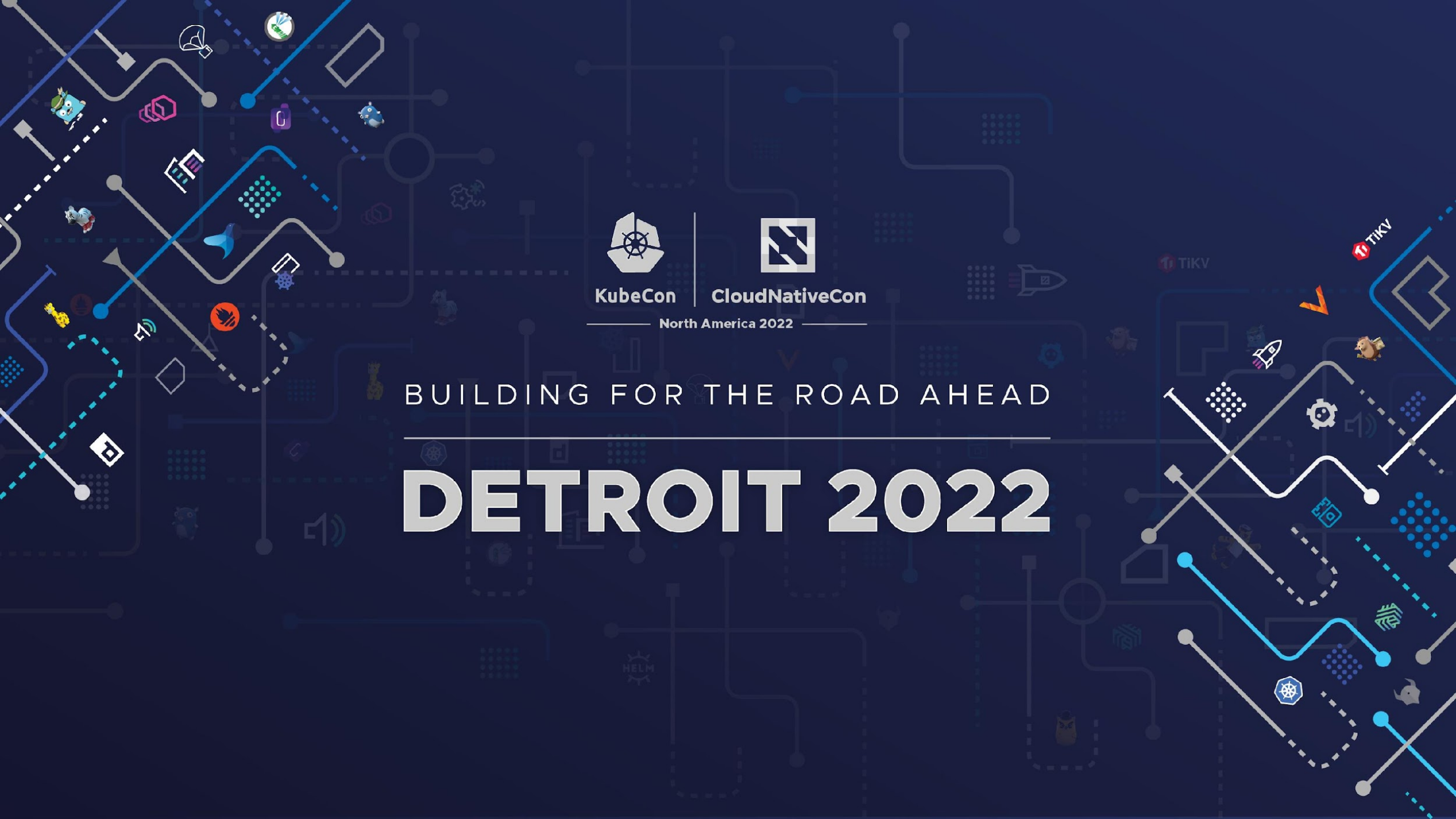

KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV
HELM

Tutorial: Reducing the Sticker Price of Kubernetes Security

KubeCon CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

KubeCon CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

October 24-28, 2021

Pushkar Joglekar
Tooling sub-project maintainer,
Kubernetes SIG Security

$whoami

@PuDiJoglekar

- Kubernetes SIG Security - Tooling Lead
- CNCF TAG Security - Tech Lead
- Associate Kubernetes Security Response Committee Member
- Formerly VMware Tanzu, Staff Security Engineer
- Based in San Francisco Bay Area
- Loves Cycling, Chai, Curries, Camera and Cricket :)
- More about me: https://pushkarj.github.io

Before we Begin

@PuDiJoglekar

- This is a 90 minute tutorial:
  - 10 mins: Intro & Prerequisites
  - 20 mins: First task (5 mins break)
  - 25 mins: Second task (5 mins break)
  - 30 mins: Third task (5 mins break)
  - 5 mins: Wrap up & Further Reading!

