MIRANTIS

PRO TALK:
Kubernetes Security Workshop

Avinash Desireddy
Sr. Solutions Architect

SPEAKER

AVINASH DESIREDDY

Sr. Solutions Architect @ Mirantis

- /avinashdesireddy
- /avinashdesireddy
- /avinashdesiredd

MODERATOR

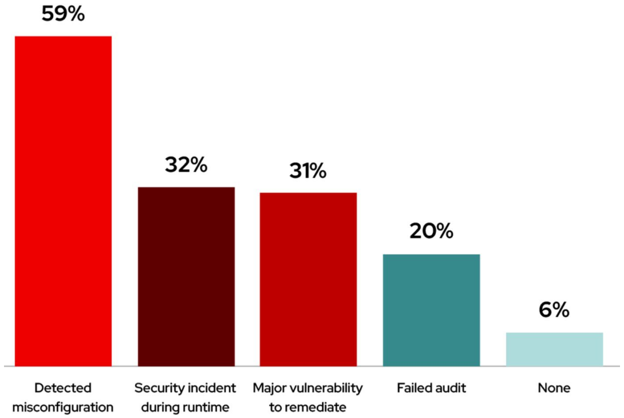
ANOOP KUMAR

Director, Professional Services @ Mirantis

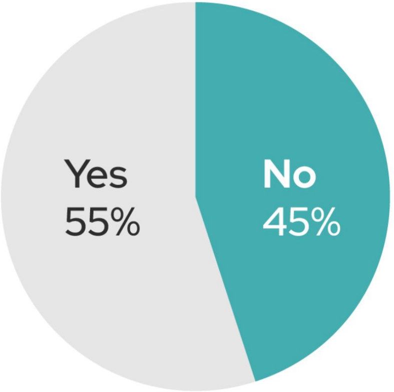
- /anokun7
- /anoopkumarv
- /anooplive

Kubernetes: Adoption, Security & Market Trends

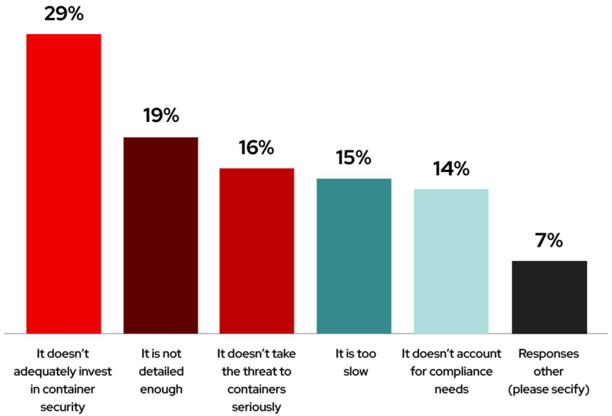
In the past 12 months, what security incidents or issues related to containers and/or Kubernetes have you experienced?



Have you ever delayed or slowed down application deployment into production due to container or Kubernetes security concerns?

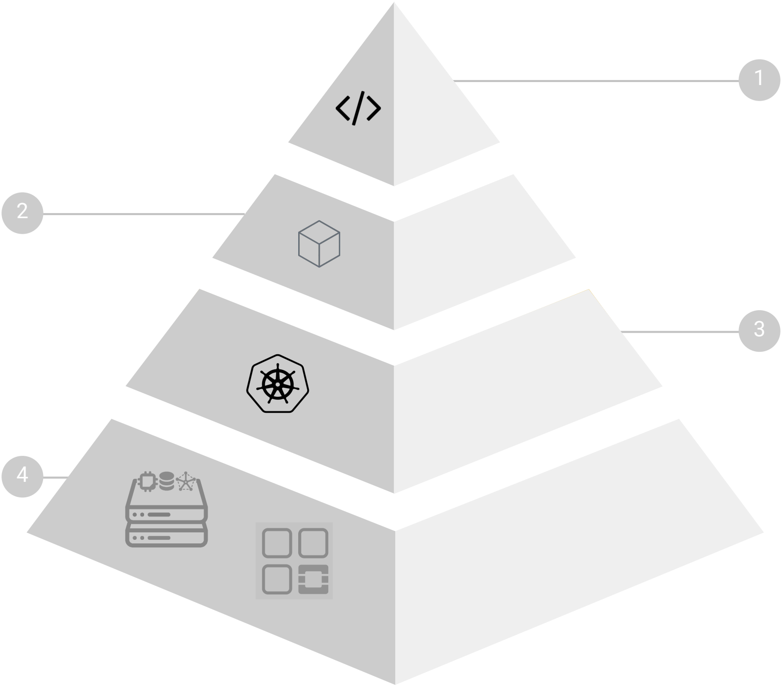


What is your biggest concern about your company's container strategy?



Source: The State of Containers and Kubernetes Security Report - Survey by StackRox(RedHat) in 2021

