The Journey to Cloud

JTC16

Kubernetes Security

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

IBM

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 Advanced Concepts

Module – ADVANCED II

Module 0: **Prepare the Labs**

Module 1: **Security Introduction**

Module 2: **Security Elements**

BREAK

Module 3: **Wrap-up and Tips**

Module 4: **Hands-On Lab**

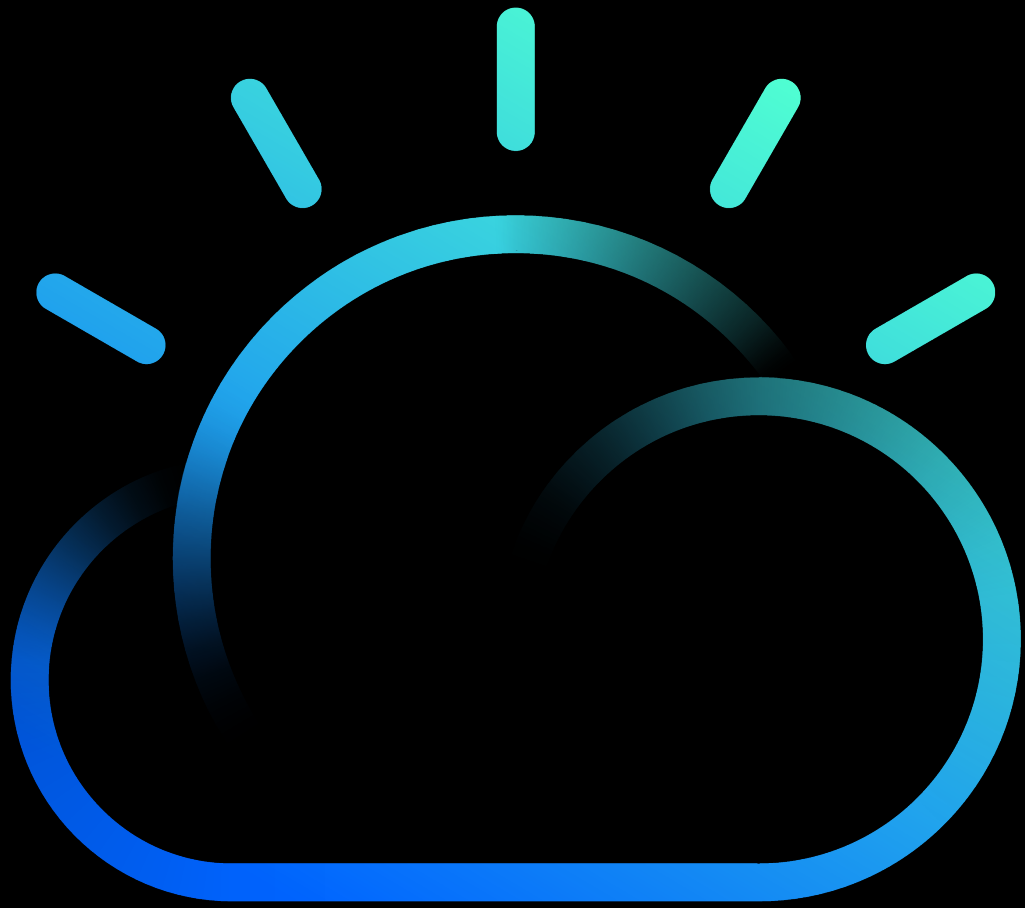
Sources and documentation will be available here:

https://github.com/niklaushirt/k8s_training_public

https://github.com/niklaushirt/training

The Journey to Cloud
Prepare the Labs

00

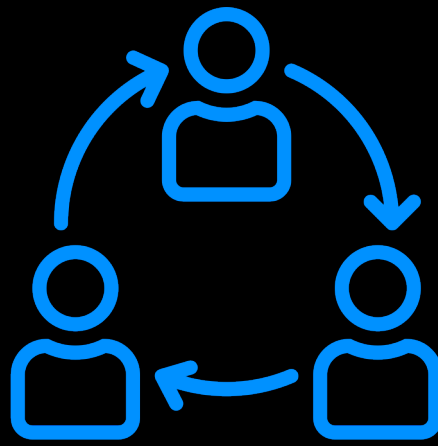


IBM **Cloud**

Session Objectives

Attendees will be grouped in ***teams*** wherever it makes sense to facilitate collaborative work.

Following some lectures will be ***hands-on*** work that each team collaborates to complete.



Teams

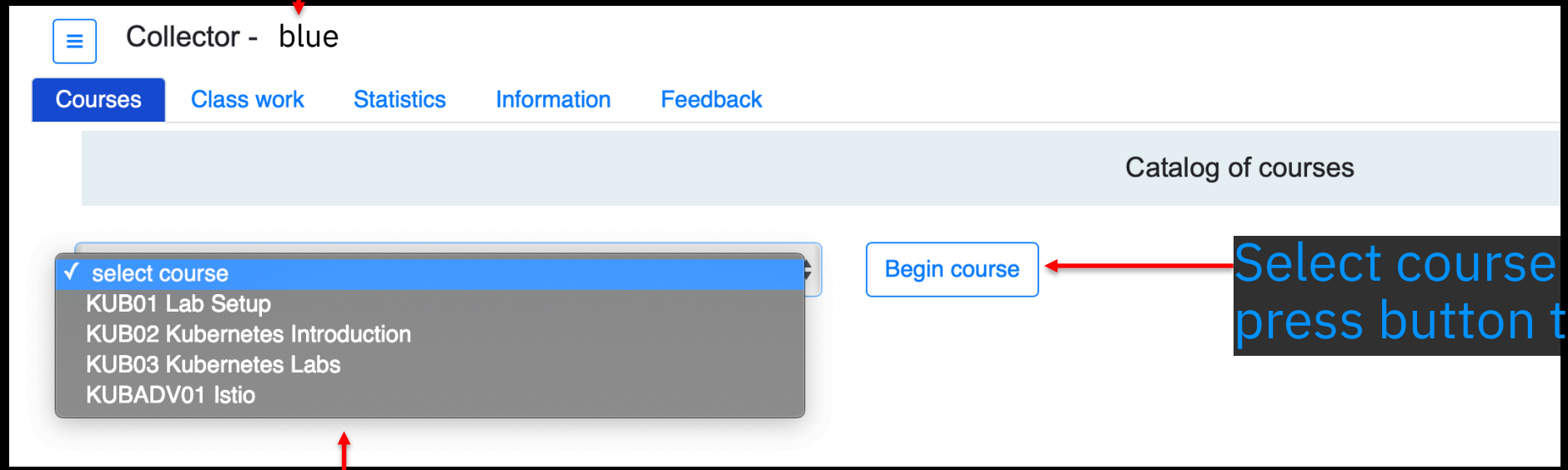
black 31701	olive 31711	peru 31715
white 31702	brown 31712	chocolate 31716
red 31703	lightblue 31713	orchid 31717
blue 31704	orange 31708	gold 31718
yellow 31705	purple 31709	pink 31719
lime 31706	maroon 31710	violet 31720
cyan 31707	firebrick 31714	

Collector - Accessing team web site

http://158.177.137.195:{port#}

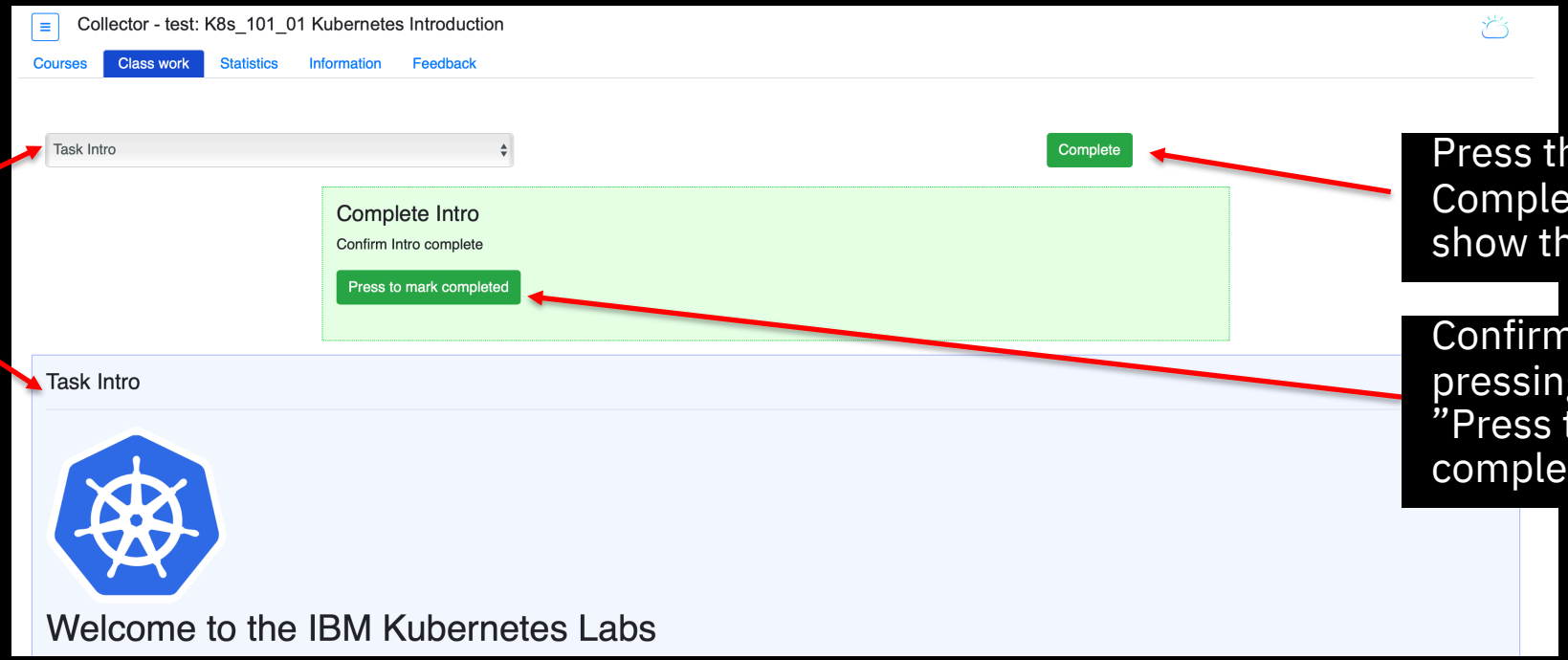
blue 31704

Team name / color will be shown



Collector – Class work

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



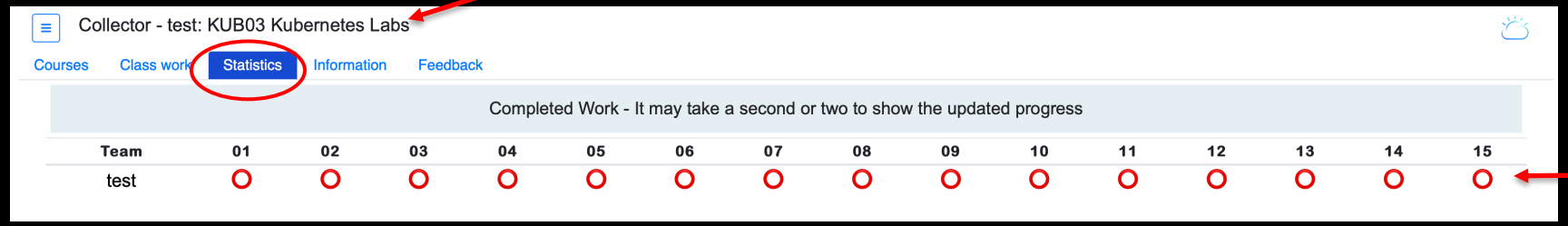
Press the green Complete button to show the green portion.

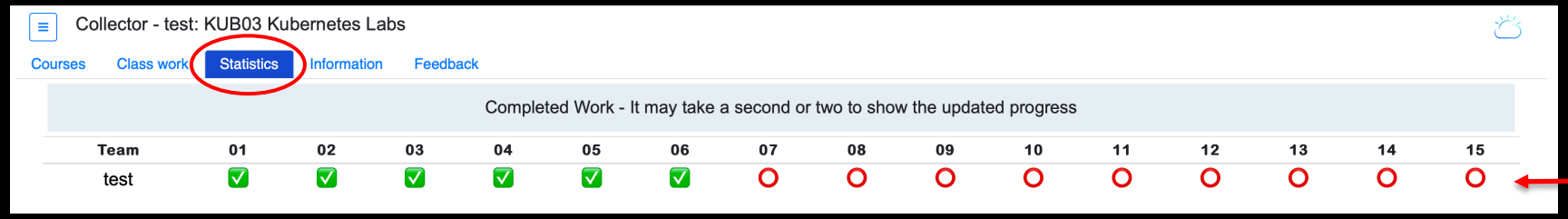
Confirm completion by pressing the green ”Press to mark completed” button.

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

Collector – Track course completed work

Course title



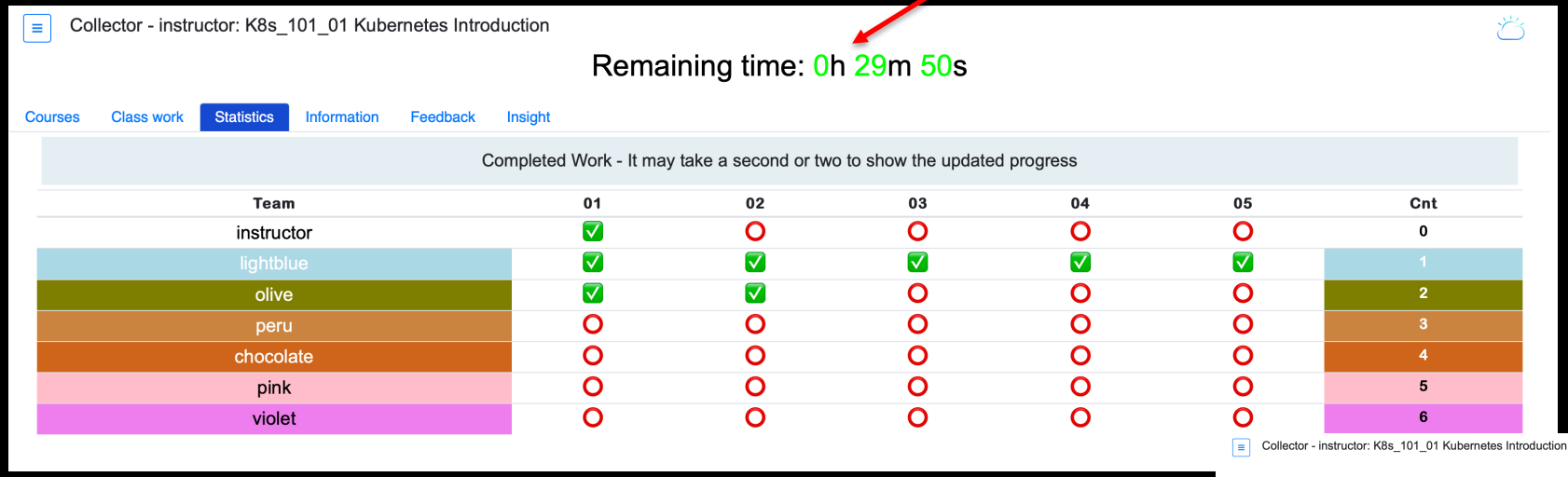


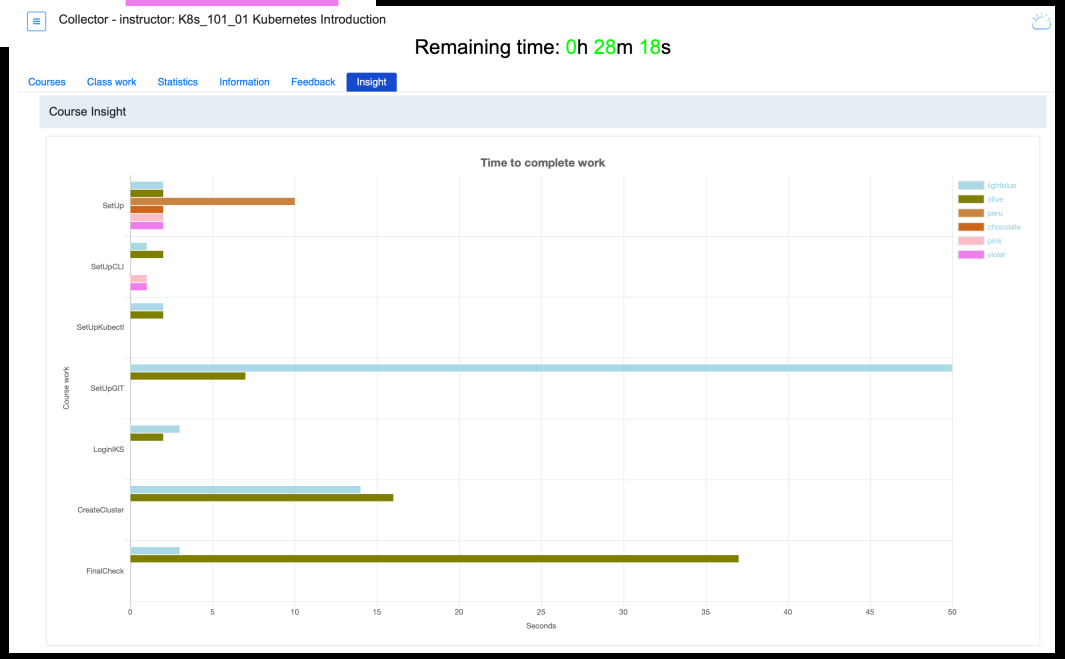
The number of items tracked will change based on the current course selected.

