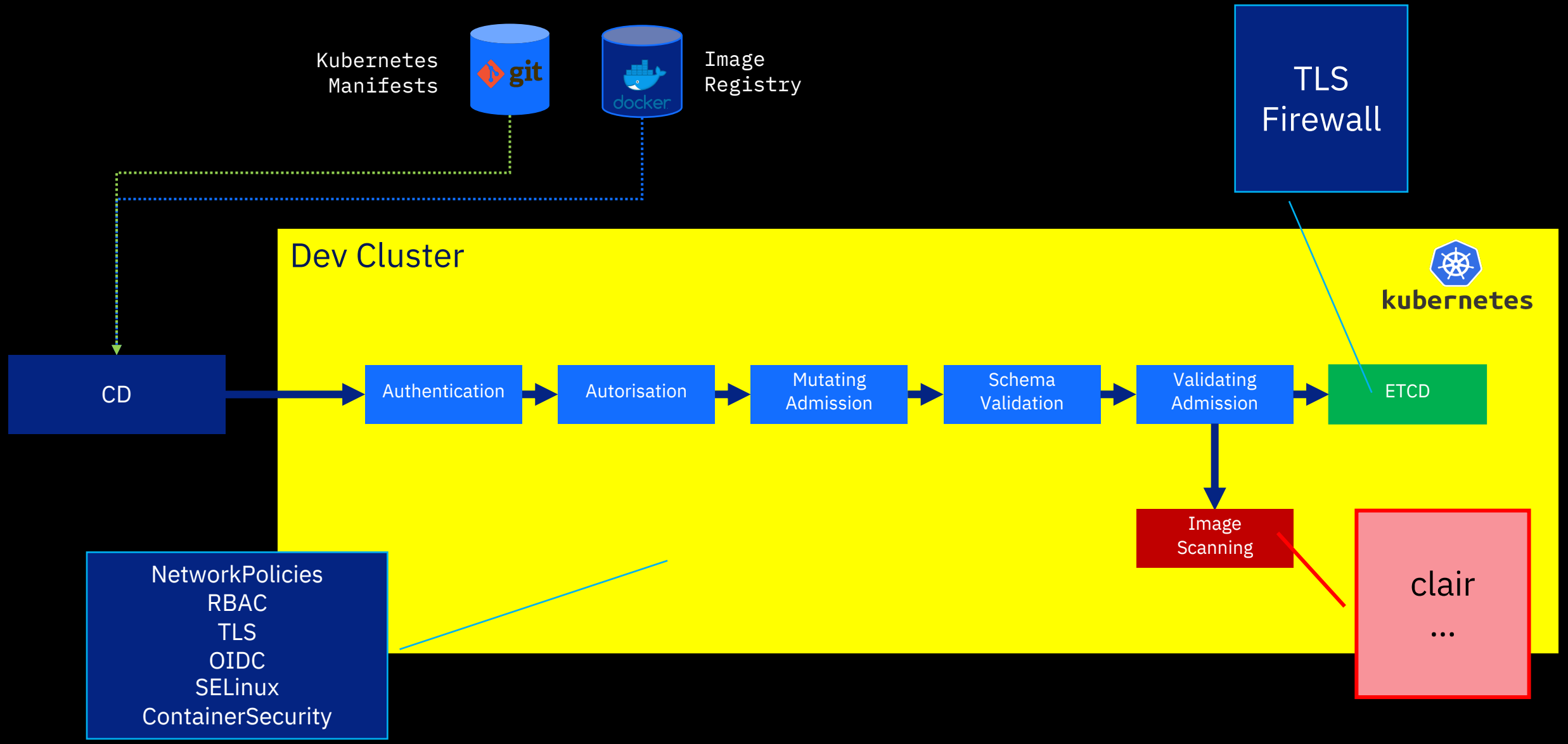
Kubernetes – CI/CD Pipeline - GitOps



Kubernetes – CI/CD Pipeline



Kubernetes – Secure CI/CD Pipeline



Kubernetes – Self Check

1. **Where** are images used in containers **coming from**?
2. What container **services are exposed** outside of the Kubernetes cluster?
3. Can we tell which **processes** are **running** in any container in any cluster?
4. Which **network communication** pathways are **active** but are not being used in production?
5. What **team** in the organization **owns** a particular **running application**?
6. How many of my clusters, namespaces, and nodes must **adhere to to specific regulations**.
7. Which deployments are using **privileged containers**, meaning they have full access to the host operating system?

Seen in JTC16

Source: https://www.stackrox.com/post/2019/07/kubernetes-security-101/

Kubernetes – Self Check

8. How long ago were the images **scanned for vulnerabilities**?
9. Which of your containers are impacted by **known vulnerabilities**, and what's their severity?
10. Are any of these containers in **production** impacted by a known **vulnerability**?
11. Which vulnerable running containers or deployments should be **prioritized for remediation**?

Source: https://www.stackrox.com/post/2019/07/kubernetes-security-101/

QUESTIONS?