The 4 C's of Cloud-Native Security



Code

Code Best Practices
Vulnerability scanners

Container

Restrict Images, Privileged

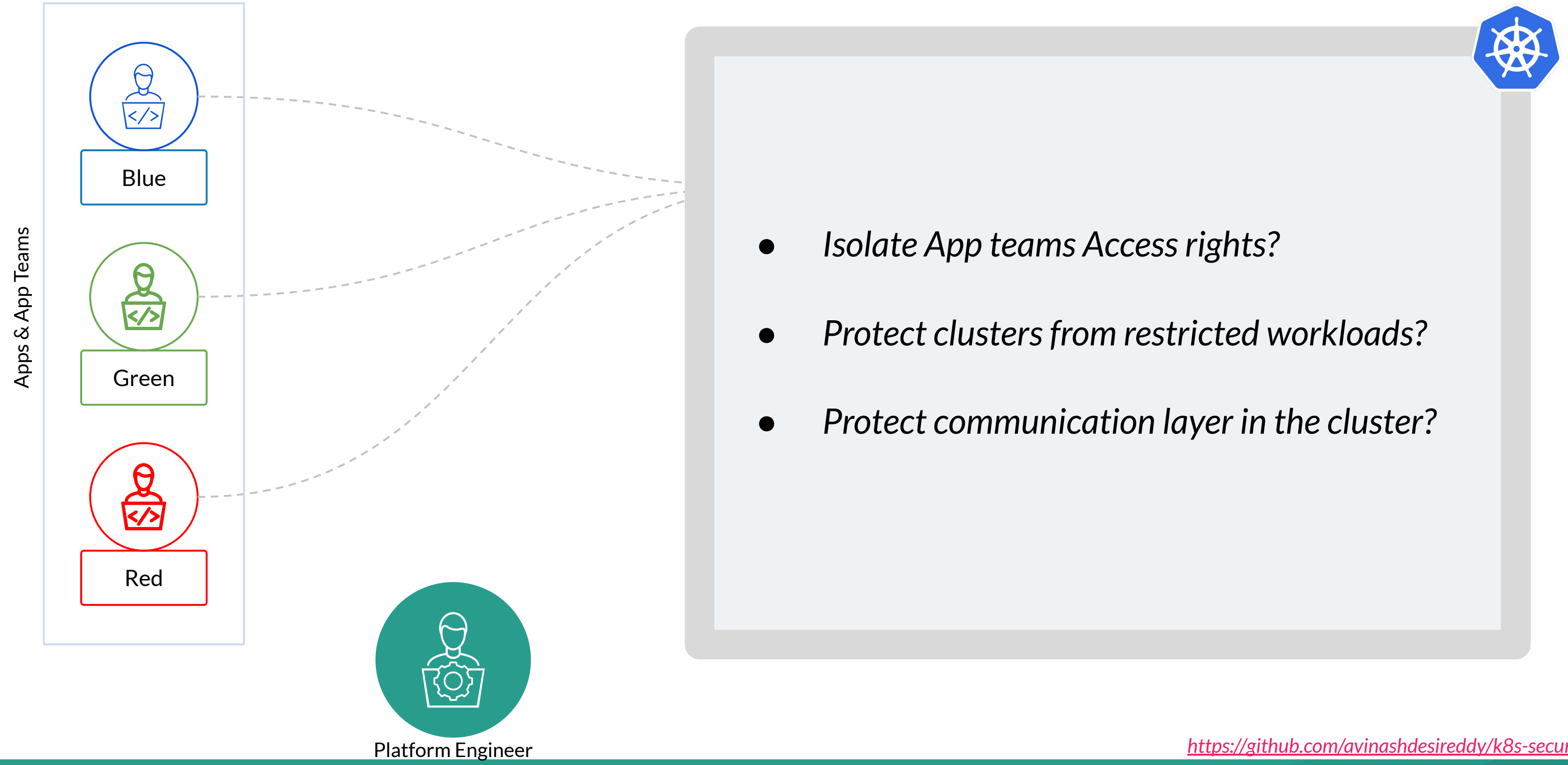
Cluster

Authentication,
Authorization, Admission, Network Policy

Cloud

Datacenter, Network,
Servers

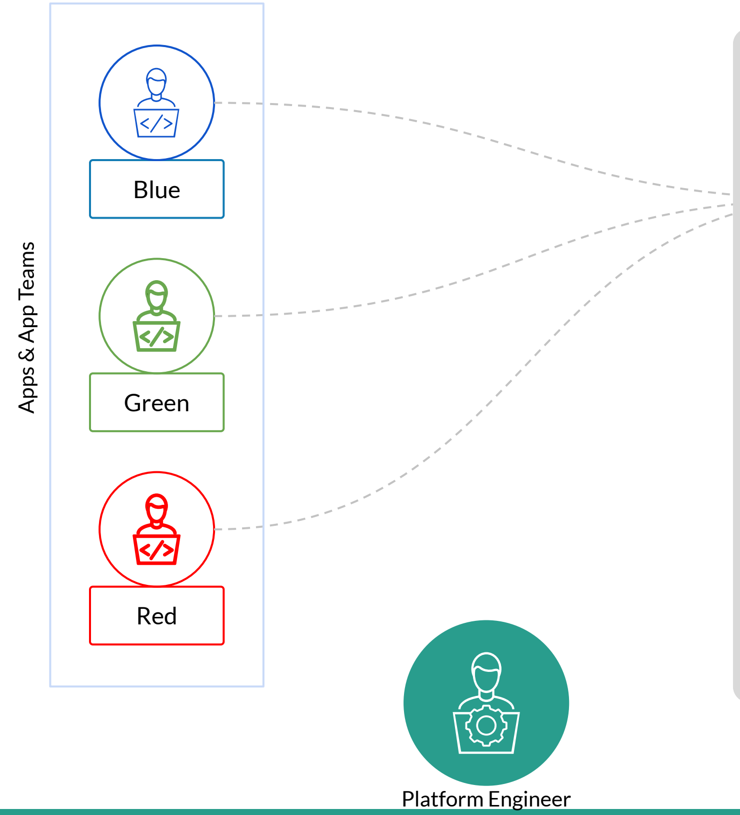
Overview: Onboard Apps Securely



- *Isolate App teams Access rights?*
- *Protect clusters from restricted workloads?*
- *Protect communication layer in the cluster?*

https://github.com/avinashdesireddy/k8s-security-workshop.git

Scenario #1 - Grant access to Users



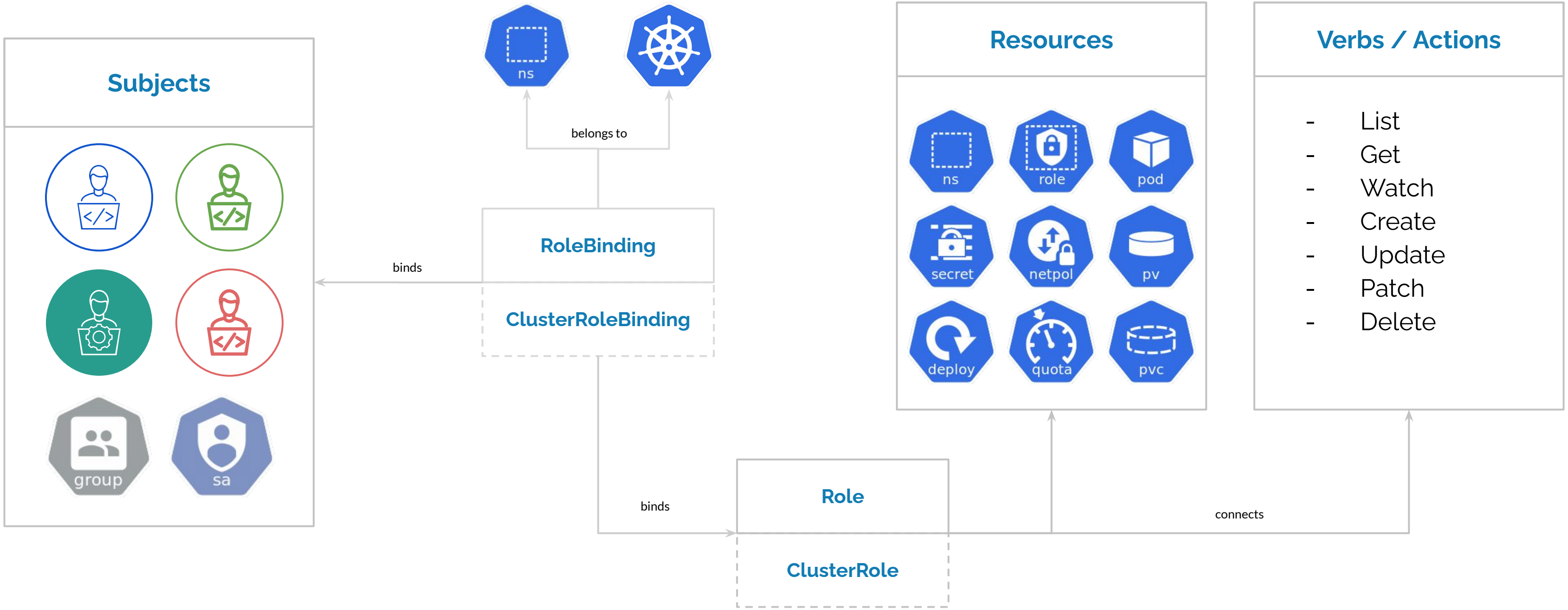
Role-Based Access Control

A way of granting users access to Kubernetes API resources

- ❏ What API Resources a user should access?
- ❏ What Operations(Verbs) can be performed?
- ❏ Who can grant access?