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

Collector – Instructor Dashboard

Remaining Time for the Lab





JTC90 Lab Setup

Task 1: Download Training VM

Task 2: Setup VirtualBox >6.14

Task 3: Start Training VM

Task 4: Login / Check

JTC90 Lab Setup

Task 1: Download Training VM

Task 2: Setup VirtualBox >6.14

Task 3: Start Training VM

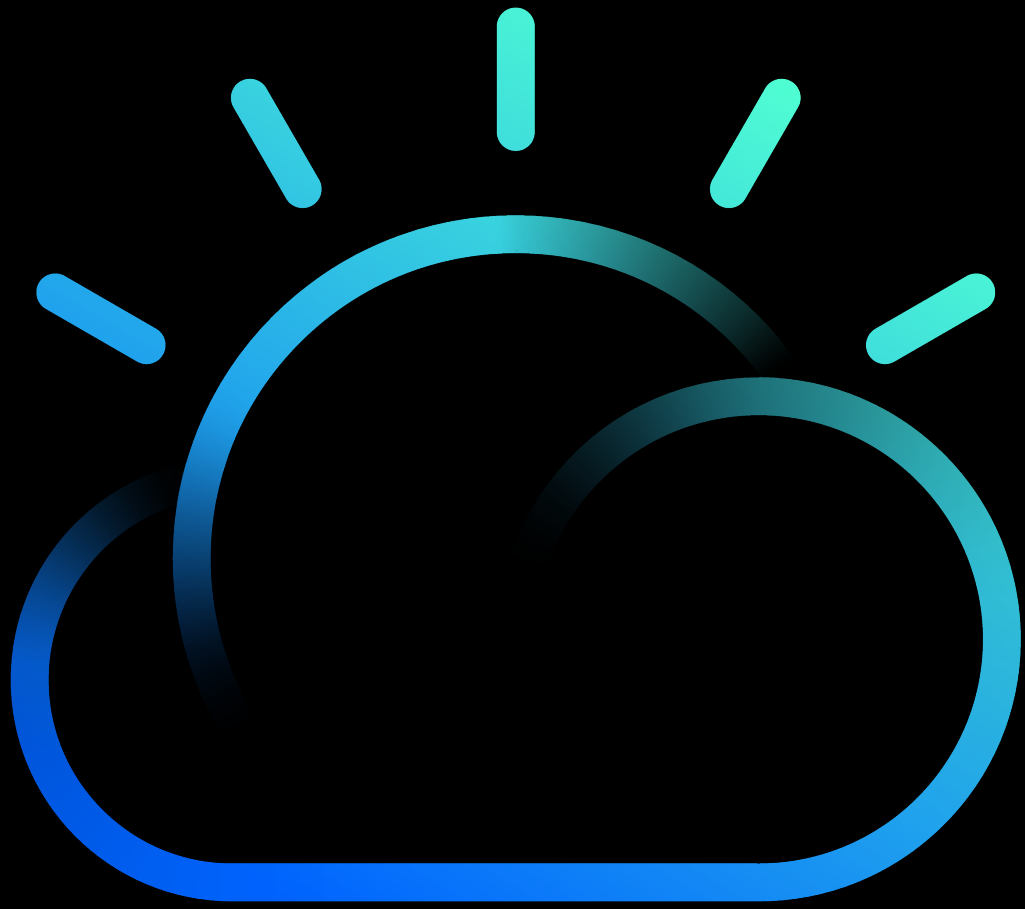
Task 4: Login / Check

EVERYBODY OK ?

The Journey to Cloud

Security Introduction

01



IBM Cloud

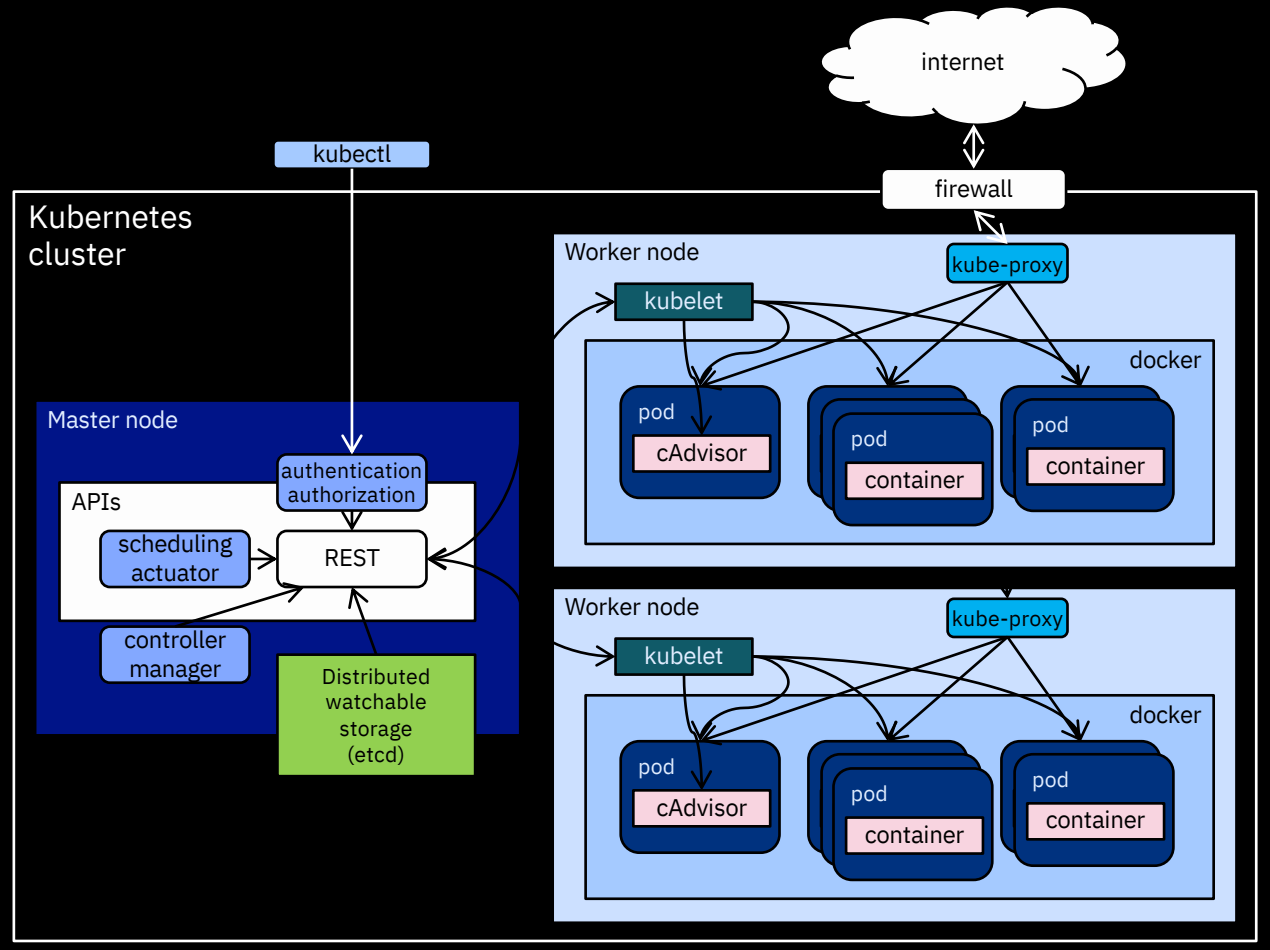
Kubernetes – **Security – Back to basics**

Master node

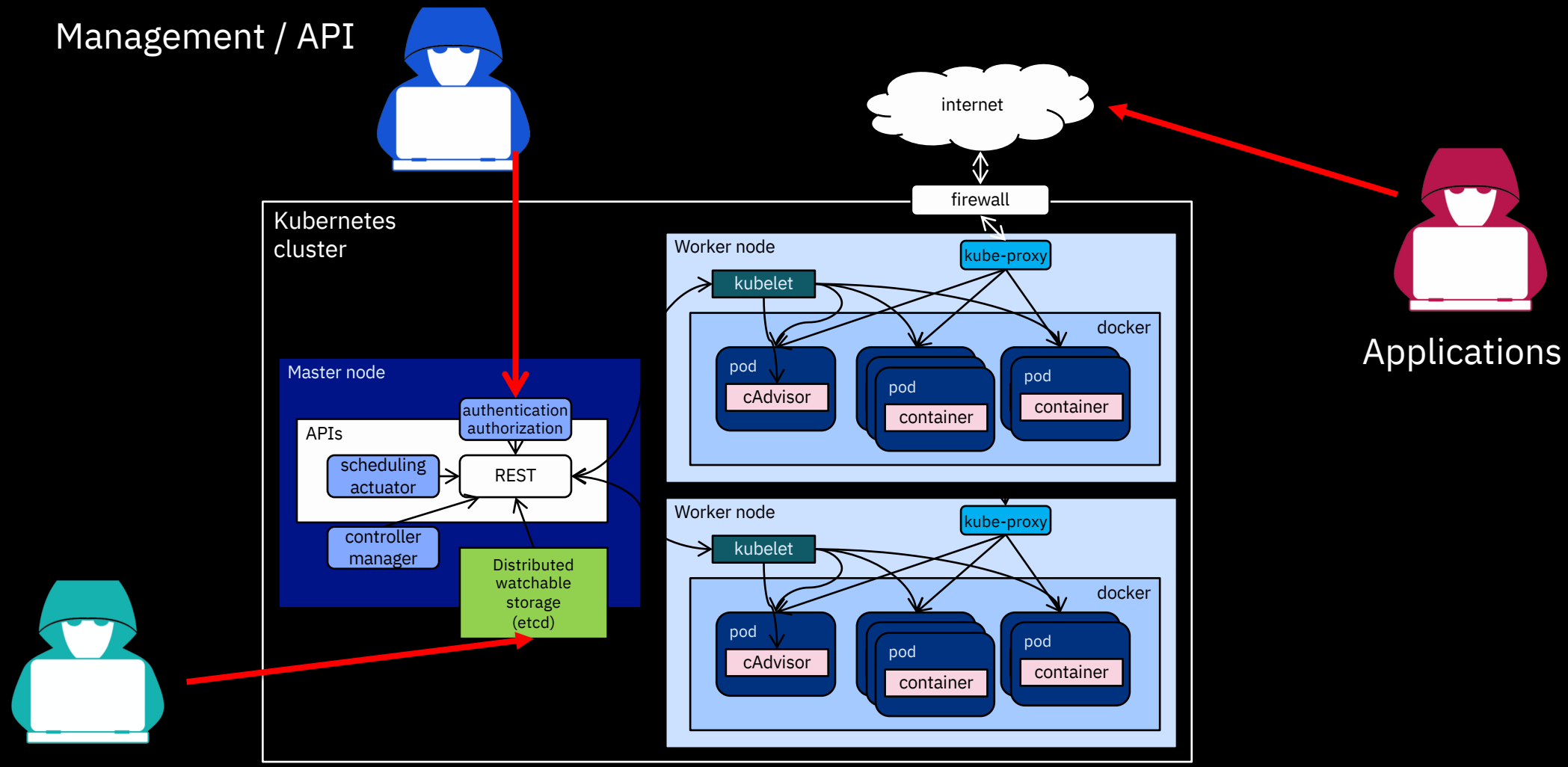
- Node that manages the cluster
- Scheduling, replication & control
- Multiple nodes for HA

Worker nodes

- Node where pods are run
- Docker engine
- kubelet agent accepts & executes commands from the master to manage pods
- cAdvisor – Container Advisor provides resource usage and performance statistics
- kube-proxy – routes inbound or ingress traffic



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**

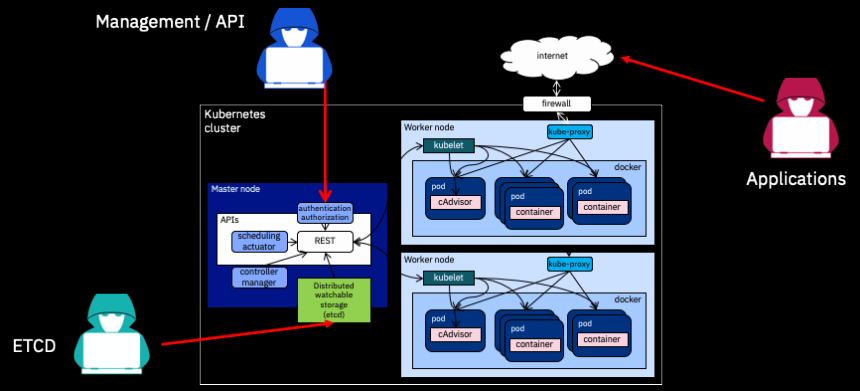
Reduce Kubernetes Attack Surfaces

etcd

Management/API

Applications

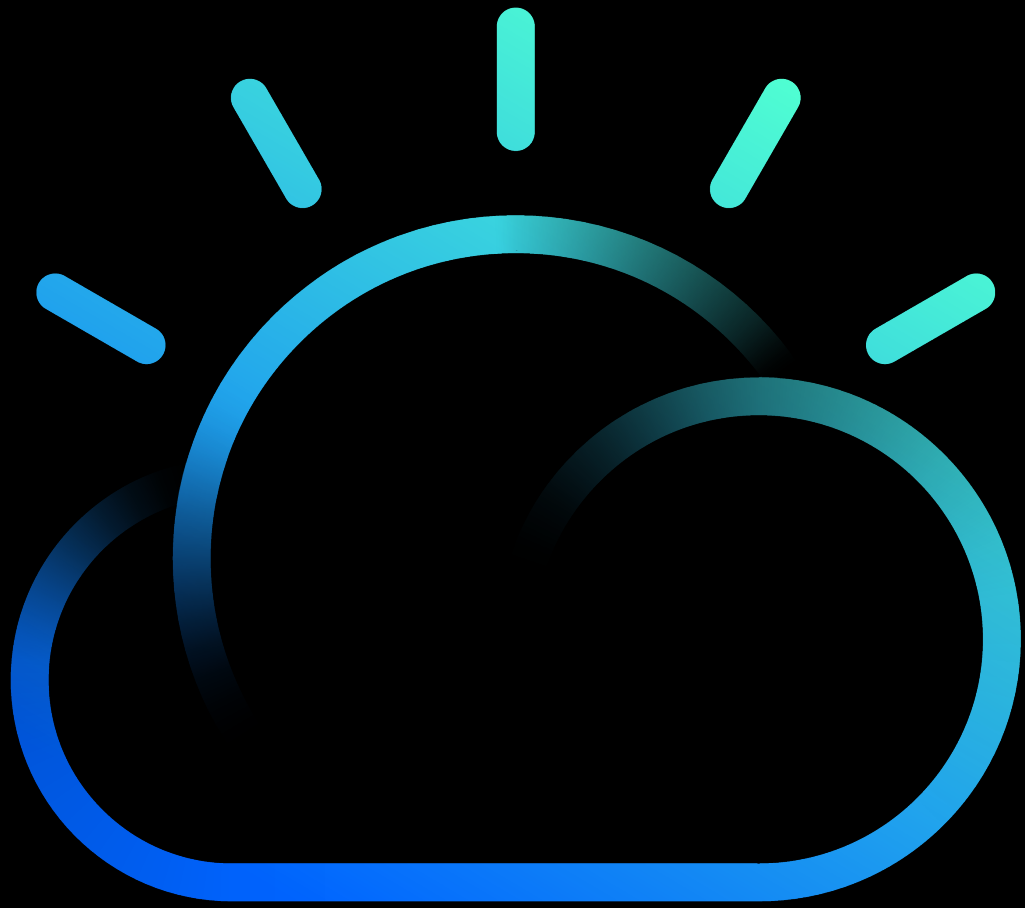
- Restrict access to etcd
- Controlling access to the Kubernetes API
- Controlling access to the Kubelet
- Enforce resource usage limits for workloads
- Controlling what privileges containers run with
- Enable audit logging
- Enforcing Network Policies
- Rotate infrastructure credentials frequently
-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



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

IBM

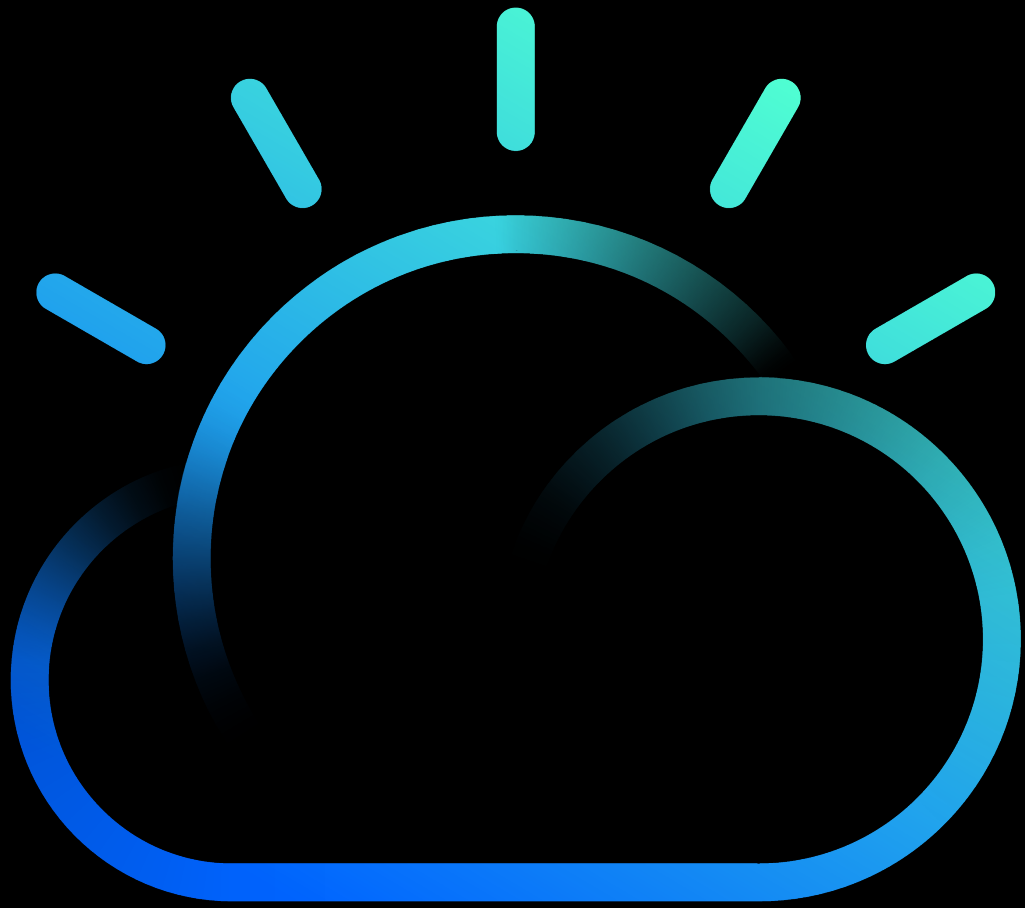
QUESTIONS?