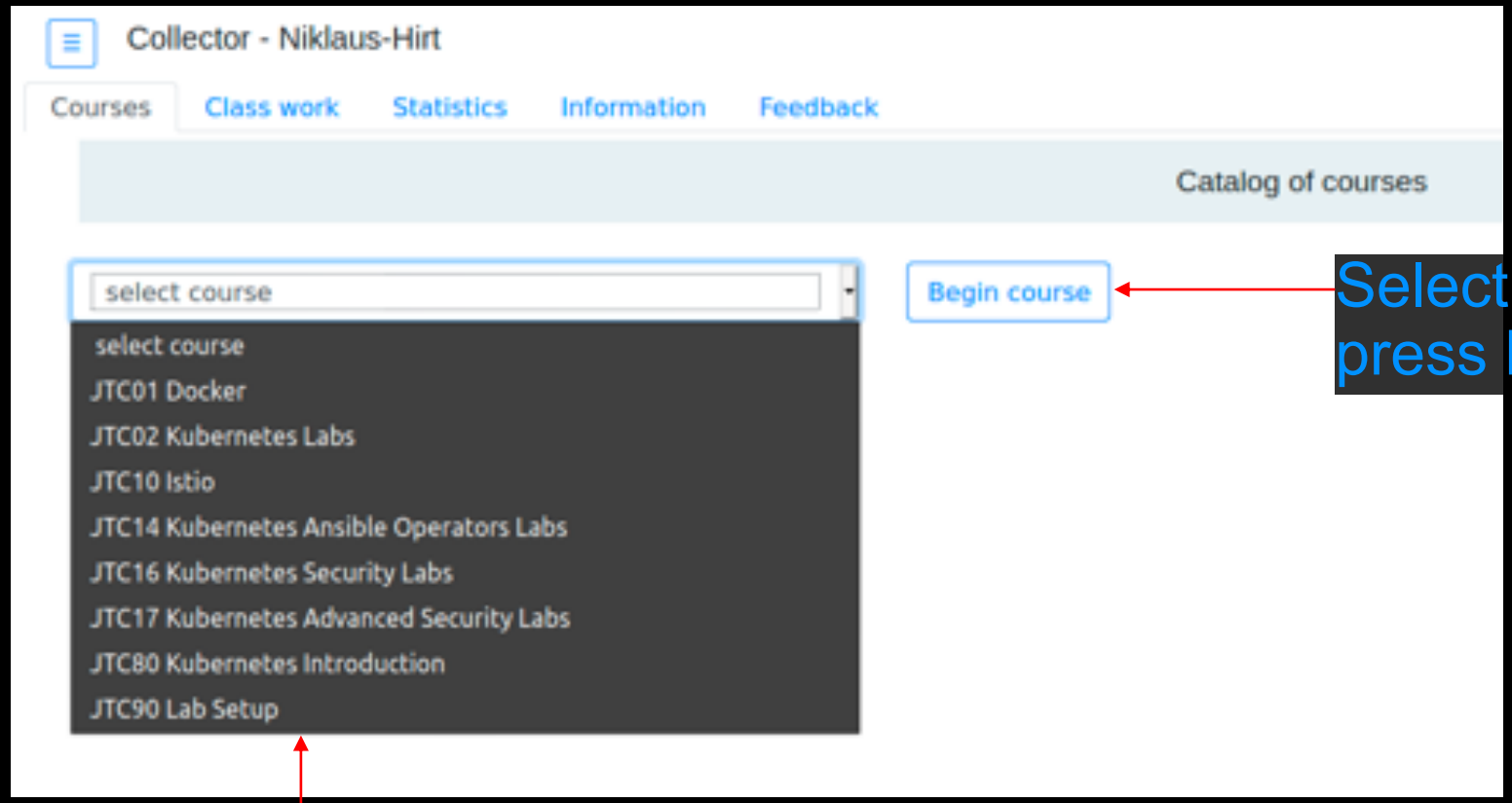
Kubernetes Workshop Series

K8s Advanced Security - *Hands-On*

04

Starting Course JTC17 K8s Advanced Security

Name will be shown



Select course and press button to begin

Current course catalog

JTC17 Kubernetes Advanced Security Labs

Lab 1 : Role Based Acces Control (RBAC)

Lab 2 : Service Accounts

Lab 3 : Security Tooling

Lab 4 : Image scanning

Kubernetes Security – Some tools I use

RBAC

RBAC-lookup - https://github.com/FairwindsOps/rbac-lookup
rakkess - https://github.com/corneliusweig/rakkess
rbac-view – https://github.com/jasonrichardsmith/rbac-view
rback - https://github.com/team-soteria/rback

Scanning

Polaris - https://github.com/FairwindsOps/polaris
Clair - https://github.com/coreos/clair

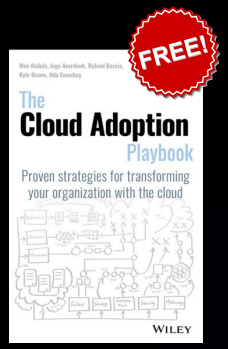
Management

K9s - https://github.com/derailed/k9s

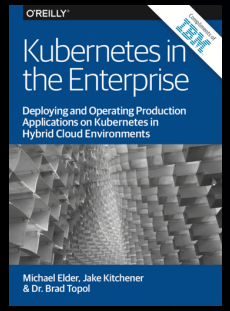
Kubernetes Security – Some Reading Tips

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

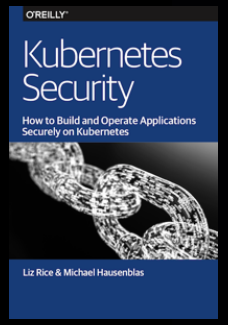
Kubernetes – Some Reading Tips



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 Operators

READY SET GO!!!!

Duration: 60 mins

QUESTIONS?

Niklaus Hirt

nikh@ch.ibm.com

@nhirt

THANK YOU!!!!