https://github.com/avinashdesireddy/k8s-security-workshop.git

Role Based Access Control

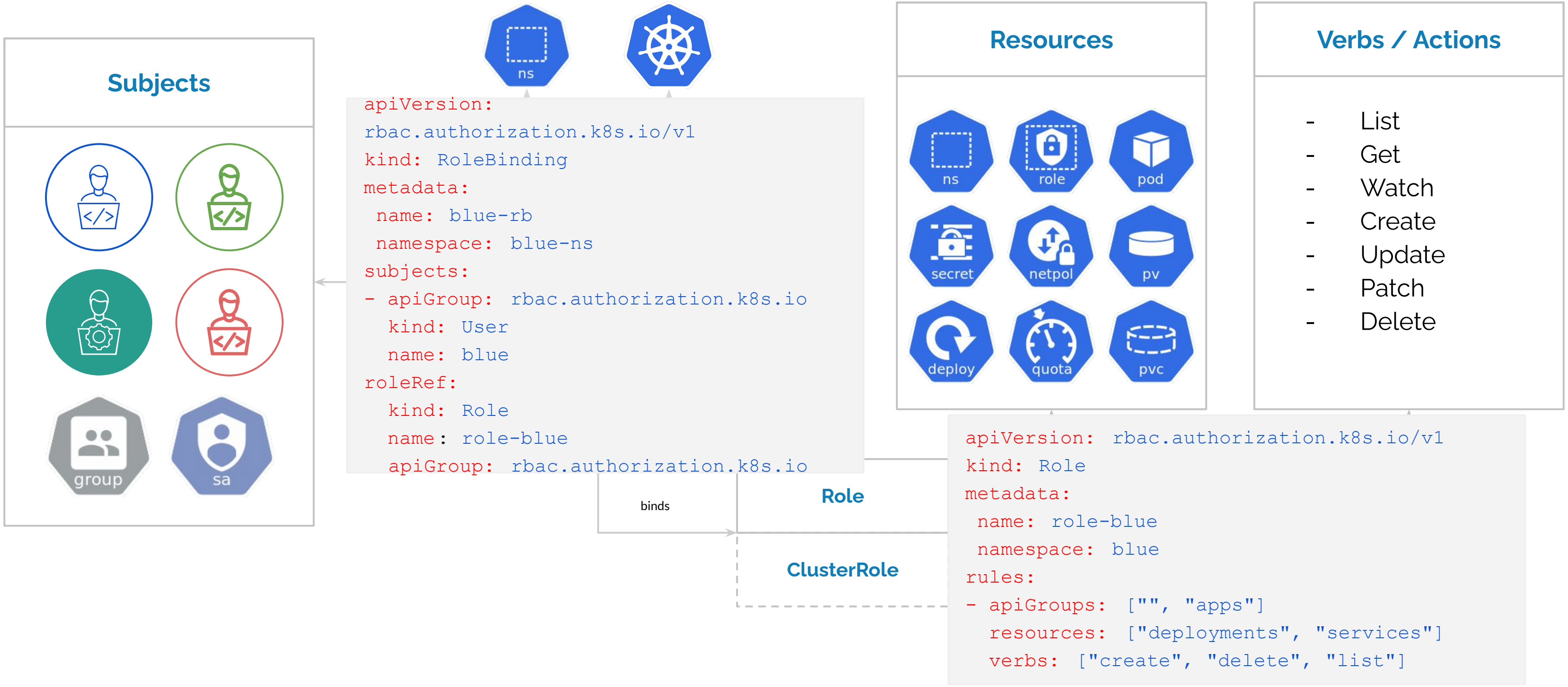


Verbs / Actions

- List
- Get
- Watch
- Create
- Update
- Patch
- Delete

https://github.com/avinashdesireddy/k8s-security-workshop.git

Role Based Access Control



```yaml
apiVersion:
rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
 name: blue-rb
 namespace: blue-ns
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: blue
roleRef:
  kind: Role
  name : role-blue
  apiGroup: rbac.authorization.k8s.io
```