IBM **Cloud**

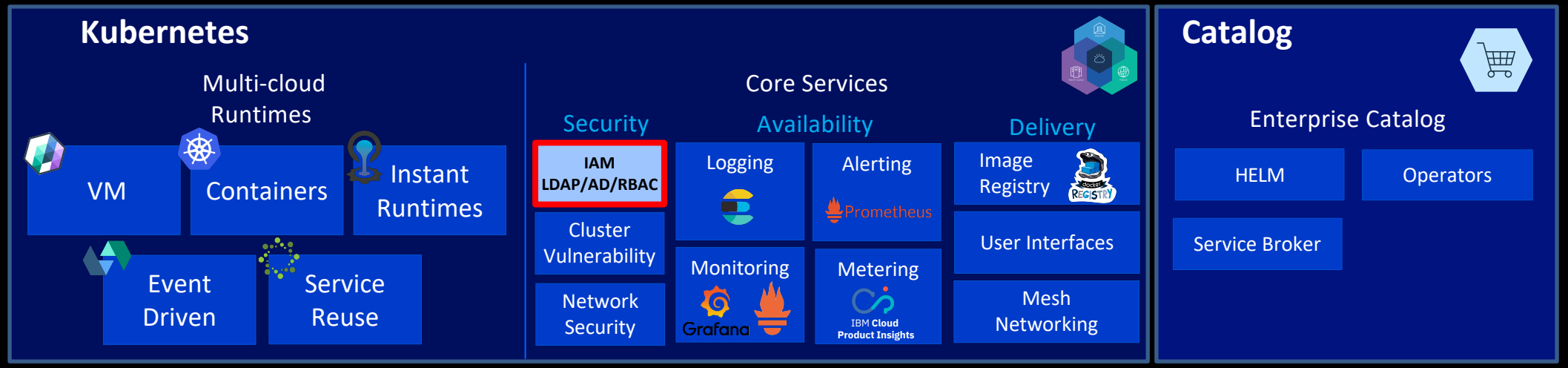
The Journey to Cloud
Security Elements

02



IBM **Cloud**

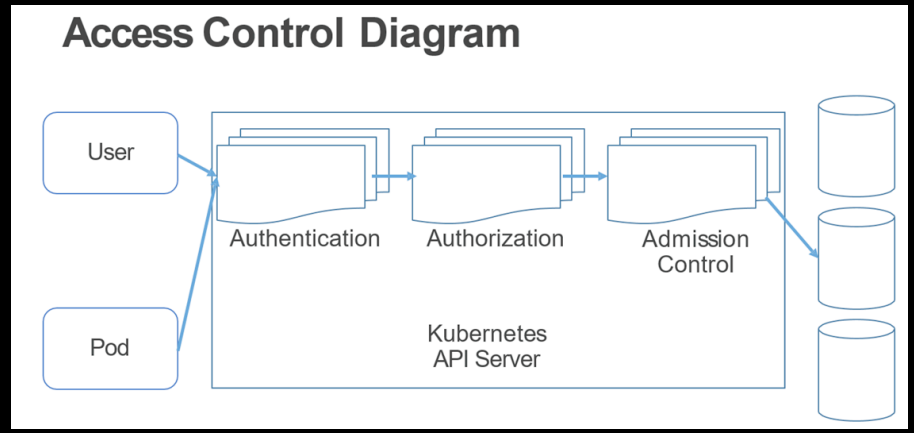
Kubernetes – Core Services - Controlling K8s Access



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

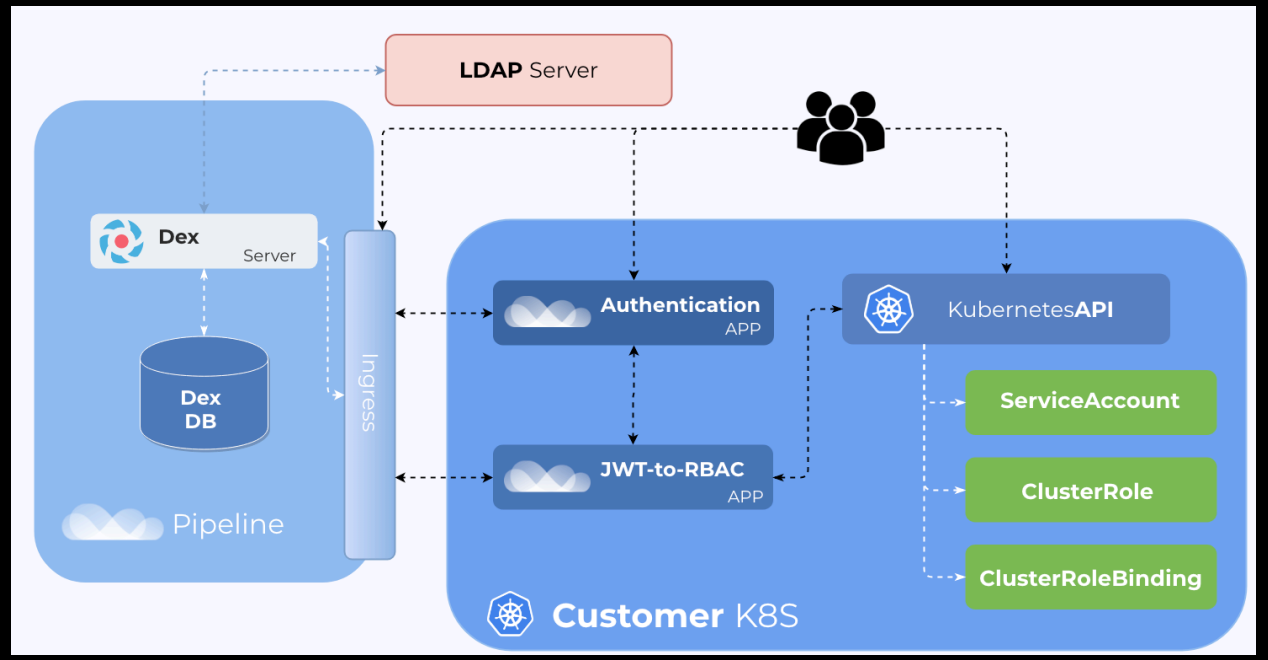


Source: https://kubernetes.io/docs/reference/access-authn-authz/controlling-access/

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.

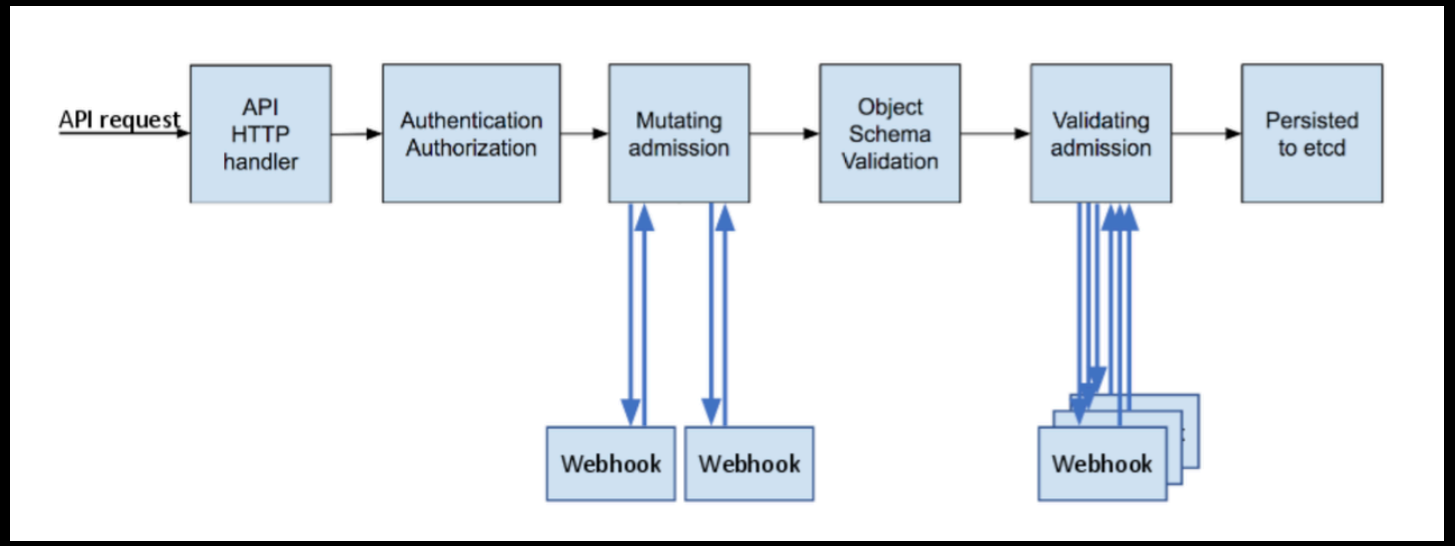


Source: https://github.com/krishnapmv/k8s-ldap and https://github.com/dexidp/dex

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.



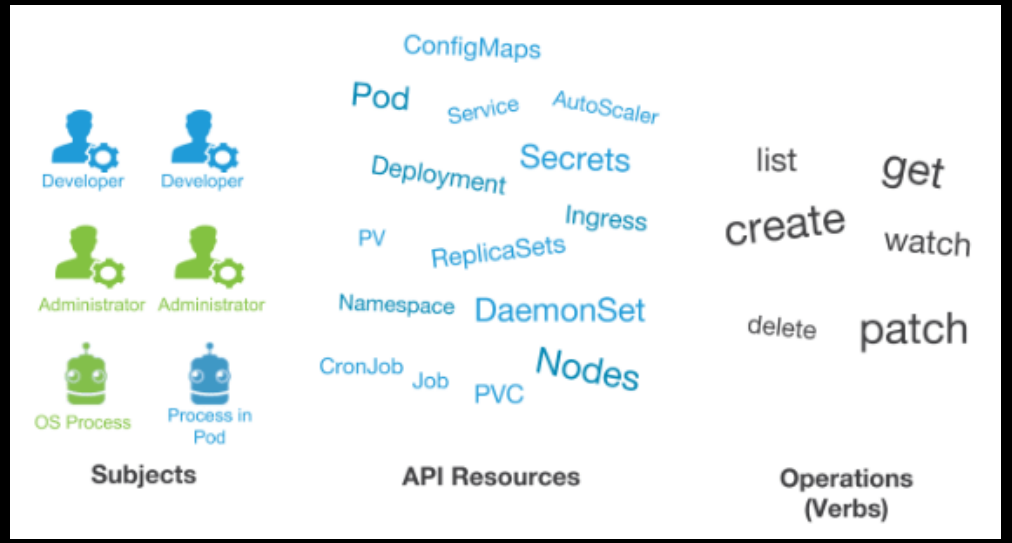
Source: https://github.com/krishnapmv/k8s-ldap and https://github.com/dexidp/dex

RBAC Basic Elements

RBAC Building Blocks

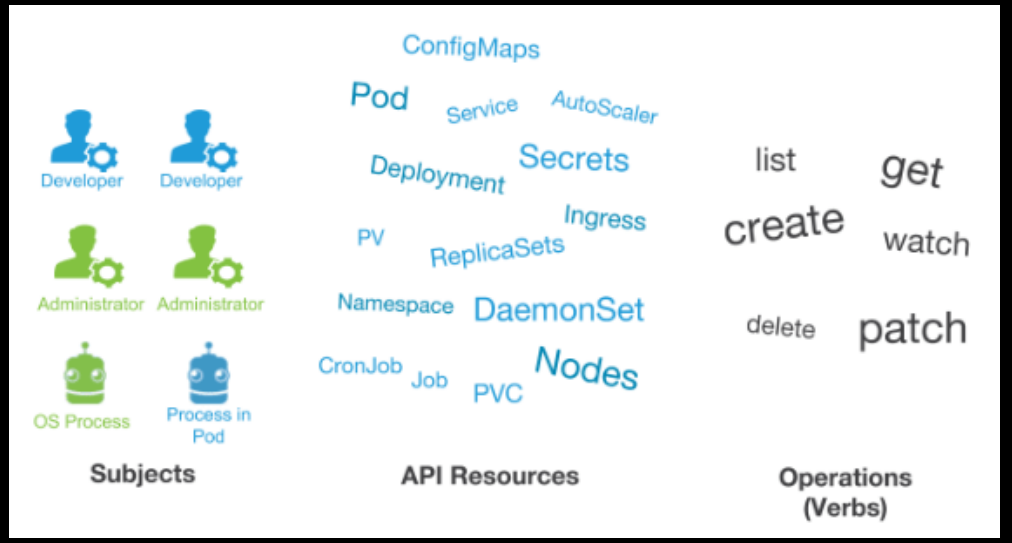
Objects
- Pods
- PersistentVolumes
- ConfigMaps
- Deployments
- Nodes
- Secrets
- Namespaces

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



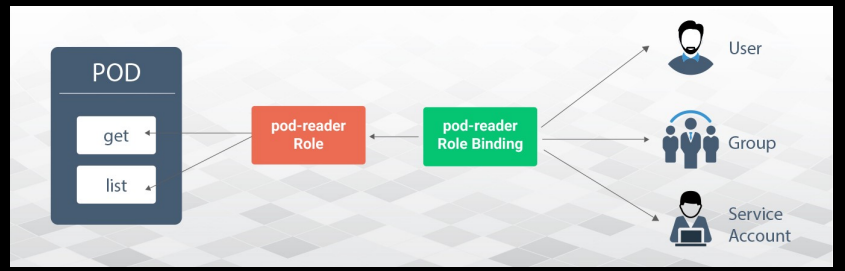
Source: https://kubernetes.io/docs/reference/access-authn-authz/rbac/

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

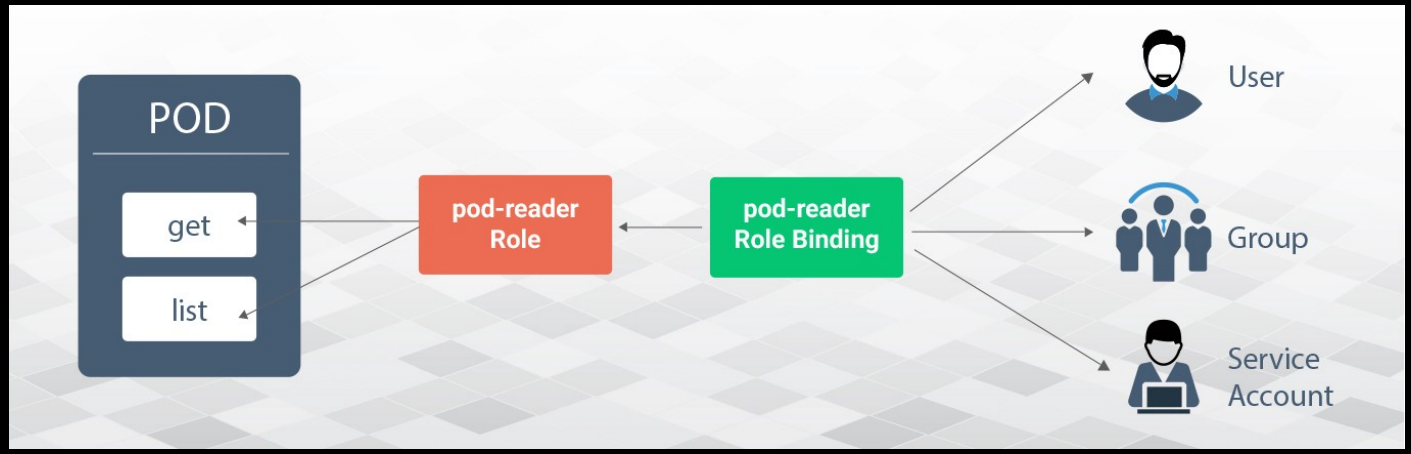


Subjects
Entity that attempts an operation in the cluster
- User Accounts: Humans or processes living outside the cluster.
- Service Accounts: Namespaced account.
- Groups: Reference multiple accounts. (default groups like cluster-admin)

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.

- **Improper Use of Role Aggregation**
  Role Aggregation can be used to combine existing roles.
  Review carefully or avoid altogether.