Verbs / Actions

- List
- Get
- Watch
- Create
- Update
- Patch
- Delete

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
 name: role-blue
 namespace: blue
rules:
- apiGroups: ["", "apps"]
  resources: ["deployments", "services"]
  verbs: ["create", "delete", "list"]
```

https://github.com/avinashdesireddy/k8s-security-workshop.git

Demo: RBAC

- Create Namespaces
- Grant Access to App Users to respective Namespaces
- Deploy 3 applications

Environment

aws

Infrastructure

Nodes, LB, DNS, etc

Mirantis Kubernetes Engine

1 Manager, 3 Worker

Version - 1.21.3

LENS

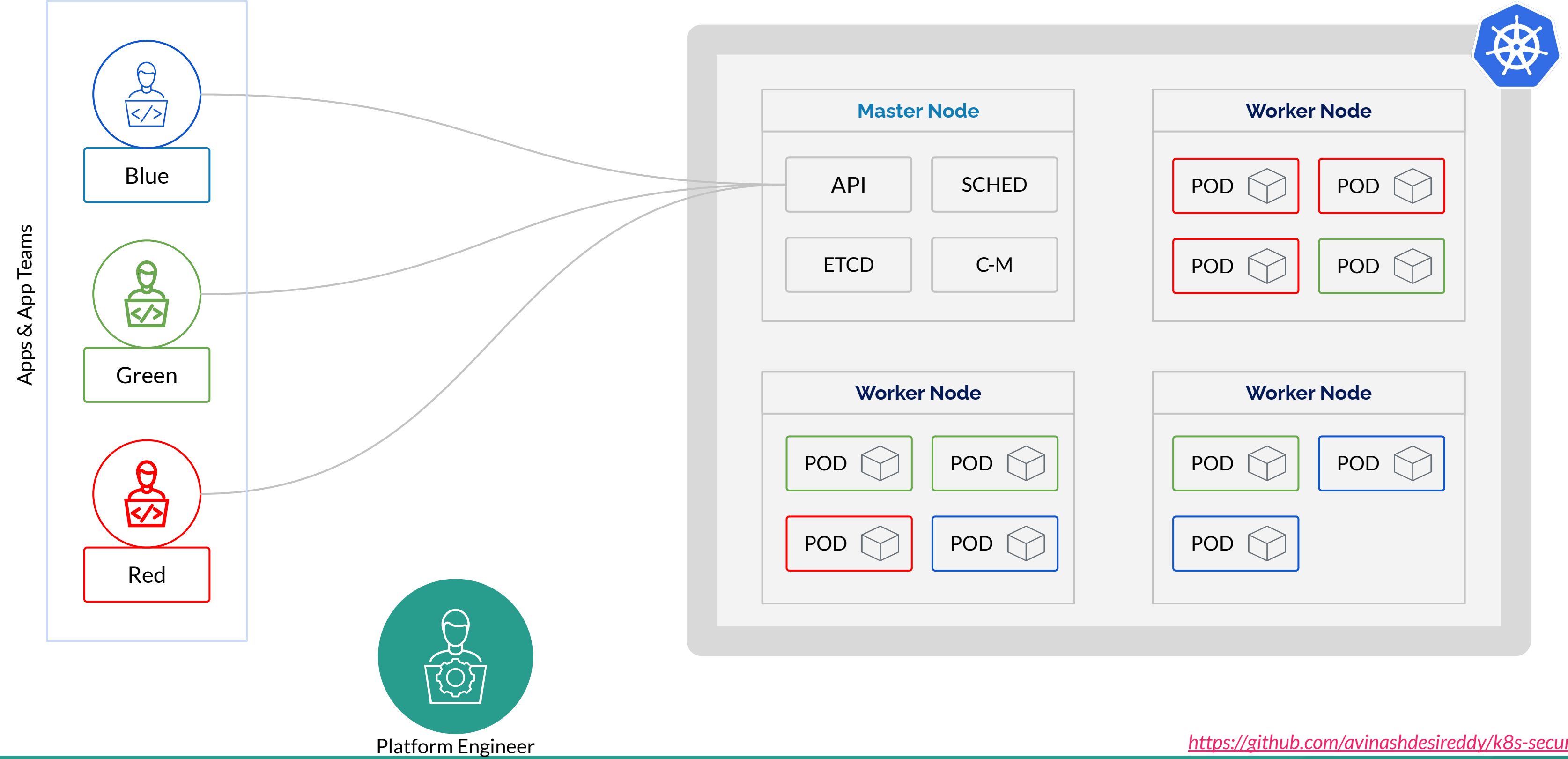
Kubernetes IDE

Access the cluster

/avinashdesireddy/k8s-security-workshop.git

https://github.com/avinashdesireddy/k8s-security-workshop.git

Happy Users!!!



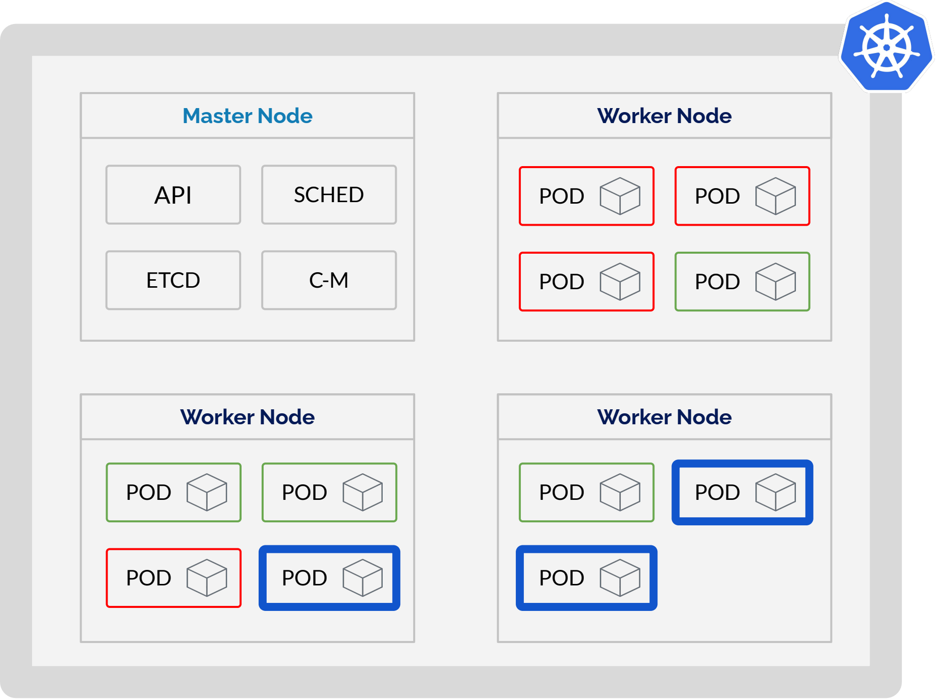
https://github.com/avinashdesireddy/k8s-security-workshop.git

Scenario #2

All of a sudden, Pods belong to ***App Blue*** *started* ***consuming a lot of memory*** *in the cluster.*

How do we fix it?

Resource Limits



https://github.com/avinashdesireddy/k8s-security-workshop.git

Scenario #2: Coordinating changes

- Identify Application Owner
- Ask Owner of App Blue to specify Memory & CPU Limits on Containers
- Configure Resource Quota & Limits on Namespaces

https://github.com/avinashdesireddy/k8s-security-workshop.git

Scenario #2: Challenges

- How can we enforce these across all the applications in the cluster?
  - Reach out to multiple application to make changes?
  - Define Best Practices?
  - Monthly Audits?

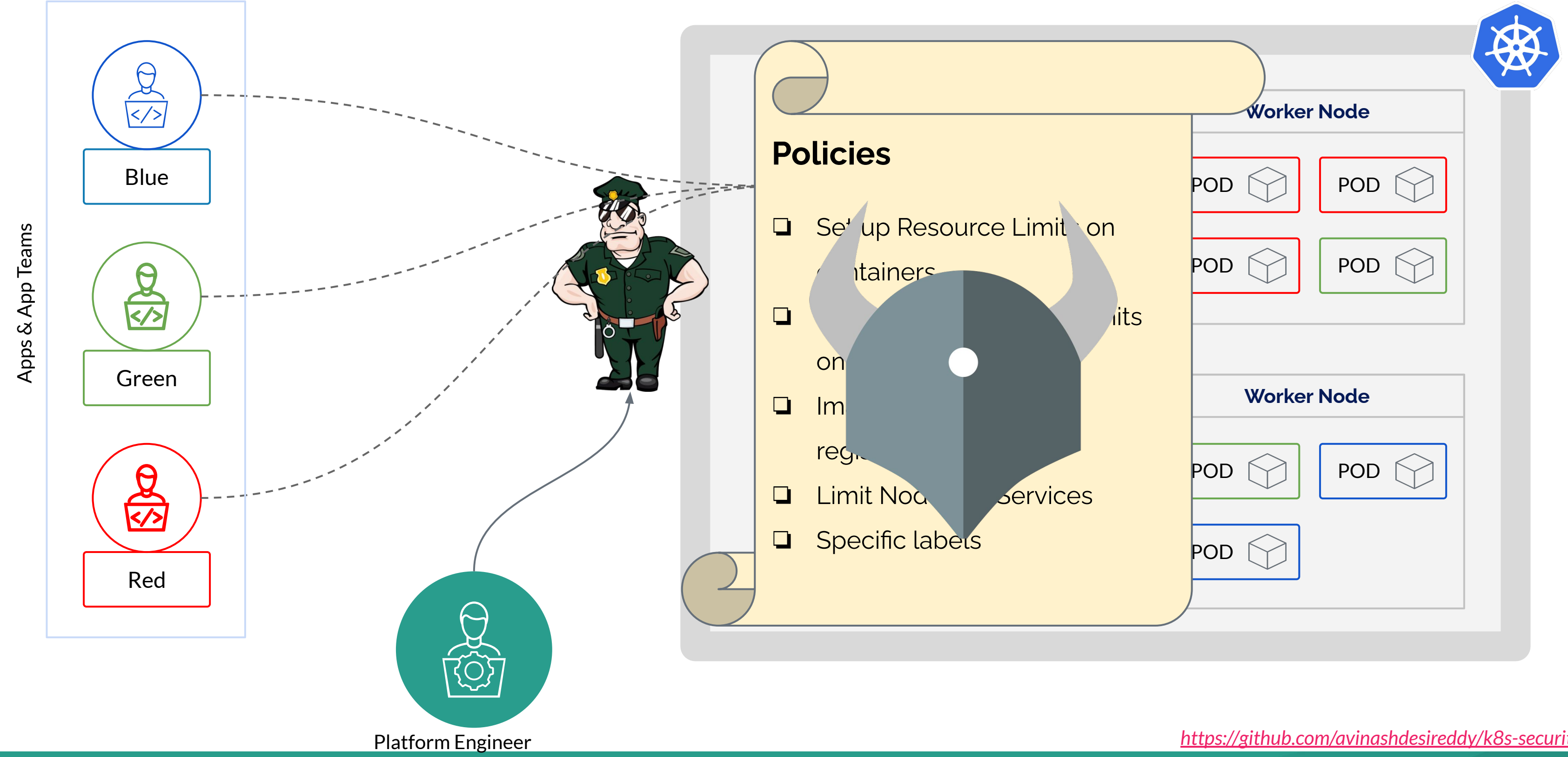
https://github.com/avinashdesireddy/k8s-security-workshop.git