- **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                    ✔     ✔       ✔       ✔
```

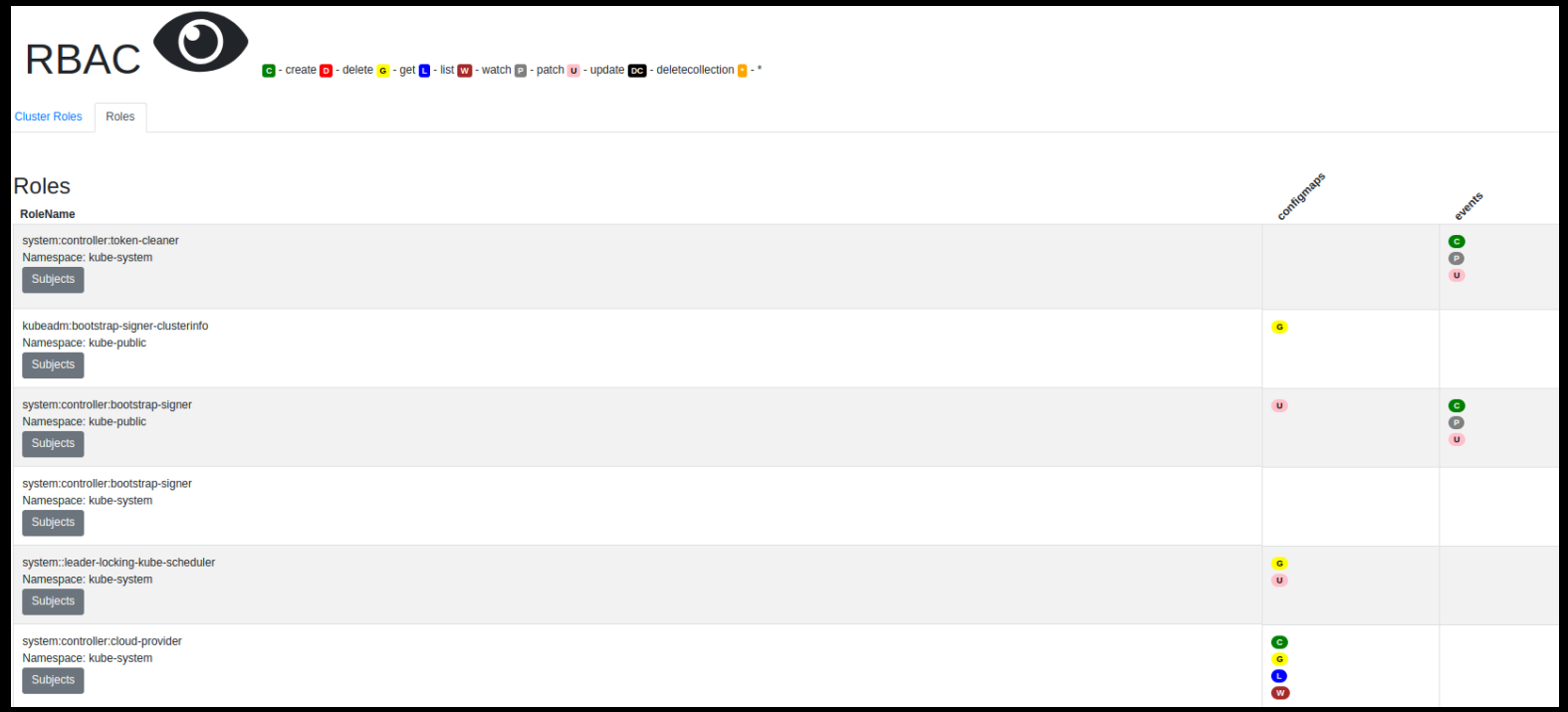
Example Usage

- 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

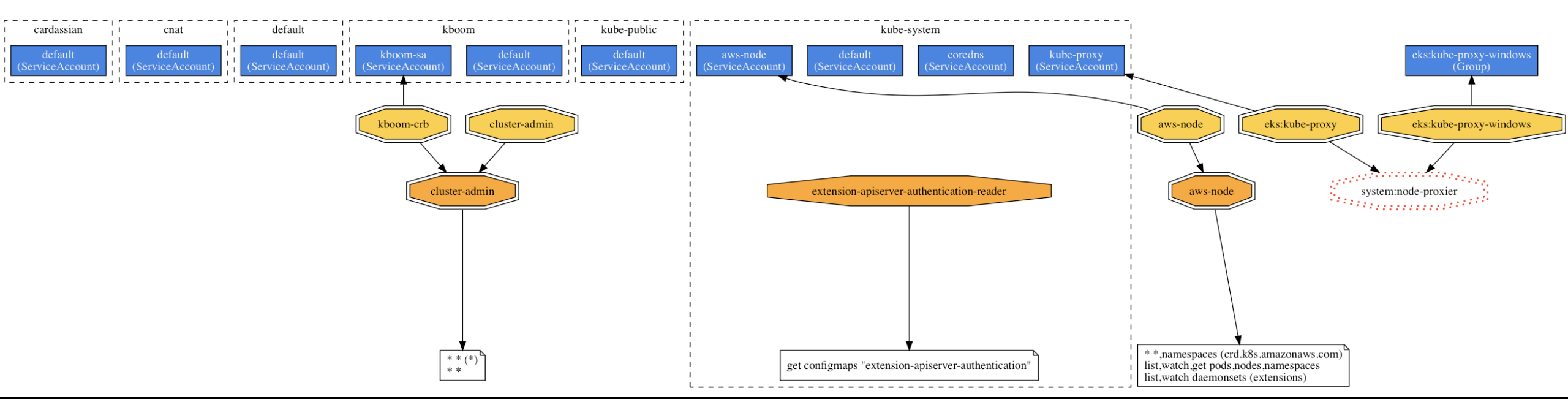
./rbac-view --render html (default)

./rbac-view --render json

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

Audit RBAC – rback

Some tools to help you see more clearly



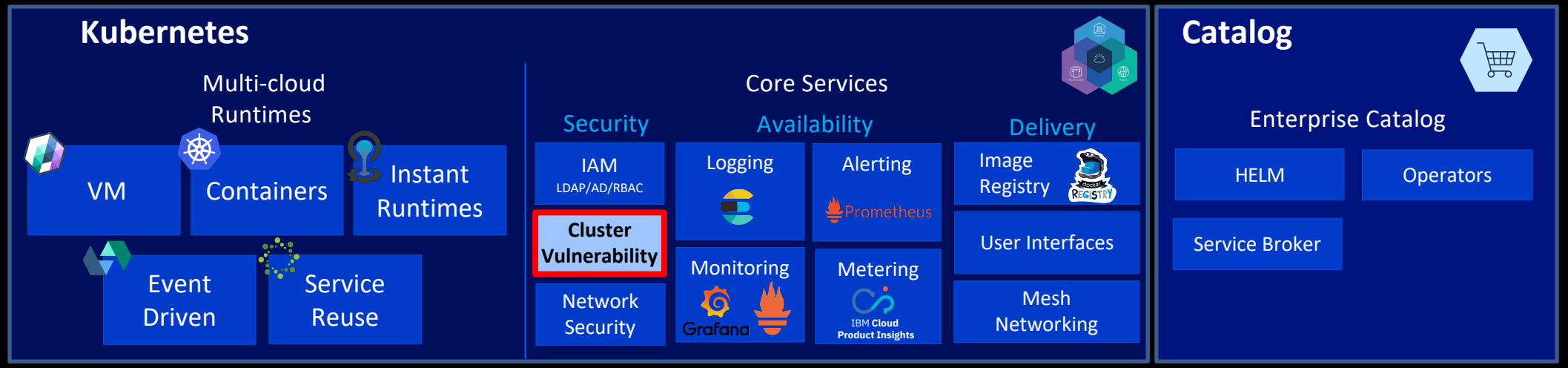
Example Usage

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

./rbac-view --render json

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

Kubernetes – Core Services – Cluster Security



Kubernetes – Cluster Security – Secure Images

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.                    |                                                     |
+----------+---------------+-------------+----------------+---------+------------------------------+-----------------------------------------------------+
```

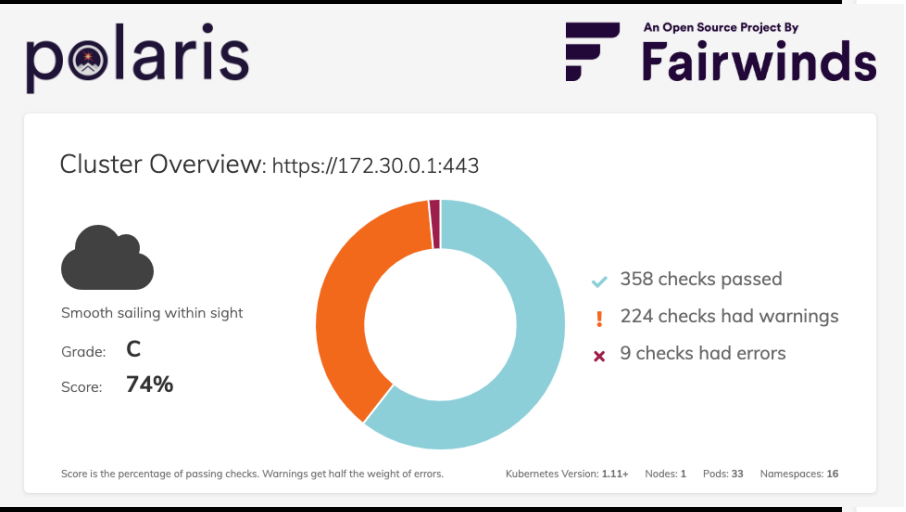
Source: https://github.com/quay/clair

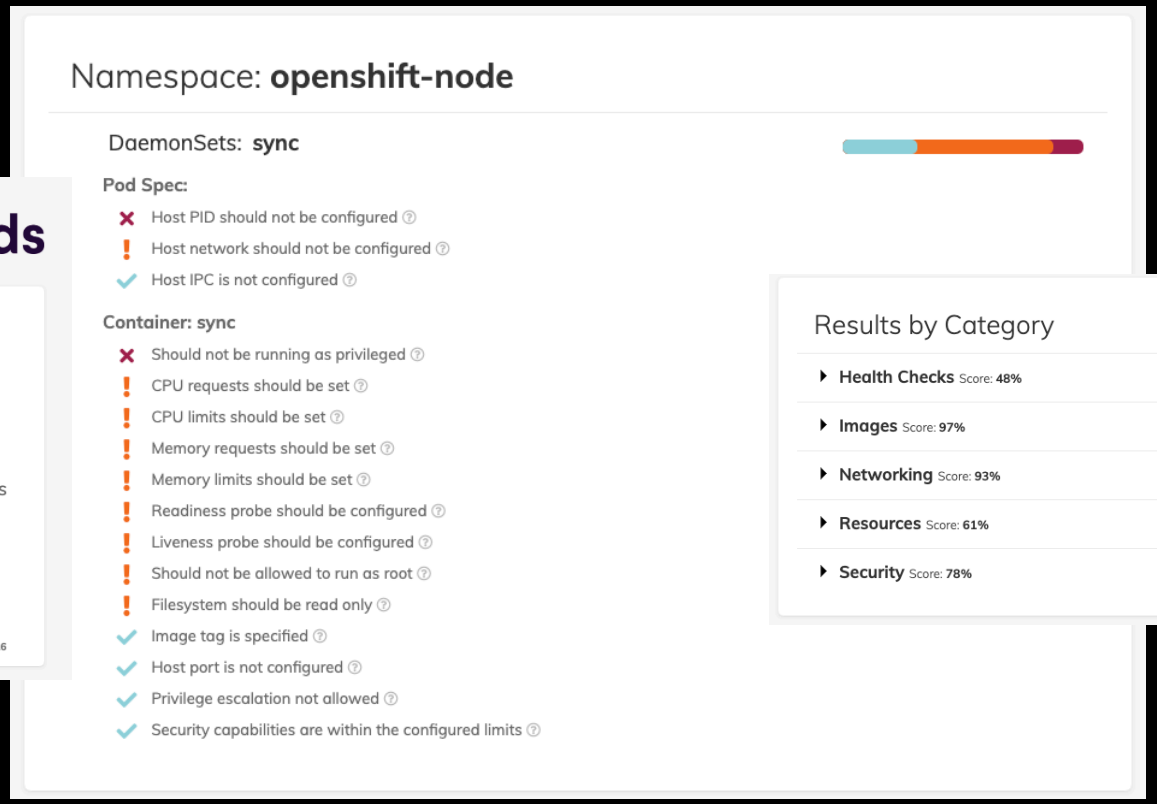
Kubernetes – Cluster Security – Secure Images

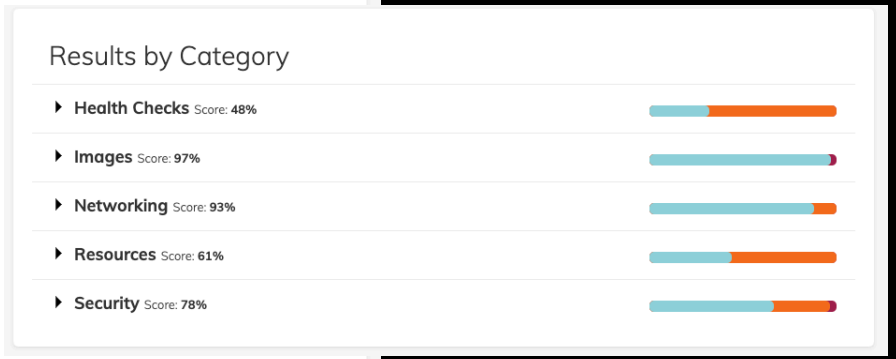
Scan for K8s Best Practices

Some tools to help you see more clearly

Polaris – Validation of best practices in your Kubernetes clusters

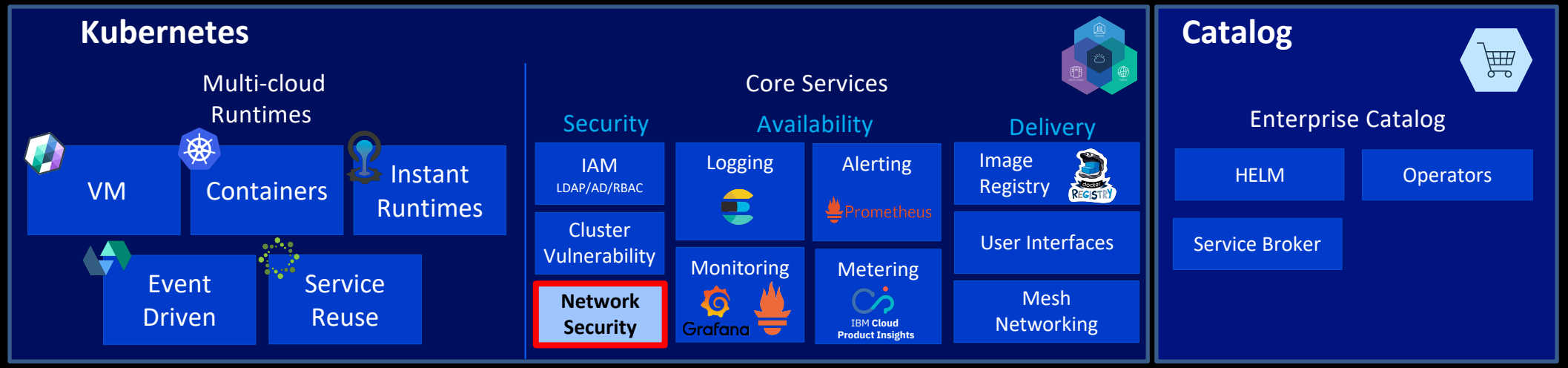




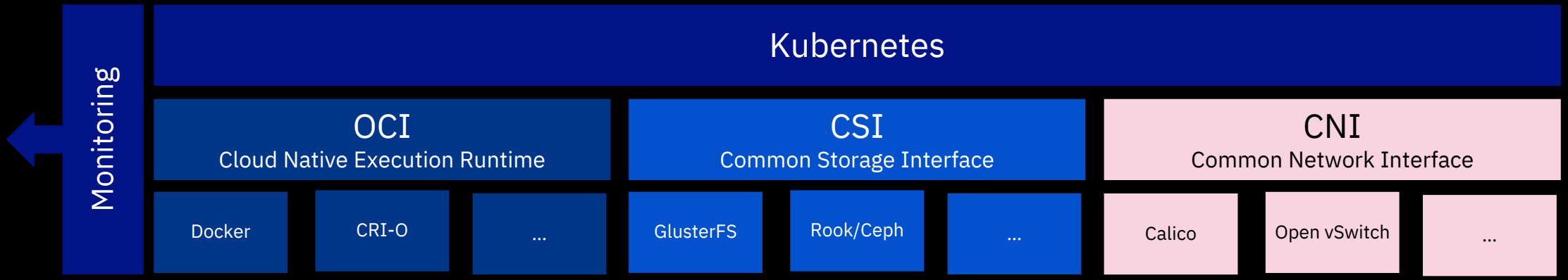


Source: https://github.com/FairwindsOps/polaris

Kubernetes – Core Services – Network Security



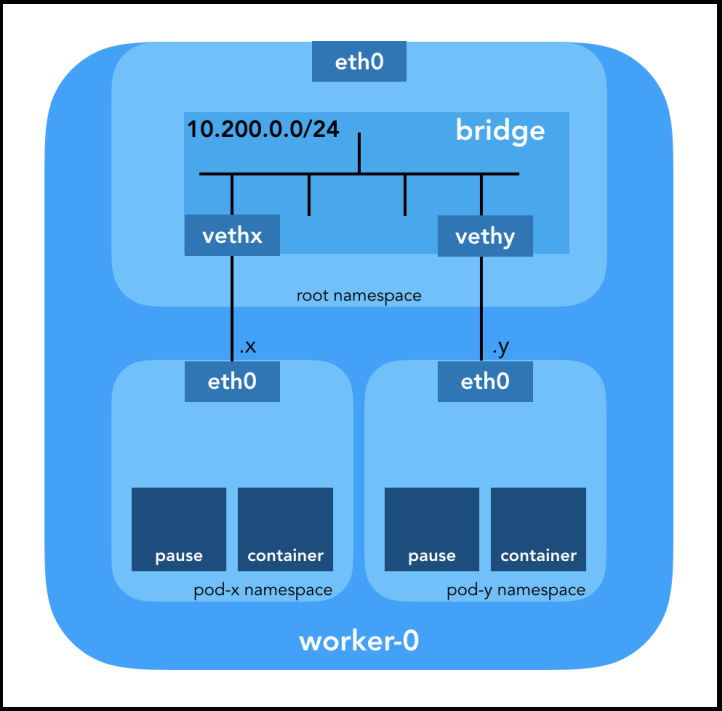
Kubernetes – Core Services – Network Security



CNI – Common Network Interface

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

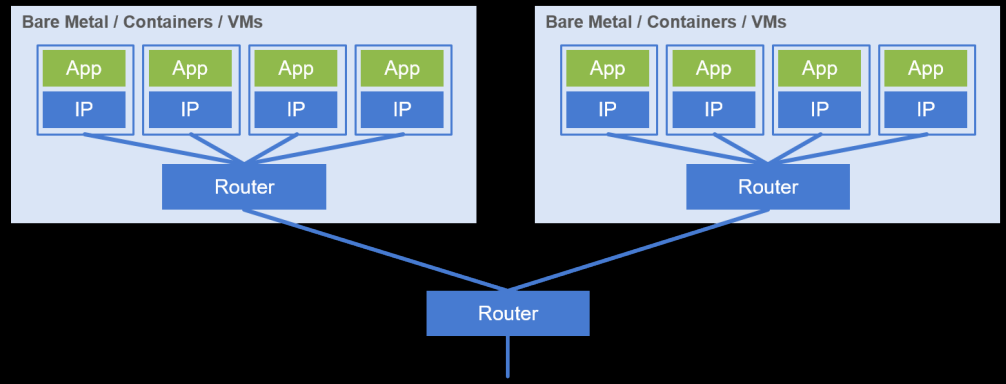
- **Calico**
  Virtual networking and network security for containers, VMs, and bare metal services.
- **Open vSwitch**
  Production quality, multilayer virtual switch
- ...