slido

Do you find it a challenge while agreeing on Cluster Best Practices with App Teams?

ⓘ Start presenting to display the poll results on this slide.

Scenario #2 - Policy Enforcement



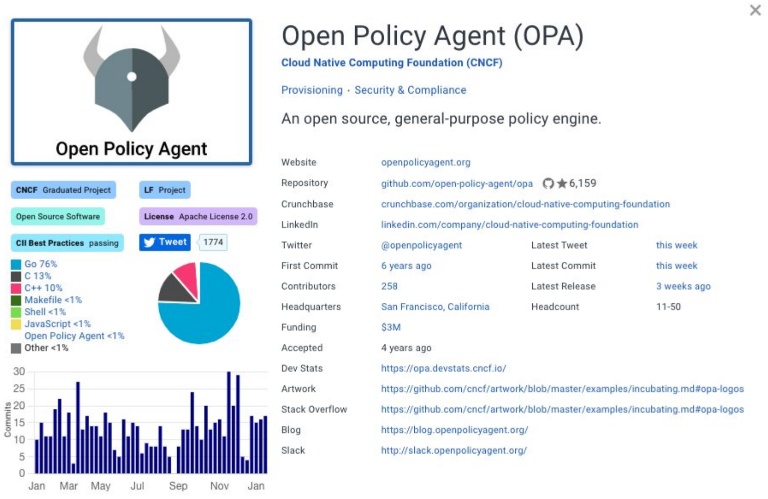
Policies

- ❑ Se up Resource Limit on ntainers
- ❑ its on
- ❑ Im reg
- ❑ Limit Nod Services
- ❑ Specific labels

https://github.com/avinashdesireddy/k8s-security-workshop.git

Open Policy Agent (OPA)

- CNCF Graduated
- General Purpose Policy Engine
- Empowers admins with more CONTROL over the system
- REGO Language
- Gatekeeper → Admission Controller implementation of OPA



Open Policy Agent (OPA)

Cloud Native Computing Foundation (CNCF)

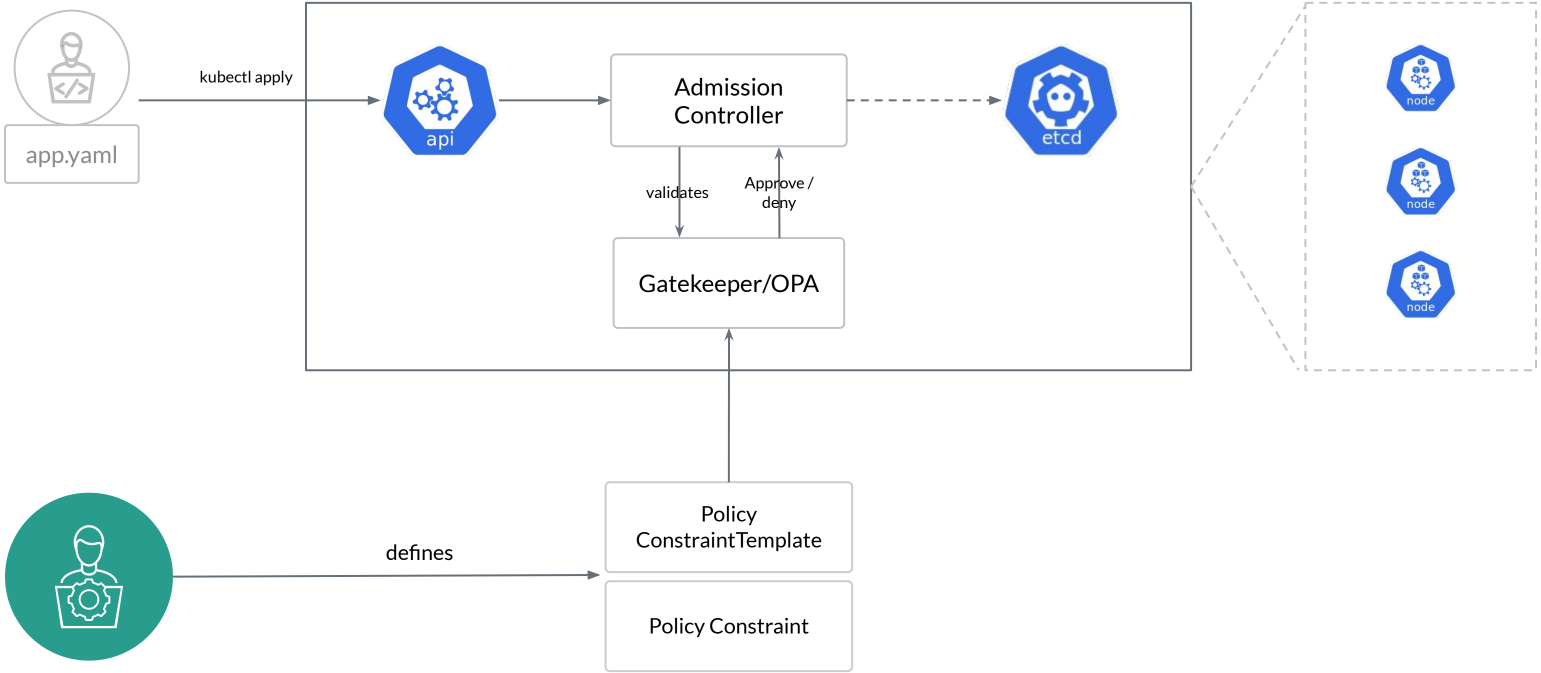
Provisioning · Security & Compliance

An open source, general-purpose policy engine.

Website	openpolicyagent.org		
Repository	github.com/open-policy-agent/opa ★6,159		
Crunchbase	crunchbase.com/organization/cloud-native-computing-foundation		
LinkedIn	linkedin.com/company/cloud-native-computing-foundation		
Twitter	@openpolicyagent	Latest Tweet	this week
First Commit	6 years ago	Latest Commit	this week
Contributors	258	Latest Release	3 weeks ago
Headquarters	San Francisco, California	Headcount	11-50
Funding	$3M		
Accepted	4 years ago		
Dev Stats	https://opa.devstats.cncf.io/		
Artwork	https://github.com/cncf/artwork/blob/master/examples/incubating.md#opa-logos		
Stack Overflow	https://github.com/cncf/artwork/blob/master/examples/incubating.md#opa-logos		
Blog	https://blog.openpolicyagent.org/		
Slack	http://slack.openpolicyagent.org/		

https://github.com/avinashdesireddy/k8s-security-workshop.git