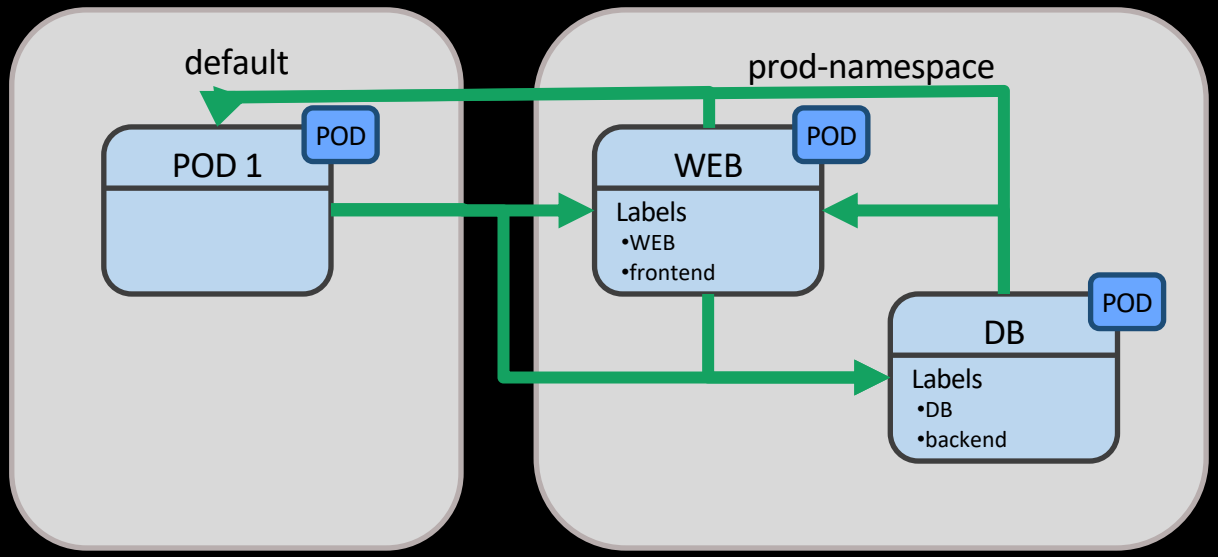
Kubernetes – Core Services – Network Security

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

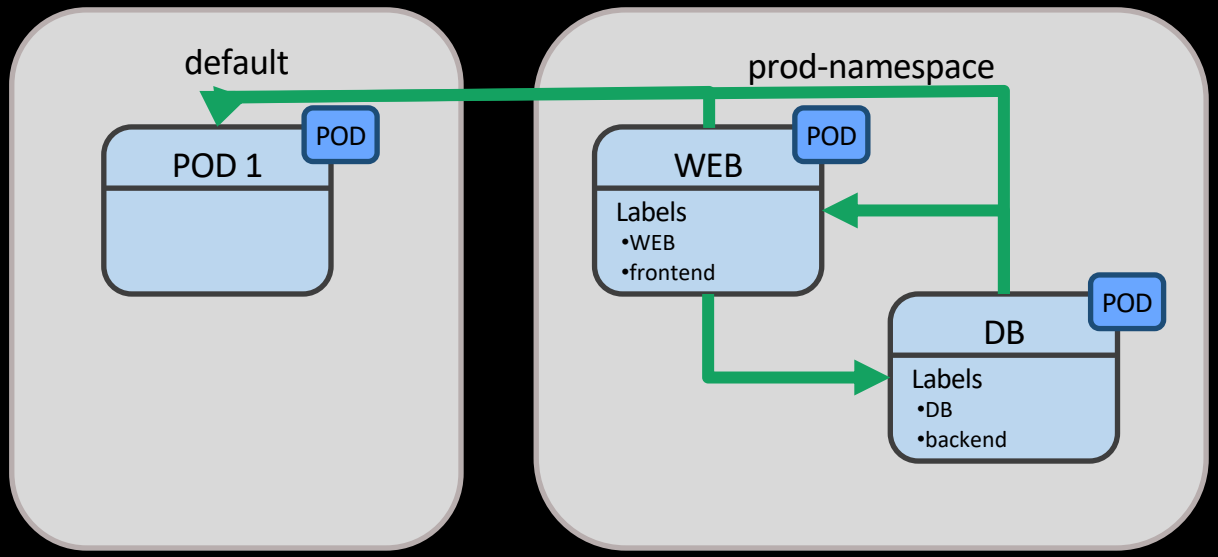
- Has the advantage of being **universal** (DNS, SQL, real-time streaming, ...)
- Can extend beyond the service mesh (including to **bare metal or VM** endpoints not under the control of Kubernetes).
- Calico's policy is enforced at the host node, outside the network namespace of the guest pods.
- Based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



Kubernetes – Core Services – Network Security

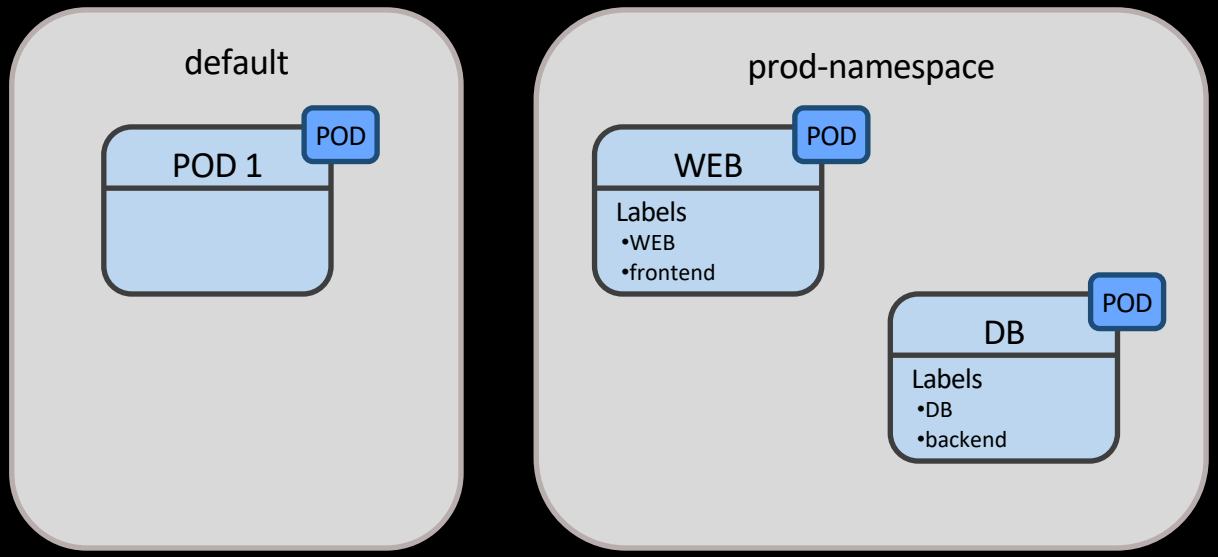


Kubernetes – Core Services – Network Security



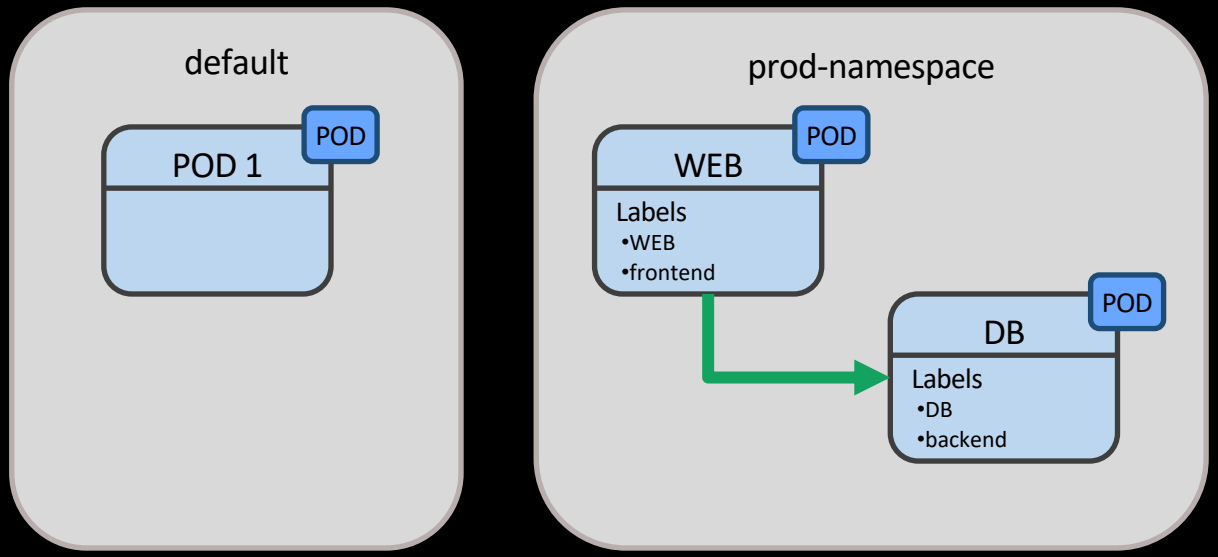

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny
  namespace: demo-namespace
spec:
  podSelector:
    matchLabels: {}
```

Kubernetes – Core Services – Network Security



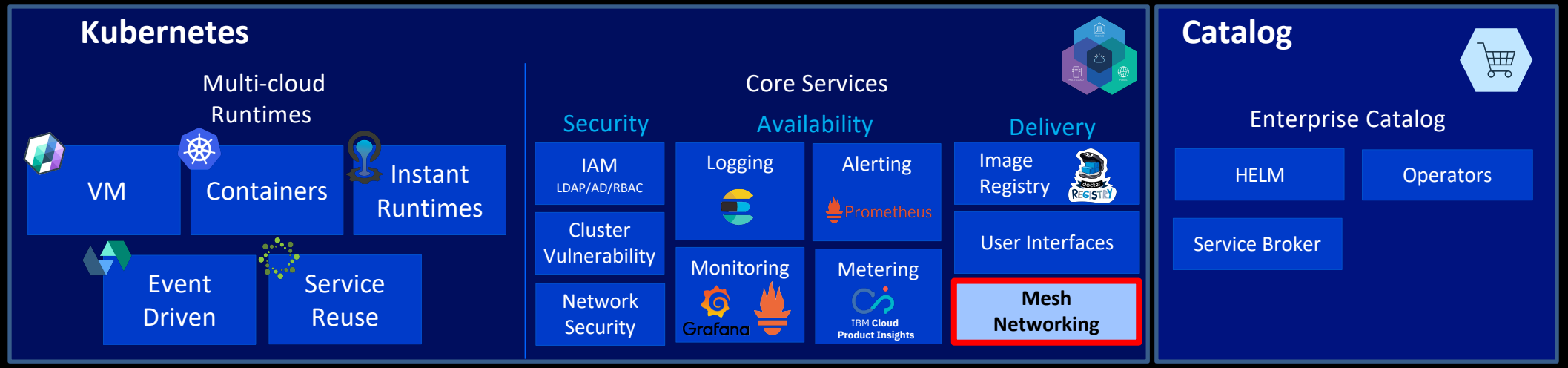

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: default-deny-all
spec:
  podSelector:
    matchLabels: {}
```

Kubernetes – Core Services – Network Security




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

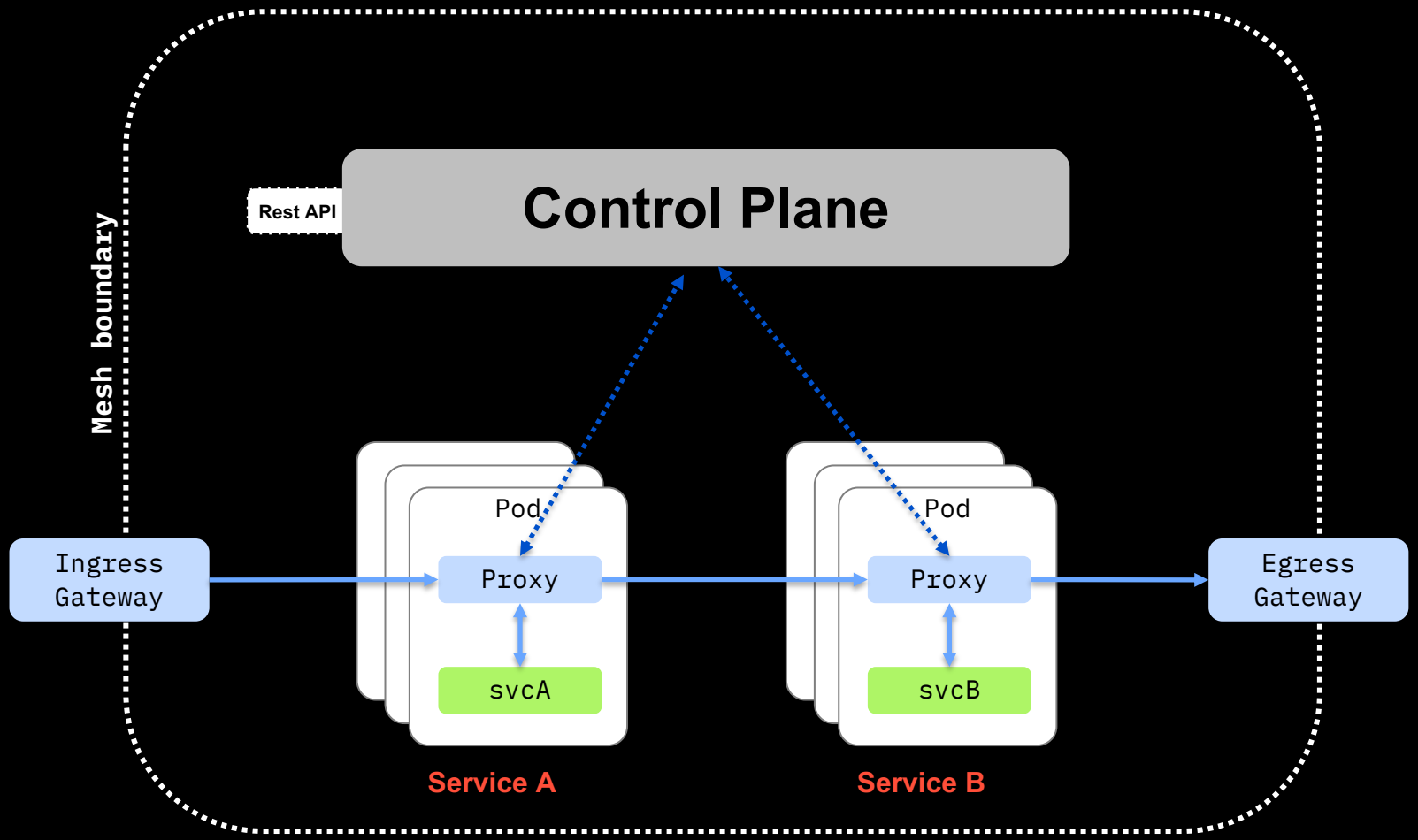
Kubernetes – Core Services – Mesh Networking



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 – Core Services – Network Security

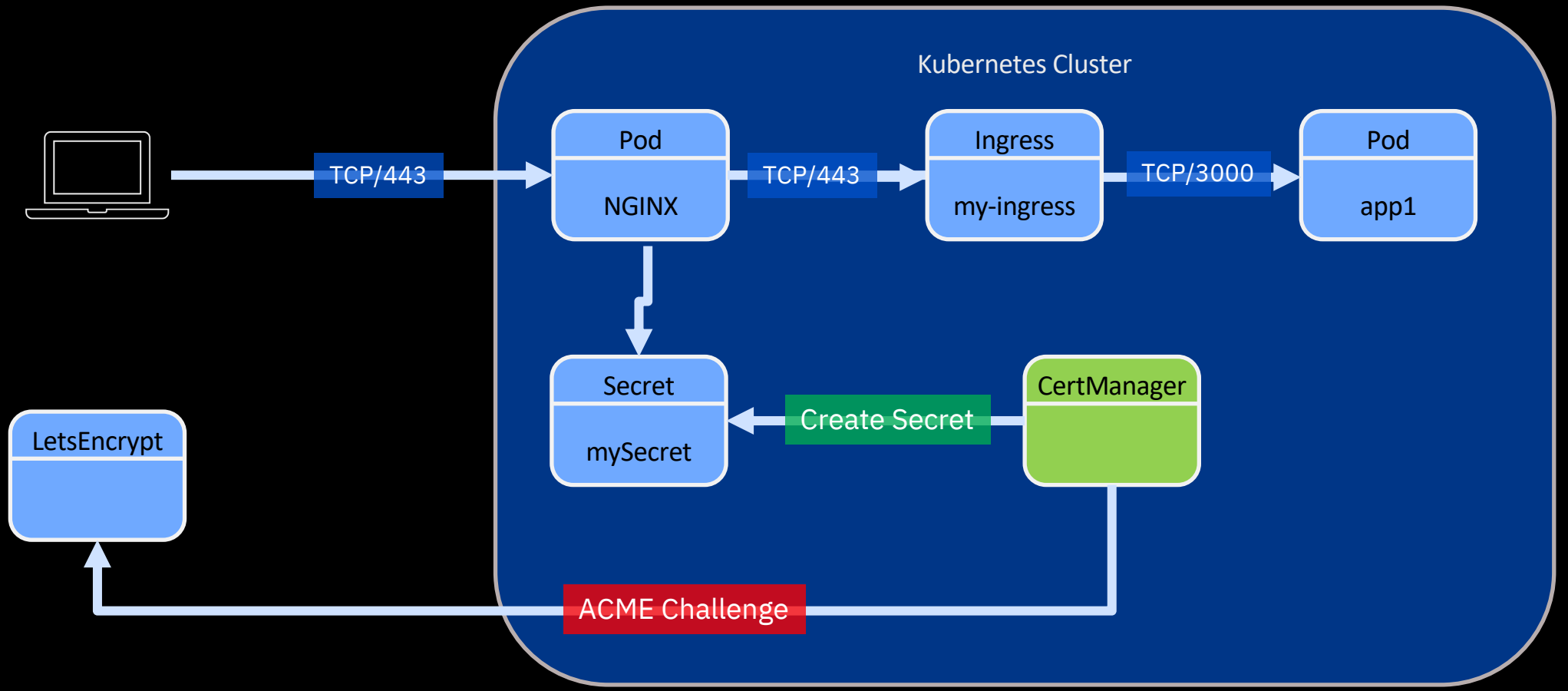
Certificate Management for TLS

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.

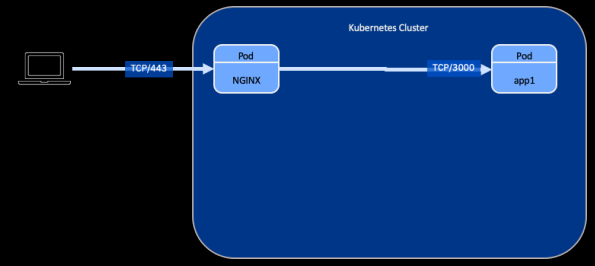
It will ensure certificates are valid and up to date, and attempt to renew certificates at a configured time before expiry.

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

Kubernetes – Core Services – Network Security



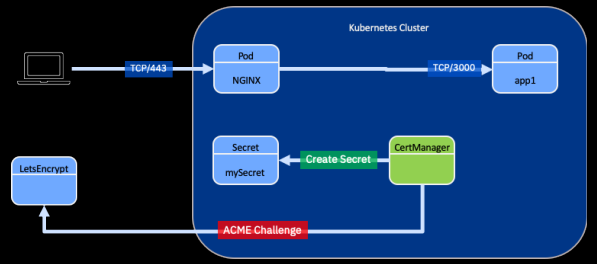
Kubernetes – Core Services – Network Security




```
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 – Core Services – Network Security

```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
```



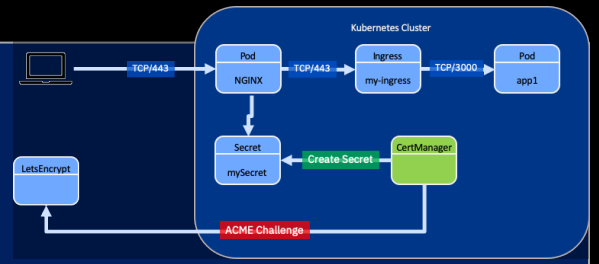

```yaml
apiVersion: certmanager.k8s.io/v1alpha1
kind: ClusterIssuer
metadata:
  name: letsencrypt-staging
spec:
  acme:
    server: https://acme-staging-v02.api.letsencrypt.org/directory
    email: <me@example.com>
    privateKeySecretRef:
      name: mySecret
    http01: {}
```

Kubernetes – Core Services – Network Security

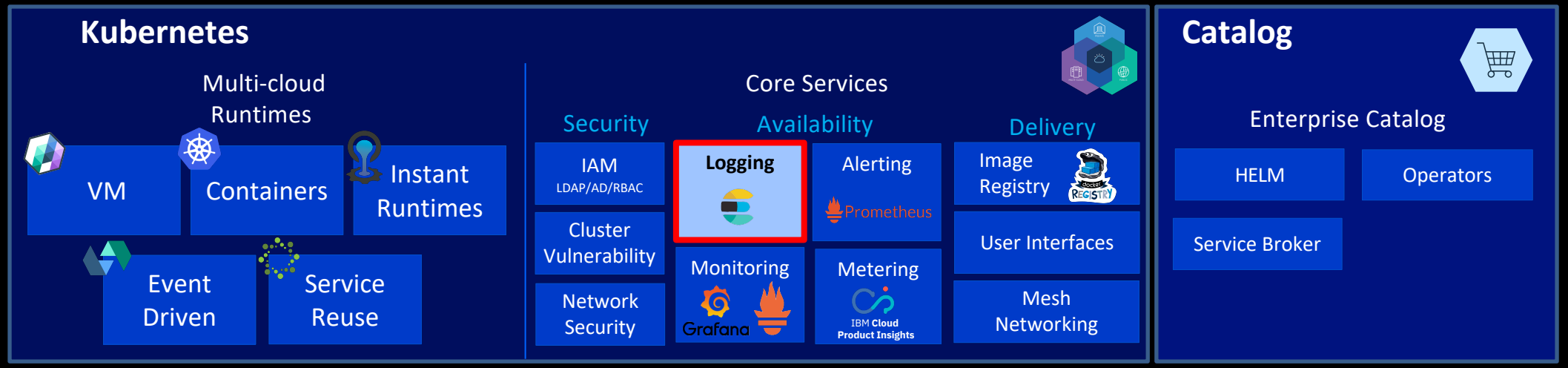
```yaml
apiVersion: certmanager.k8s.io/v1alpha1
kind: ClusterIssuer
metadata:
  name: letsencrypt-staging
spec:
  acme:
    server: https://acme-staging-v02.api.
    email: <me@example.com>
    privateKeySecretRef:
      name: mySecret
    http01: {}
```

```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
```

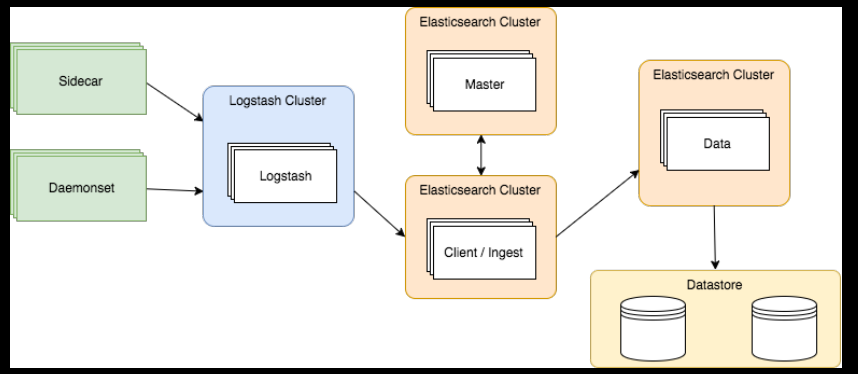
```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: my-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
    certmanager.k8s.io/cluster-issuer: letsencrypt-staging
    kubernetes.io/tls-acme: "true"
spec:
  rules:
  - host: app.example.com
    http:
      paths:
      -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



Kubernetes – Core Services - Logging



- **Logging (ELK):** The easiest and most embraced open-source logging solution for containerized applications



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.

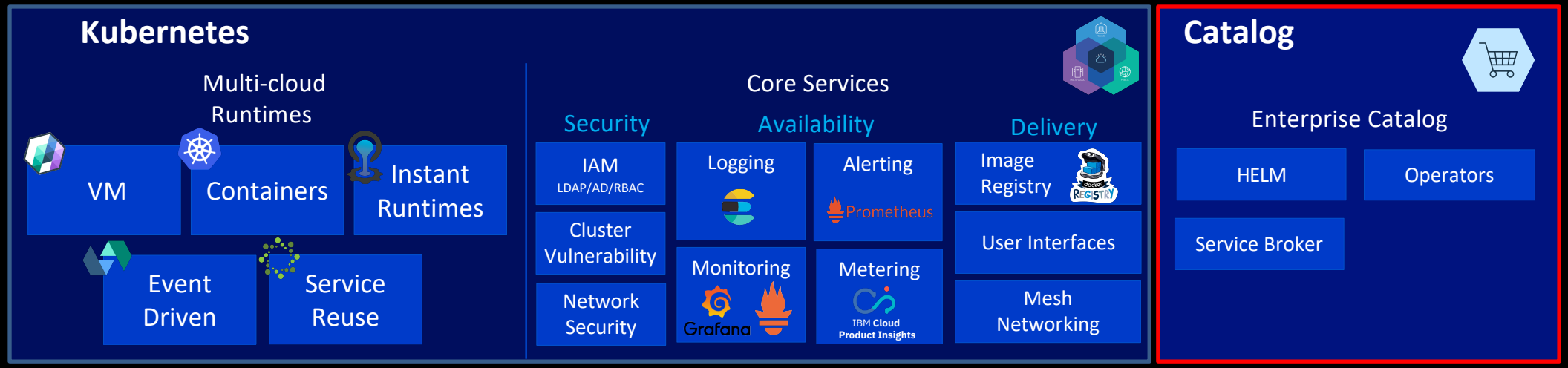
Elasticsearch: An open source full-text search engine based on Lucene. It provides HTTP web interface and schema-free JSON documents.

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

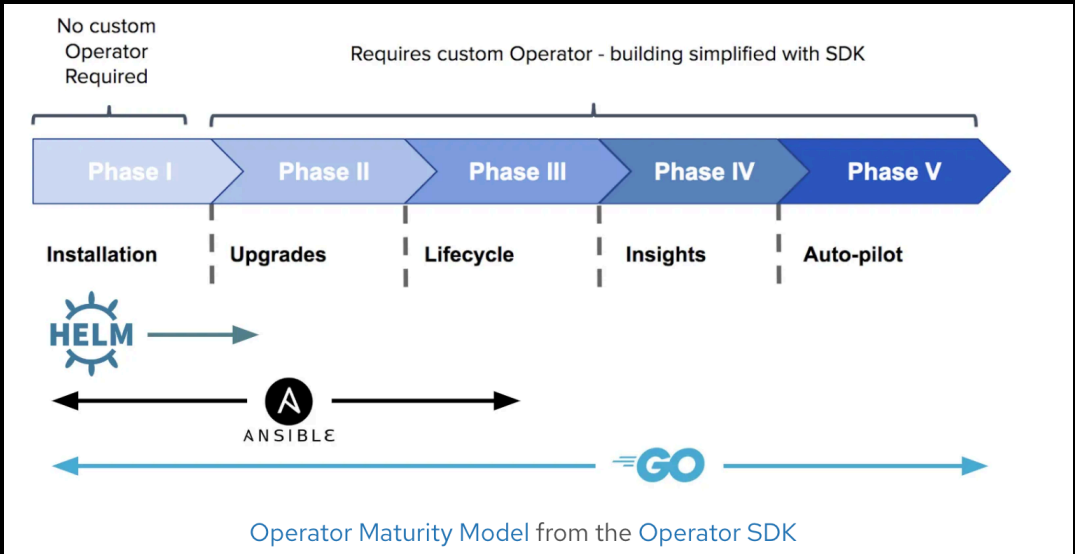
Heapster: The Kubernetes network proxy runs on each node.

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.

Kubernetes – Catalog – Helm & Operators



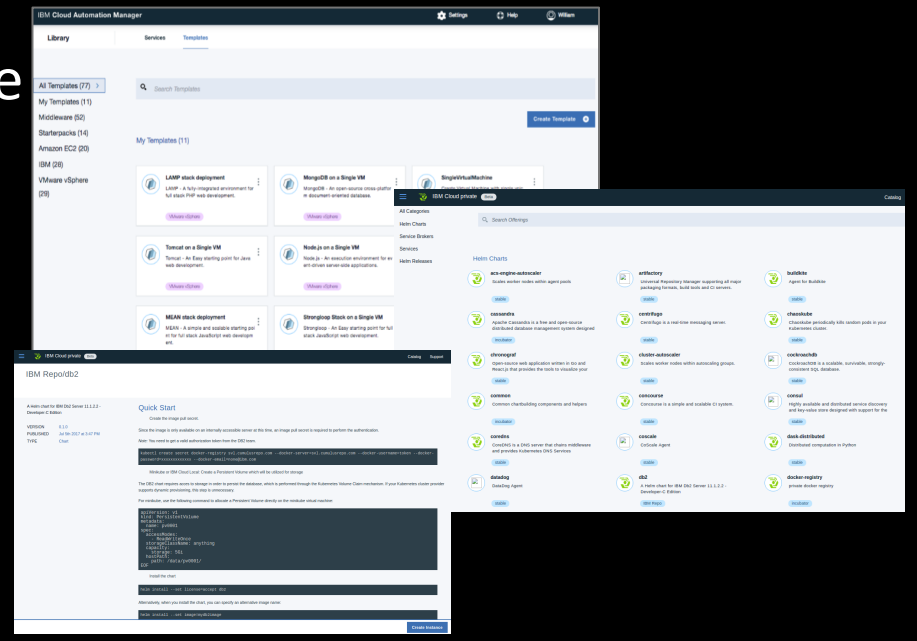
Kubernetes – Catalog – Helm & Operators



Operator Maturity Model from the Operator SDK

Provide a centralized location from which you can control the installation of packages into your cluster.

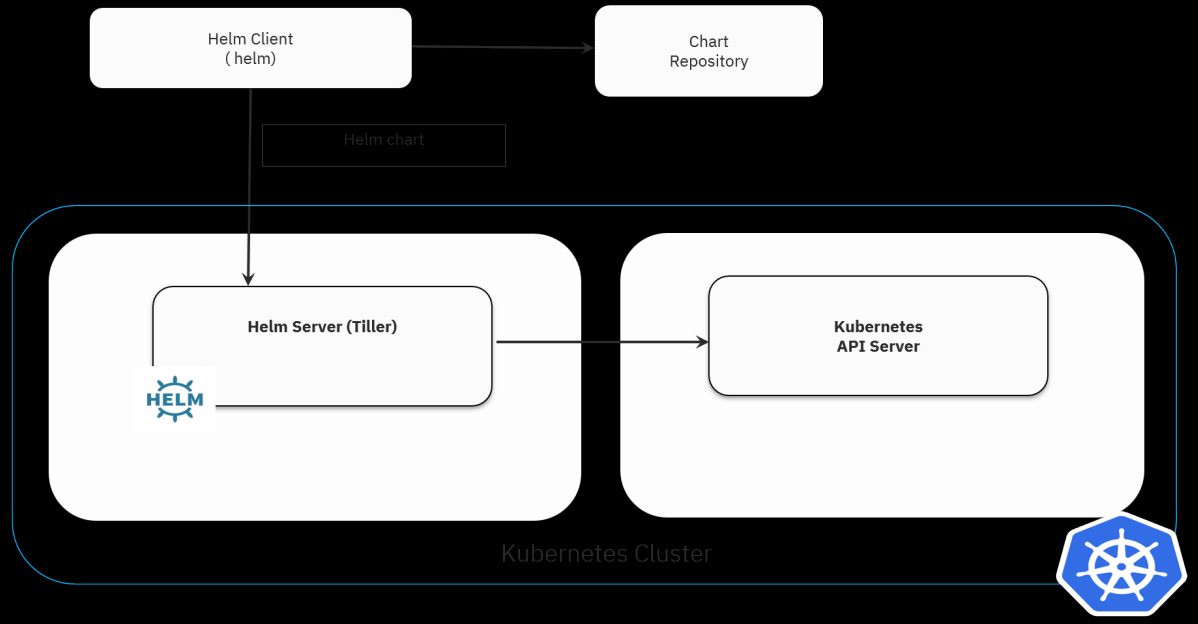
- Catalog of Open source content
- Leverage pre-build curated Helm Chart, Terraform Configurations and Chef scripts
- Reusable and customizable
- Day 2 Operations



Helm Charts

The Helm is a tool that streamlines installation and management of Kubernetes applications. It uses a packaging format called "charts", which are a collection of files that describe Kubernetes resources.

It has a client-server architecture with the client called helm and the server called Tiller



https://helm.sh/

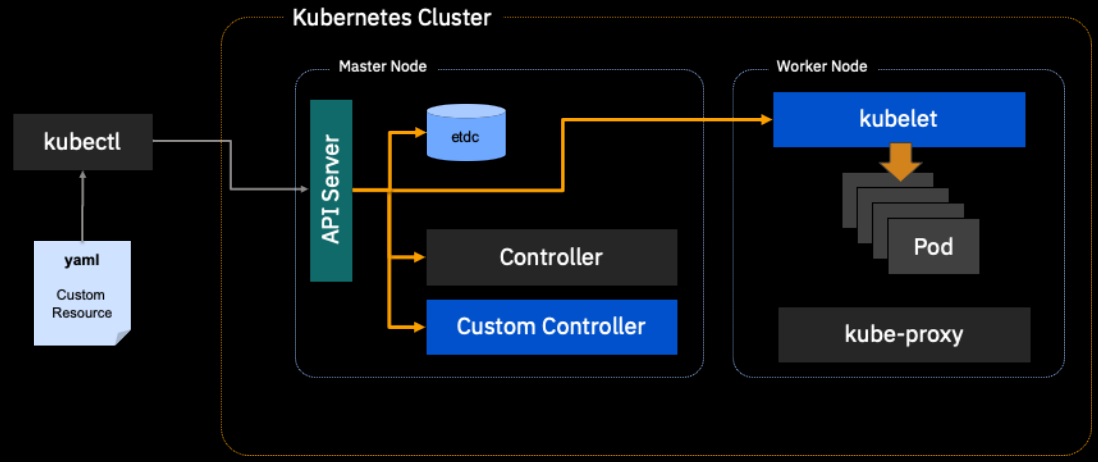
Kubernetes Operators

Operators are a **design pattern** made public in a 2016 CoreOS blog post.

The goal of an Operator is to put **operational knowledge into software**.