OPA in Kubernetes



https://github.com/avinashdesireddy/k8s-security-workshop.git

Demo: OPA

- Restrict NodePort Usage
- Enforce Container Resource Limits

Environment

Mirantis Kubernetes Engine

1 Manager, 3 Worker

Version - 1.21.3

LENS

Kubernetes IDE

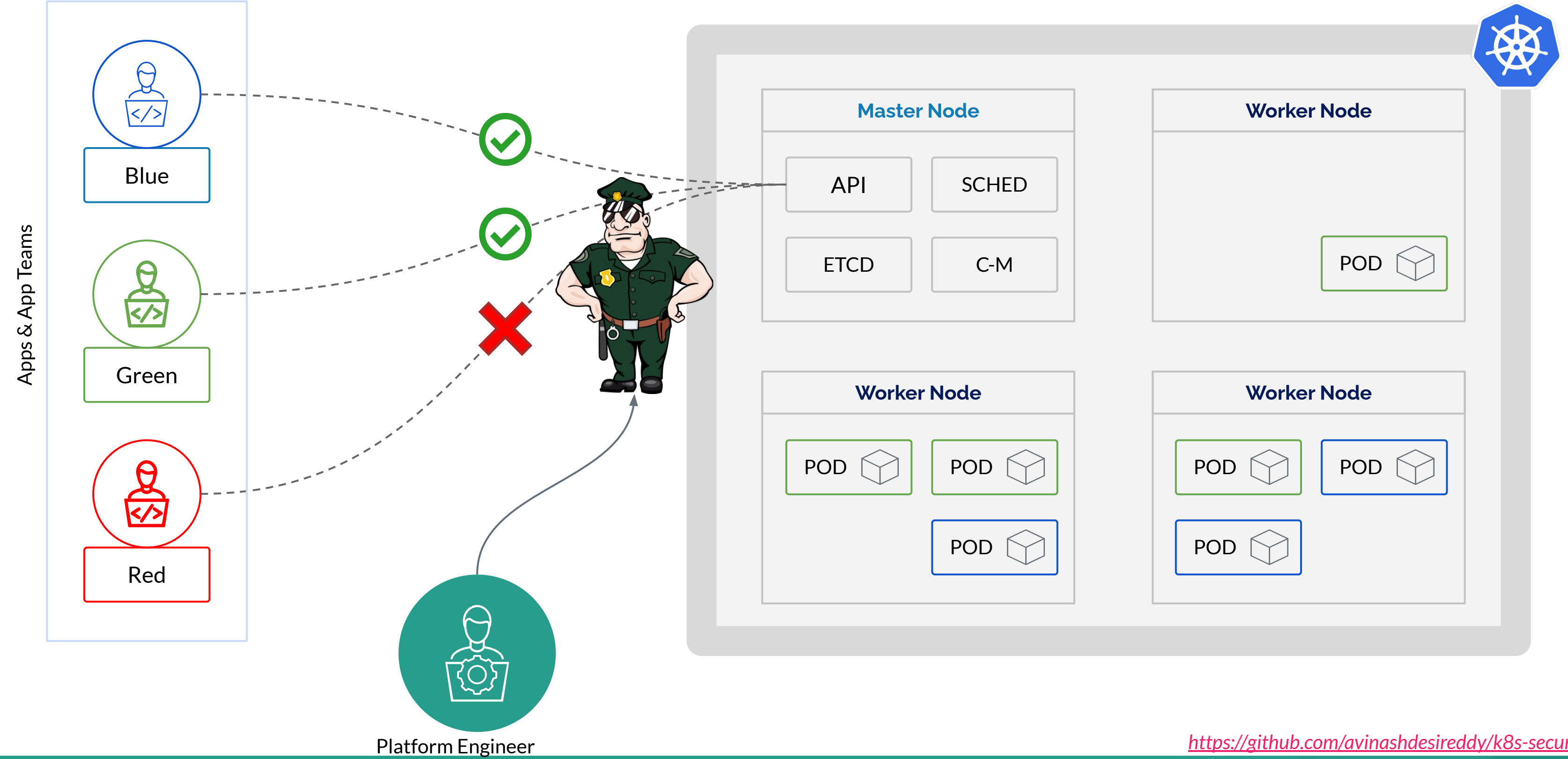
Access the cluster

Open Policy Agent /

Gatekeeper

/avinashdesireddy/k8s-security-workshop.git

Happy Users... Happy Cluster!!!



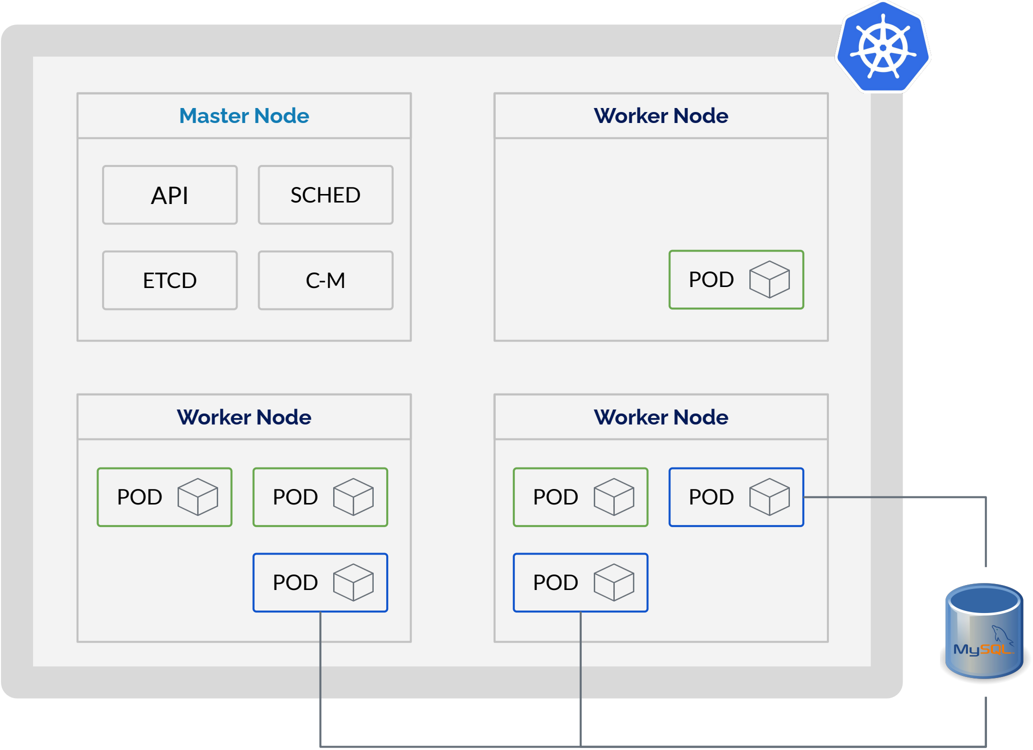
https://github.com/avinashdesireddy/k8s-security-workshop.git

Scenario #3: Network Security

*New features are added to **App Blue**, the pods must connect to an external MySql DB and to an exposed API in **Green App Pod***

How do we control Network Traffic to/from Pods?

Network Policies

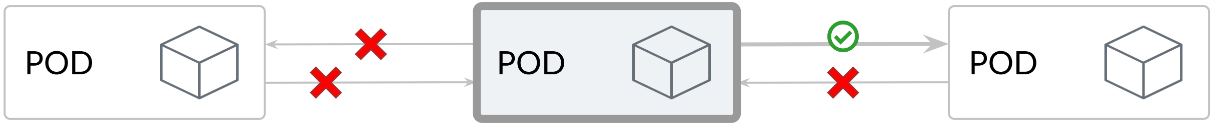


https://github.com/avinashdesireddy/k8s-security-workshop.git

Network Policy

- Control Traffic to/from pods
- Traffic between pods are non-Isolated
- Namespace scoped
- Can be defined based on -
  - Pod, Namespace or IP Range

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
 name: test-network-policy
 namespace: default
spec:
 podSelector:
   matchLabels:
     role: db
 policyTypes:
 - Ingress
 - Egress
 ingress:
 - from:
   - ipBlock:
       cidr: 172.17.0.0/16
       except:
       - 172.17.1.0/24
   - namespaceSelector:
       matchLabels:
         project: myproject
   - podSelector:
       matchLabels:
         role: frontend
   ports:
   - protocol: TCP
     port: 6379
 egress:
 - to:
   - ipBlock:
       cidr: 10.0.0.0/24
   ports:
   - protocol: TCP
     port: 5978
```