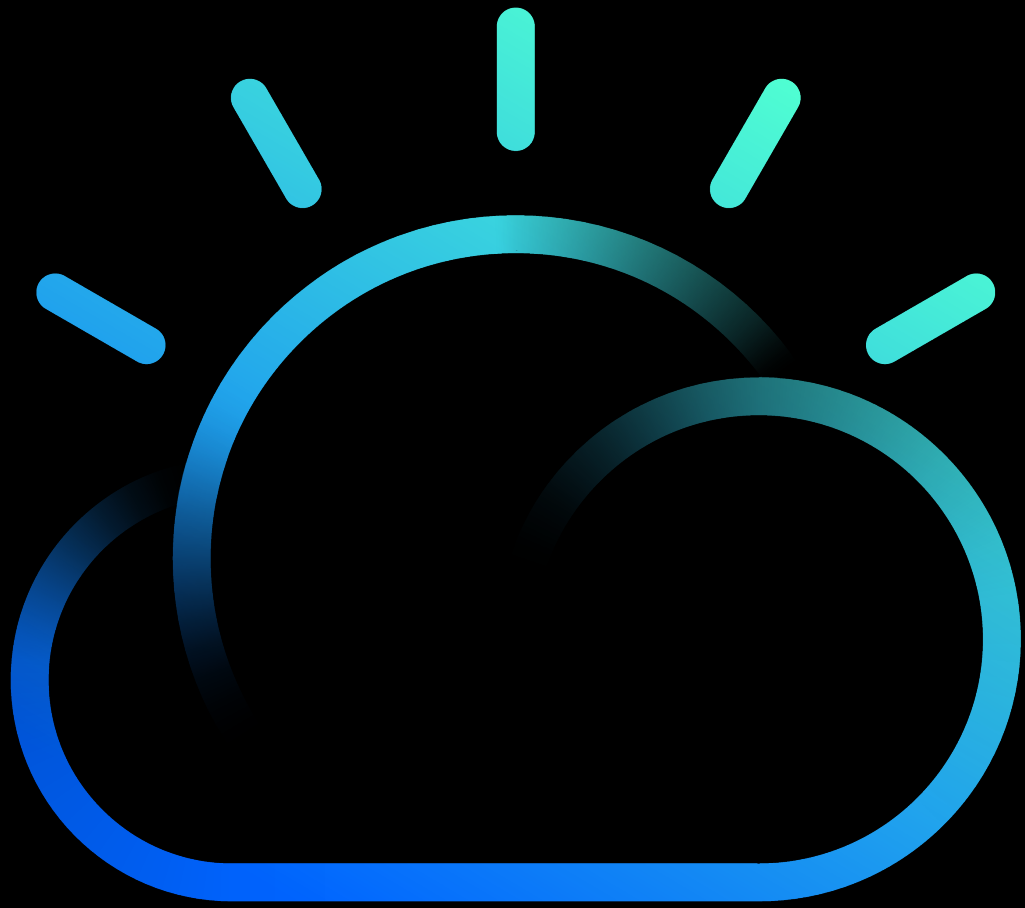
Previously this knowledge only resided in the minds of administrators, various combinations of shell scripts or automation software like Ansible. It was outside of your Kubernetes cluster and hard to integrate.

Operators implement and automate common **Day-1** (installation, configuration, etc) and **Day-2** (re-configuration, update, backup, failover, restore, etc.) **activities** in a piece of software running inside your Kubernetes cluster, by integrating natively with Kubernetes concepts and APIs.



IBM

QUESTIONS?

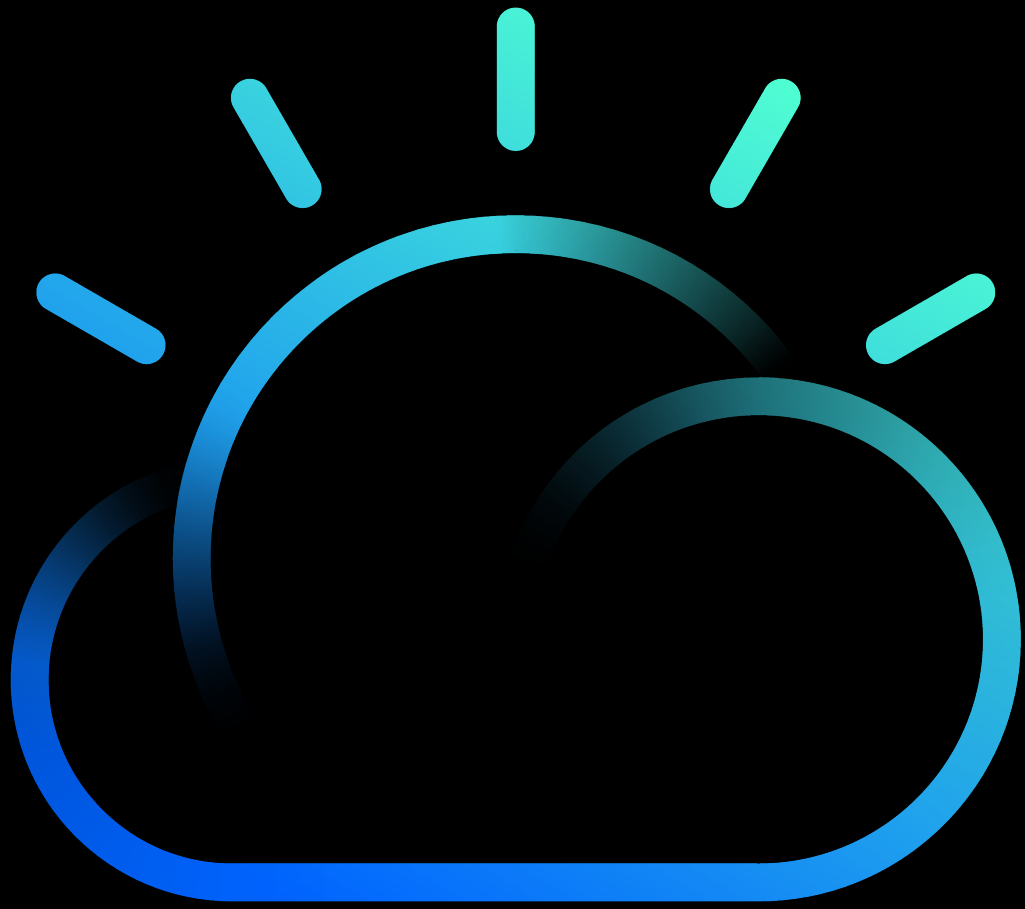


IBM **Cloud**

IBM

The Journey to Cloud
Wrap-up and Tips

03



IBM **Cloud**

Kubernetes – Tip 1

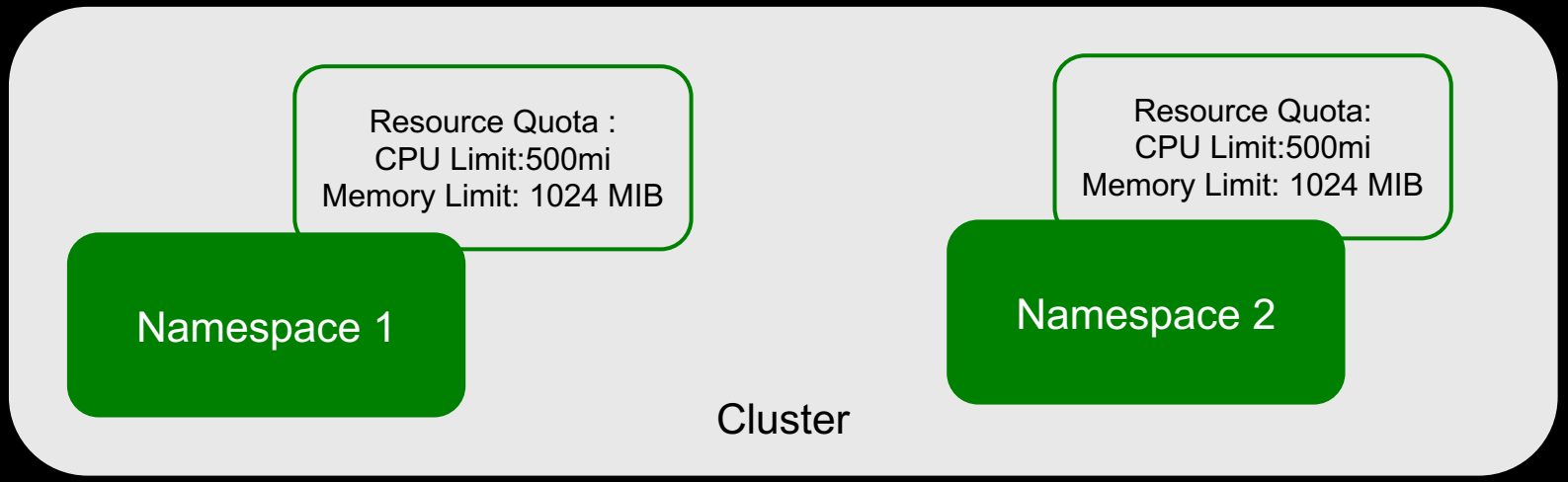
Use Namespaces Liberally

“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 – Tip 2

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 – Tip 3

Scan your containers for vulnerabilities

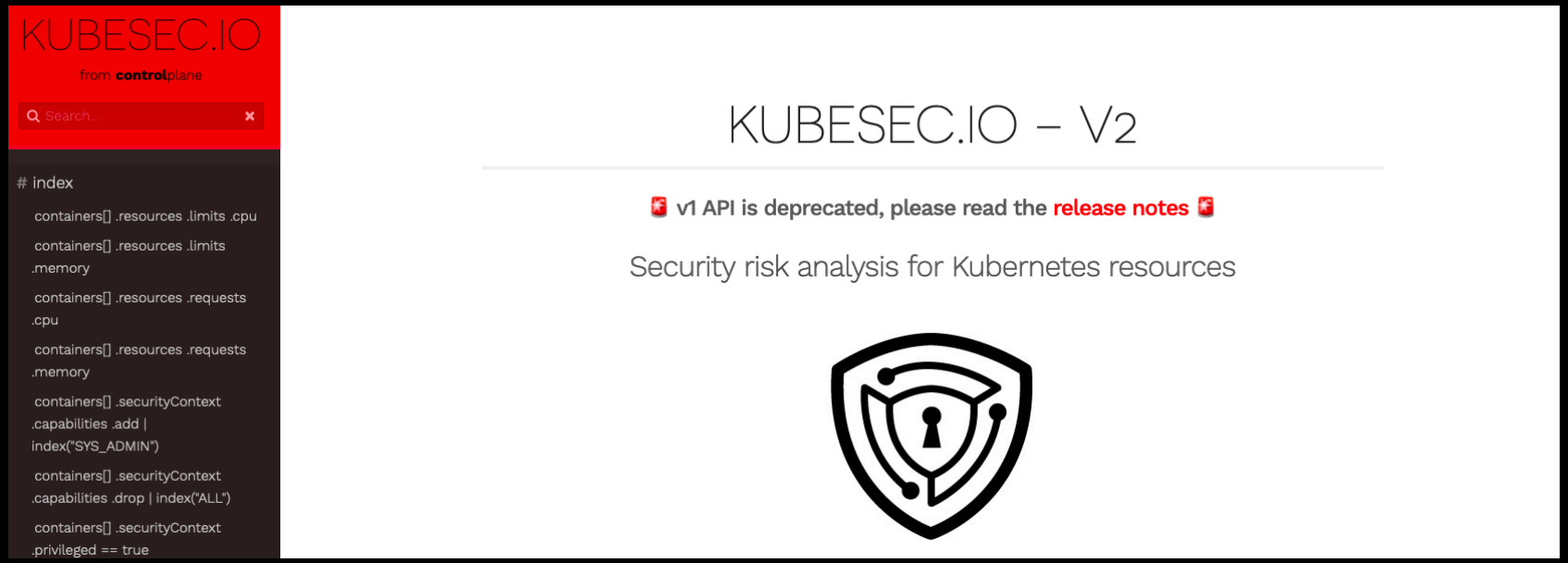
Known vulnerabilities account for a large portion of breaches. Use a tool to scan your containers for them and then mitigate them. Anchore, Clair, and Quay are among my favorite tools, but there are others out there.

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

Kubernetes – Tip 4

Look At KUBSEC.IO

This tool will analyze your deployment yaml and tell you what you should change in order to improve the security of those pods. It even gives you a score that you can use to create a minimum standard. The score incorporates all of the best practices that I have outlined above, as well as several more.



Kubernetes – Tip 5

Occam's razor (also **Ockham's razor** or Ocham's **razor**: Latin: novacula Occami; or law of parsimony: Latin: lex parsimoniae) is the problem-solving principle that states:

“Entities should not be multiplied without necessity.”

Or

“other things being equal, simpler explanations are generally better than more complex ones””

Kubernetes – Tip 6

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 – Tip 6

Secure kubelet

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

Kubernetes – Tip 6

Secure etcd

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

Kubernetes – Tip 6

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.

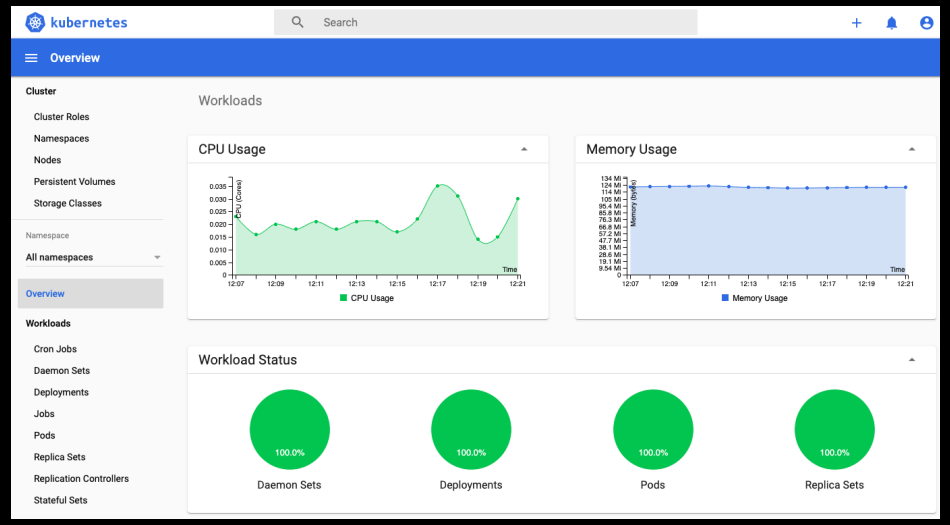
Best practice is to set `--insecure-port flag=0` and remove the `--insecure-bind-address` flag from the API server manifest.
And remove the `--secure-port` flag from the API server spec to ensure that all requests to the secure port are authenticated and authorized.

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

Kubernetes – Tip 7

Secure Kubernetes Dashboard

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



Best practice is to:

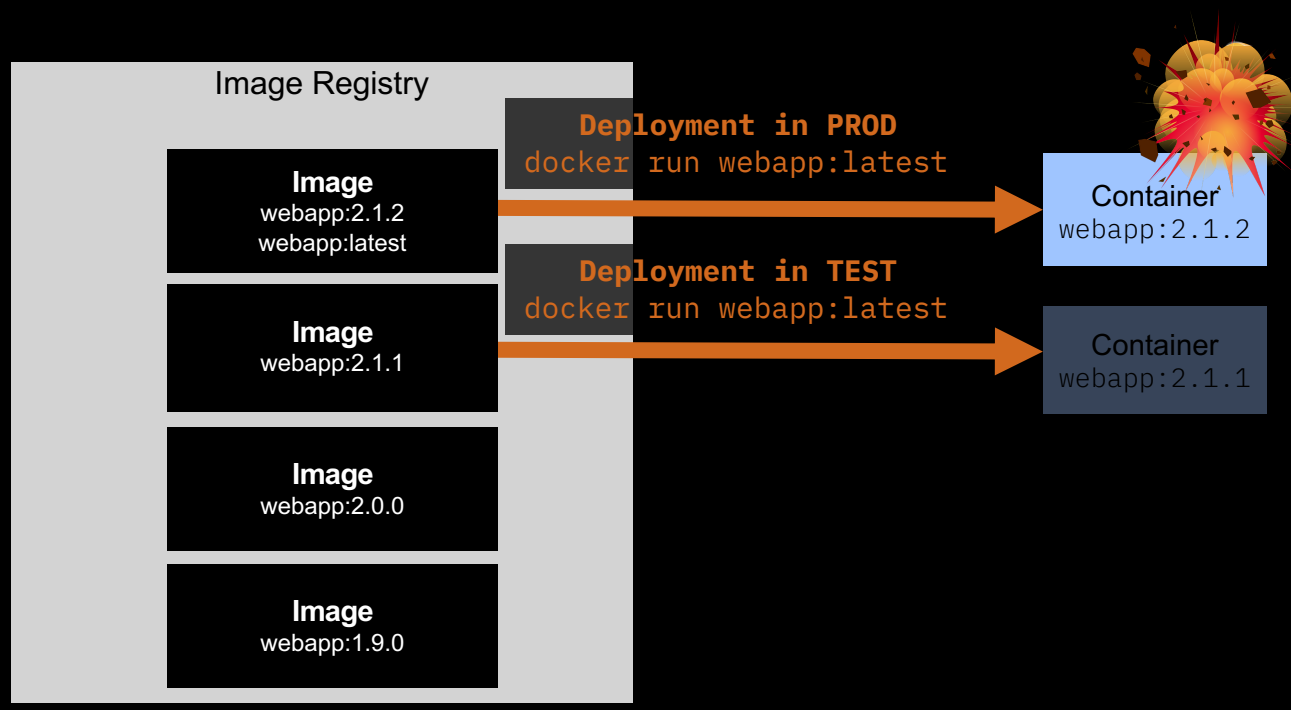
- Allow only authenticated access
- Use RBAC
- Ensure Dashboard service account has limited access
- Don’t expose to the internet (use kubectl proxy for local access)

Kubernetes – Tip 8

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 – Tip 9

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 – Tip 10

Don’t run containers as the root user

By default, containers 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 – Tip 11

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 – Tip 12

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 – Self Check

1. **Where** are images used in containers **coming from**?
2. How long ago were the images **scanned for vulnerabilities**?
3. Which of your containers are impacted by **known vulnerabilities**, and what's their severity?
4. Are any of these containers in **production** impacted by a known **vulnerability**?
5. Which vulnerable running containers or deployments should be **prioritized for remediation**?
6. Which deployments are using **privileged containers**, meaning they have full access to the host operating system?