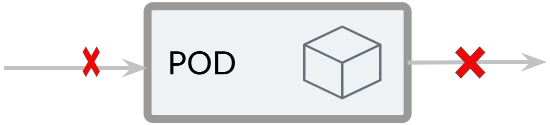
slido

Who are using Network Policies?

ⓘ Start presenting to display the poll results on this slide.

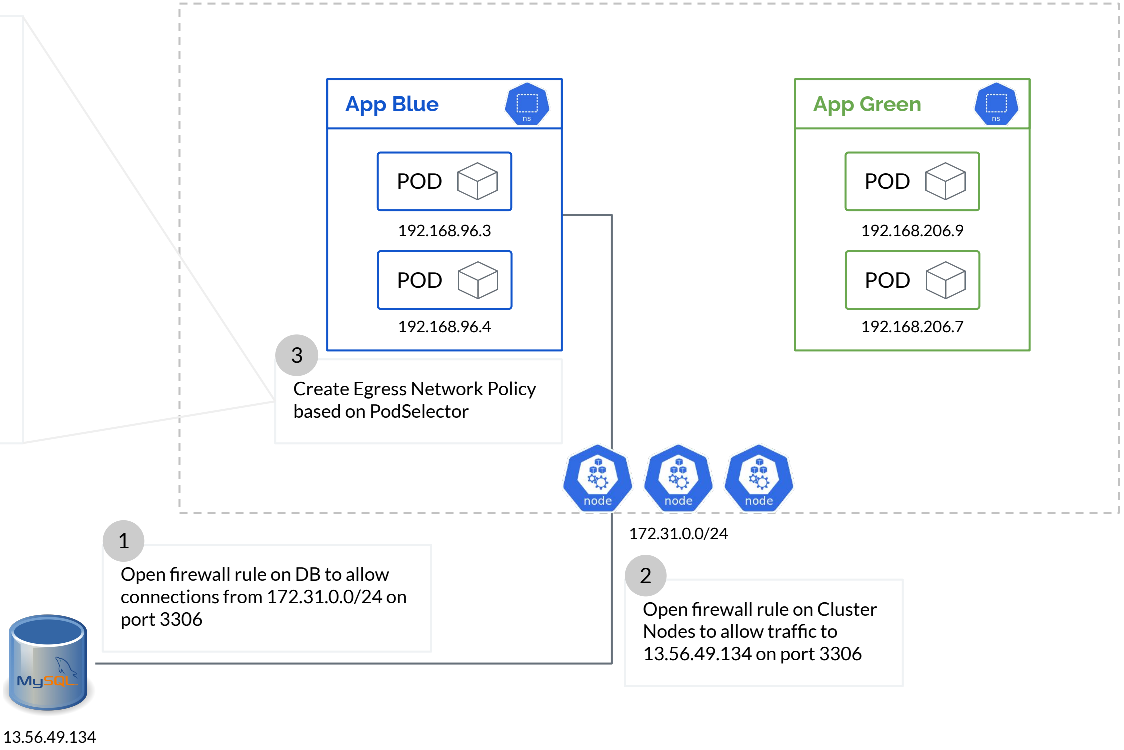
Default Deny Policy

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
 name: default-deny-all
spec:
 podSelector: {}
 policyTypes:
 - Ingress
 - Egress
 egress:
 - to:
   - namespaceSelector:
       matchLabels:
         kubernetes.io/metadata.name: kube-system
     podSelector:
       matchLabels:
         k8s-app: kube-dns
   ports:
   - protocol: UDP
     port: 53
   - protocol: TCP
     port: 53
```



Scenario #3: App Blue connecting to MySQL

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
 name: mysql-port-egress
spec:
 podSelector:
   matchLabels:
     app: blue
     backend: mysql
 policyTypes:
 - Egress
 egress:
 - to:
   - ipBlock:
       cidr: 13.56.49.134/32
   ports:
   - protocol: TCP
     port: 3306
```



3 Create Egress Network Policy based on PodSelector

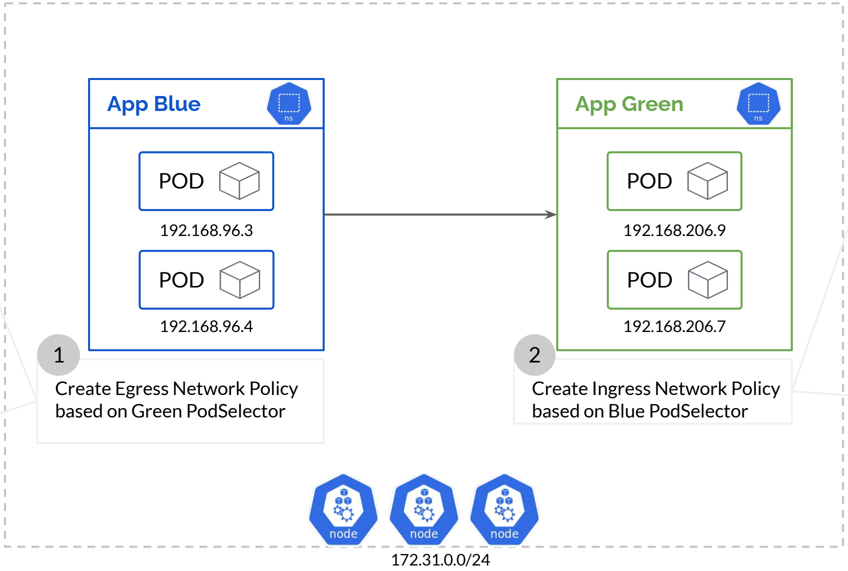
1 Open firewall rule on DB to allow connections from 172.31.0.0/24 on port 3306

2 Open firewall rule on Cluster Nodes to allow traffic to 13.56.49.134 on port 3306

https://github.com/avinashdesireddy/k8s-security-workshop.git

Scenario #3: App Blue connecting to App Green

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
 name: to-green-egress
 namespace: blue
spec:
 podSelector:
   matchLabels:
     app: blue
 policyTypes:
 - Egress
 egress:
 - to:
   - podSelector:
       matchLabels:
         app: green
   ports:
   - protocol: TCP
     port: 8080
```



1 Create Egress Network Policy based on Green PodSelector

2 Create Ingress Network Policy based on Blue PodSelector

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
 name: from-blue-ingress
 namespace: green
spec:
 podSelector:
   matchLabels:
     app: green
 policyTypes:
 - Ingress
 ingress:
 - from:
   - podSelector:
       matchLabels:
         app: blue
```

https://github.com/avinashdesireddy/k8s-security-workshop.git

Demo: Network Policies

- Create Default Network Policies
- Allow access for “Blue” App to MySQL on Port 3306
- Allow access for “Blue” App to access “Green” Application’s API

Environment

Mirantis Kubernetes Engine

1 Manager, 3 Worker

Version - **1.21.3**

Kubernetes IDE

Access the cluster

Kubernetes Network Policies

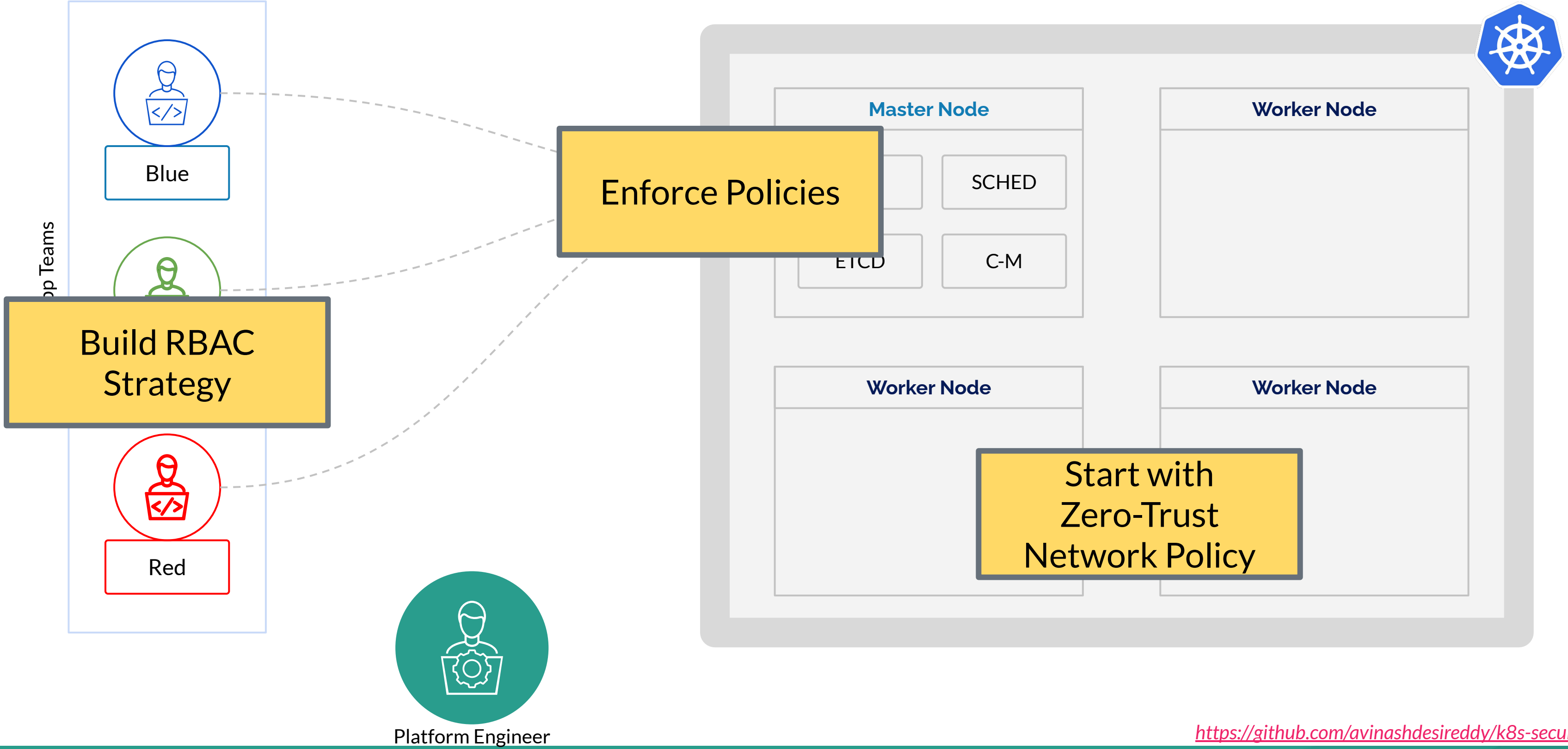
/avinashdesireddy/k8s-security-workshop.git

CNIs with Network Policy Support

- Weave
- Calico
- Cilium
- Kube-router
- Istio

https://github.com/avinashdesireddy/k8s-security-workshop.git

Takeaways...



Enforce Policies

Build RBAC Strategy

Start with Zero-Trust Network Policy

https://github.com/avinashdesireddy/k8s-security-workshop.git

Thank you!

/avinashdesiredd /anooplive

https://github.com/avinashdesireddy/k8s-security-workshop.git