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

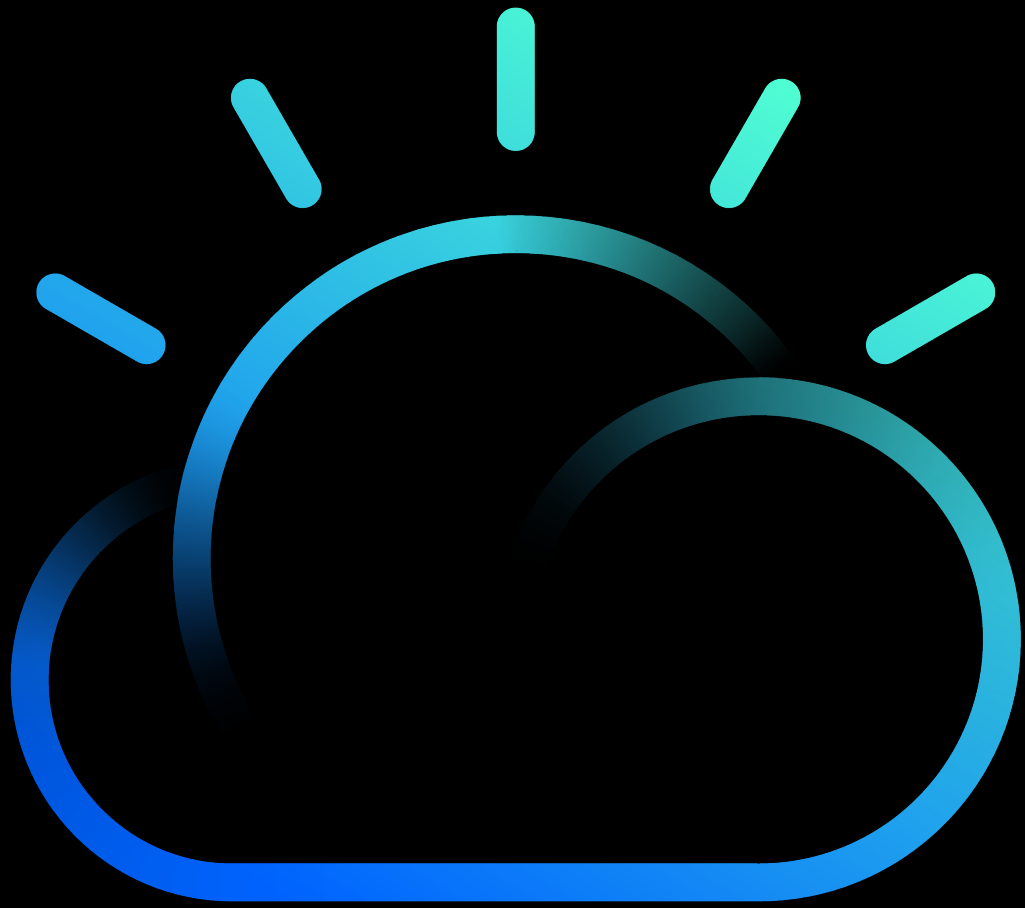
Kubernetes – Self Check

7. What container **services are exposed** outside of the Kubernetes cluster?
8. Can we tell which **processes** are **running** in any container in any cluster?
9. Which **network communication** pathways are **active** but are not being used in production?
10. What **team** in the organization **owns** a particular **running application**?
11. How many of my clusters, namespaces, and nodes must **adhere to to specific regulations**.

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

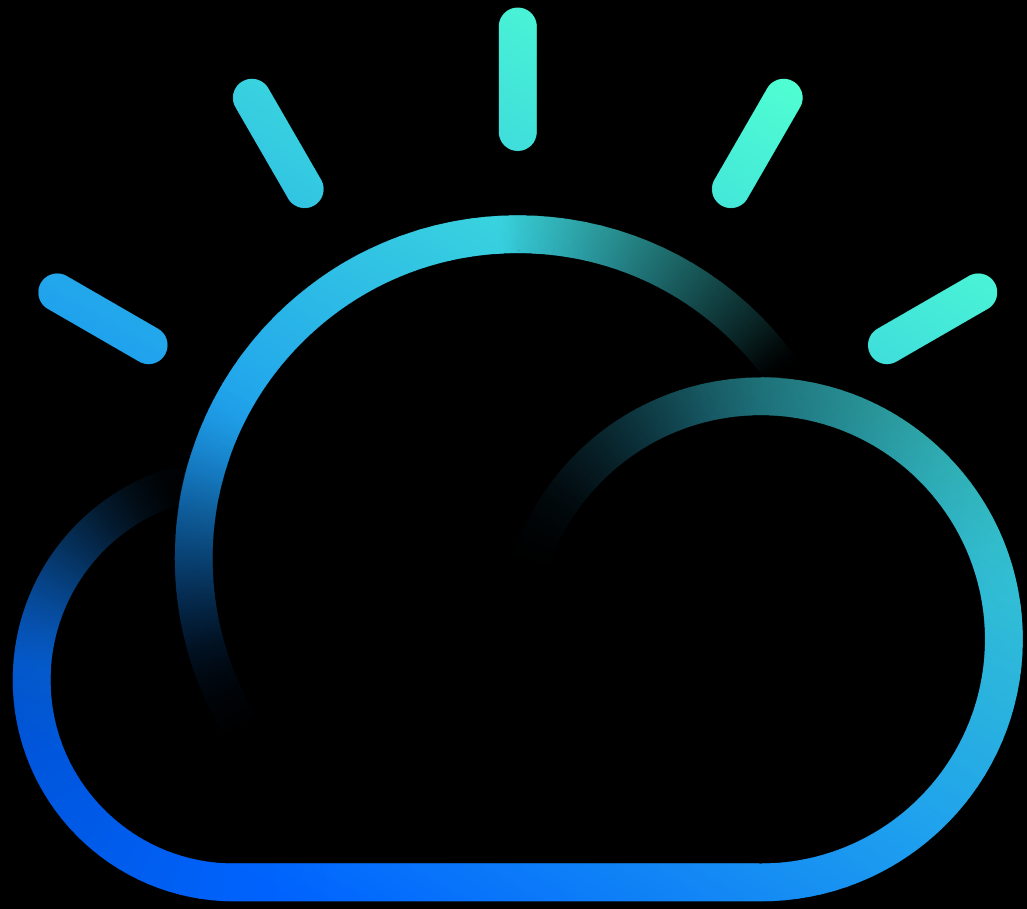
IBM

QUESTIONS?



IBM **Cloud**

The Journey to Cloud
K8s Security - *Hands-On*



04

IBM **Cloud**

Remember your Team Color

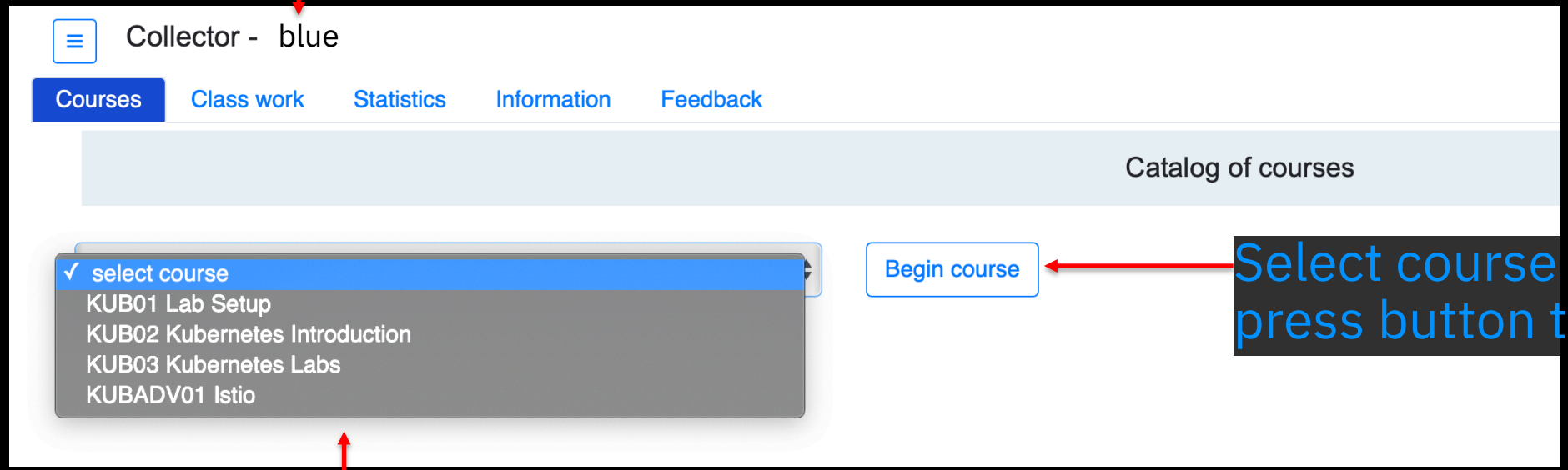
black 31701	olive 31711	peru 31715
white 31702	brown 31712	chocolate 31716
red 31703	lightblue 31713	orchid 31717
blue 31704	orange 31708	gold 31718
yellow 31705	purple 31709	pink 31719
lime 31706	maroon 31710	violet 31720
cyan 31707	firebrick 31714	

Collector - Accessing team web site

http://158.177.137.195:{port#}

blue 31704

Team name / color will be shown



Current course catalog –
Select **JTC16 Kubernetes Security Labs**

JTC16 Kubernetes Security Labs

Lab 1 : NetworkPolicies

Lab 2 : Role Based Acces Control (RBAC)

Lab 3 : Service Accounts

Lab 4 : Security Tooling

Lab 5 : Image scanning

JTC16 Kubernetes
Security Labs

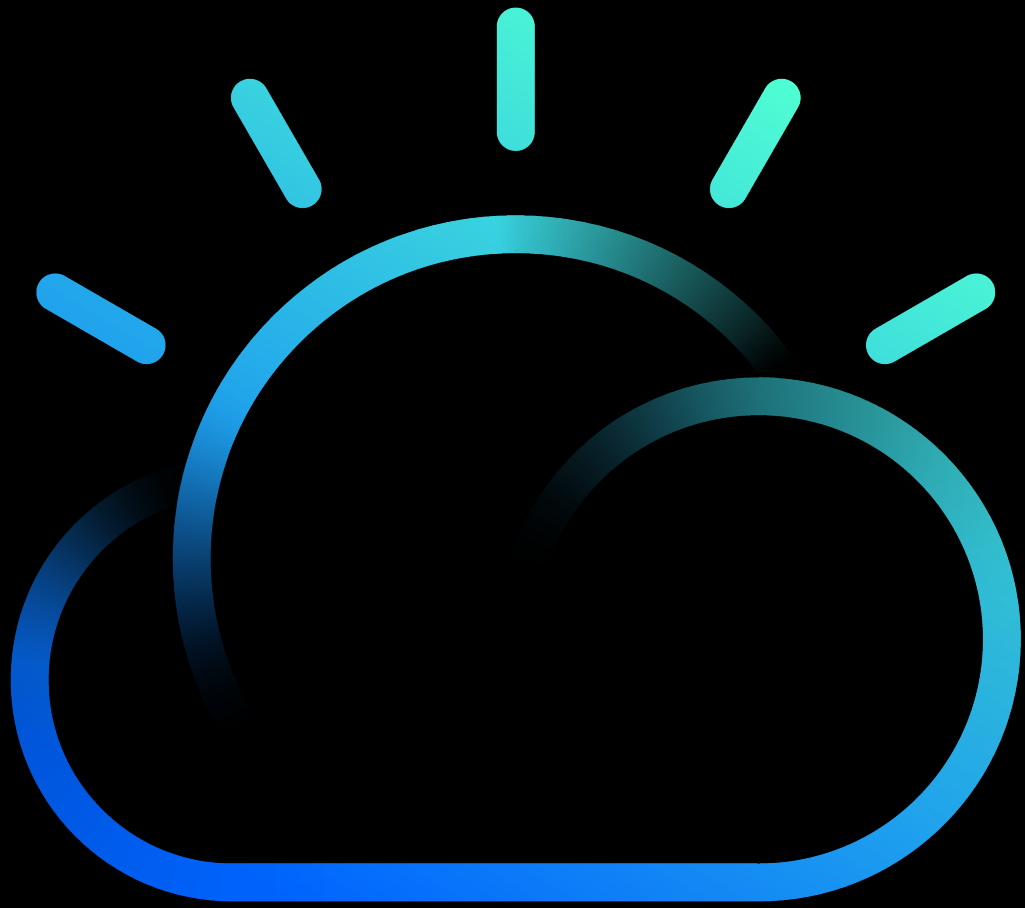
READY
SET
GO!!!!

http://158.177.137.195:{port#}

Duration: 60 mins

IBM

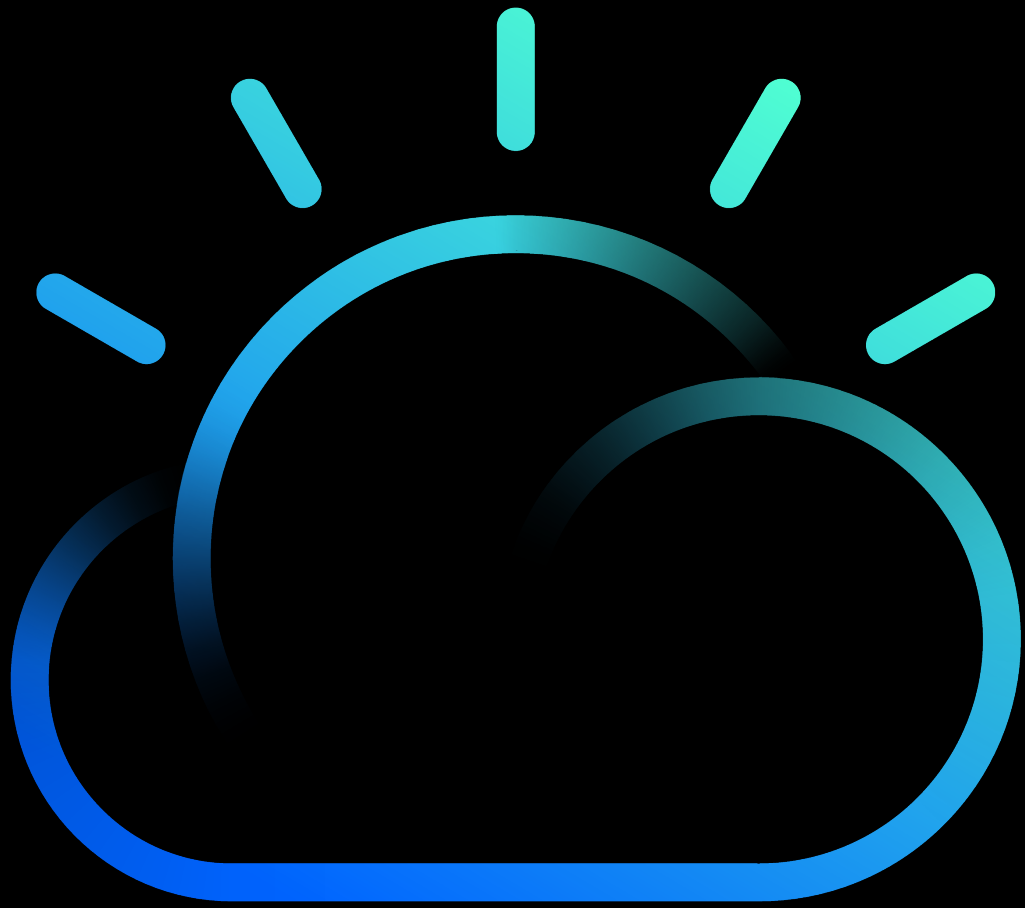
QUESTIONS?



IBM **Cloud**

The Journey to Cloud
Wrap Up

99



IBM **Cloud**

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/

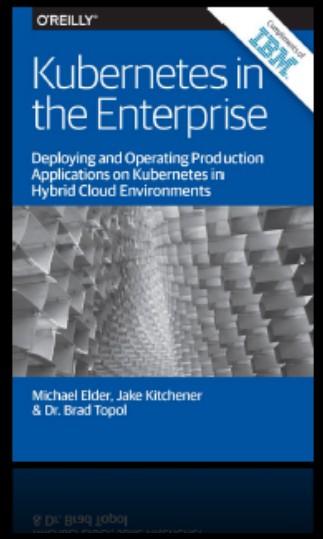
https://kubernetes.io/blog/2019/03/21/a-guide-to-kubernetes-admission-controllers/

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 Security – Some Reading Tips

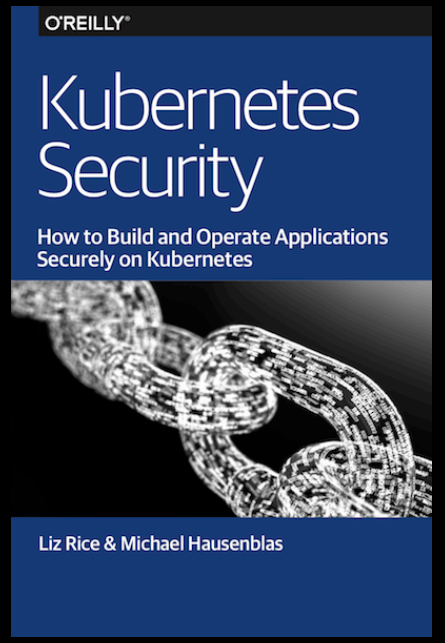


We wrote the books on cloud adoption…

ibm.com/cloud/garage/adoption

…and Kubernetes in the enterprise

https://ibm.co/2LQketN



https://kubernetes-security.info/

Sources and documentation will be available here:

https://github.com/niklaushirt/k8s_training_public

https://github.com/niklaushirt/training

THANK YOU!!!!

Niklaus Hirt

nikh@ch.ibm.com

@nhirt