Pulse Check

@PuDiJoglekar

Mac OS:

- M1 (use *darwin-arm64* binaries)
- Intel (use *darwin-amd64* binaries)

Windows (use *windows* binaries)

Linux (use *linux* binaries)

Note

@PuDiJoglekar

If Mac or Windows: you will need Docker Desktop.

Choice of your lab machine is up to you but…

Please DO **NOT** RUN this tutorial on **Shared / Corporate / Production** machines

Introduction

@PuDiJoglekar

- Secure by Design
- Built-in v/s Bolted-on
- Transparent
- It Just Works! User Experience
- <whispers> *Low Cost* </whispers>

Focus

@PuDiJoglekar

Built-in Kubernetes Security Features!

Tasks

@PuDiJoglekar

- **First**: *Verify* “Signed” Container Images
- **Second**: *Enforce* “Baseline” Pod Security
- **Third**: *Enable* “RuntimeDefault” SecComp Filter

Prerequisite Knowledge

@PuDiJoglekar

- Unix Commands e.g.
  - curl
  - file redirects
  - file editors
- Kubernetes Basics e.g.
  - Namespaces
  - Pods
  - Images

Prerequisite Tooling - Pulse Check

@PuDiJoglekar

- git
- go
- kind
- docker
- kubectl
- cosign
- curl

Asking Questions

@PuDiJoglekar

- Ask Early and Often
- Create GitHub Issues in the tutorial repo:
  - Helps with debugging and follow up
- If needed, use the 5 min breaks too for questions
- Will be hanging around for a few mins after the session

Let’s start

@PuDiJoglekar

```
git clone https://github.com/PushkarJ/kccncna-22-tutorial.git
```

Demo

First Task:

Verify “Signed” Images

Sign and Verify

@PuDiJoglekar

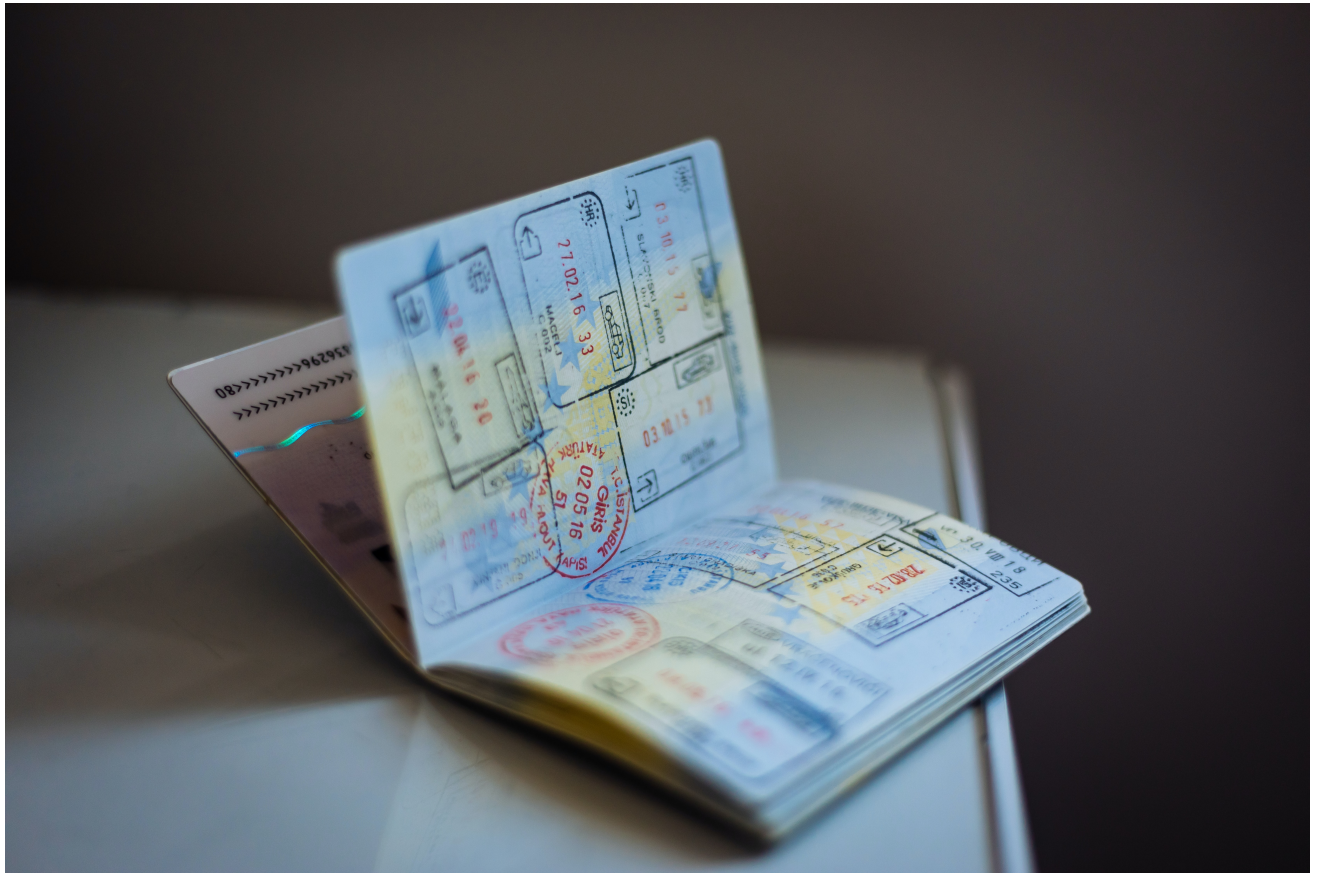


Photo by Global Residence Index on Unsplash

Sign and Verify

@PuDiJoglekar

- Signed by makers: Private Key
- Verified by end users: Public Key
- Tells you where an artifact came from
- Forging of signature is dependent on access to signing key
  - Should be kept private

Get All Images

@PuDiJoglekar

```
curl -Ls https://sbom.k8s.io/$(curl -Ls https://dl.k8s.io/
release/latest.txt)/release | grep 'PackageName:
registry.k8s.io/' | awk '{print $2}' > images.txt

input=images.txt
while IFS= read -r image
do
  COSIGN_EXPERIMENTAL=1 cosign verify "$image"
done < "$input"
```

Verify One image

@PuDiJoglekar

```
COSIGN_EXPERIMENTAL=1 cosign verify \
registry.k8s.io/kube-apiserver-amd64:v1.25.2
```

Verify All Images

@PuDiJoglekar

```
input=images.txt
while IFS= read -r image
do
  COSIGN_EXPERIMENTAL=1 cosign verify "$image"
done < "$input"
```

Demo

5 mins…
Break / Catch Up

Stretch, Drink Water, Relax your eyes

Second Task:

Enforce “Baseline” Pod Security

PSP: deprecated and removed

@PuDiJoglekar

Pod Security Policy (PSP):

* Deprecated (since v1.21)

* Removed (since v1.25)

For migration from Pod Security Policy to Pod Security Admission: https://sched.co/182Jx (happened Thursday)

Pod Security Standards

@PuDiJoglekar

Restricted

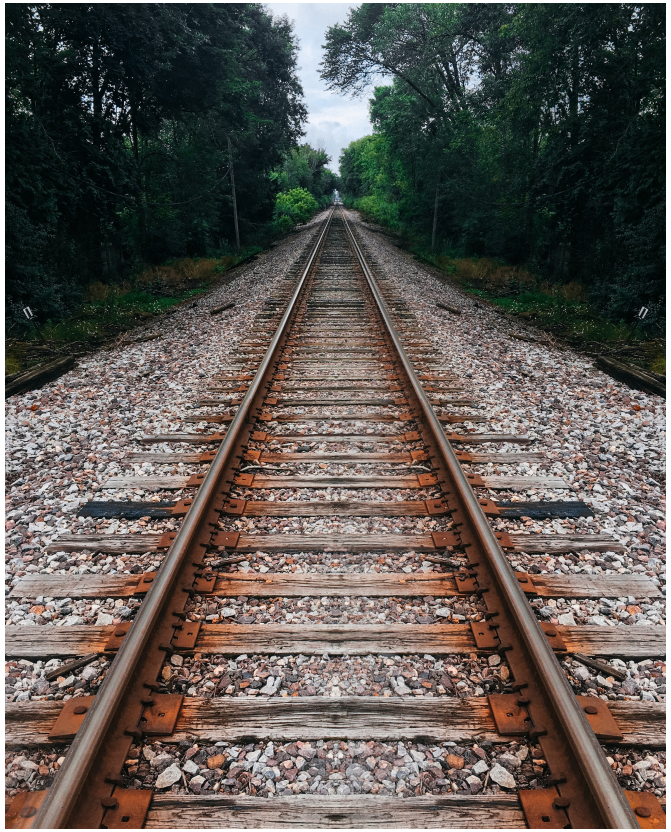


Photo by Tom Barrett on Unsplash

Baseline

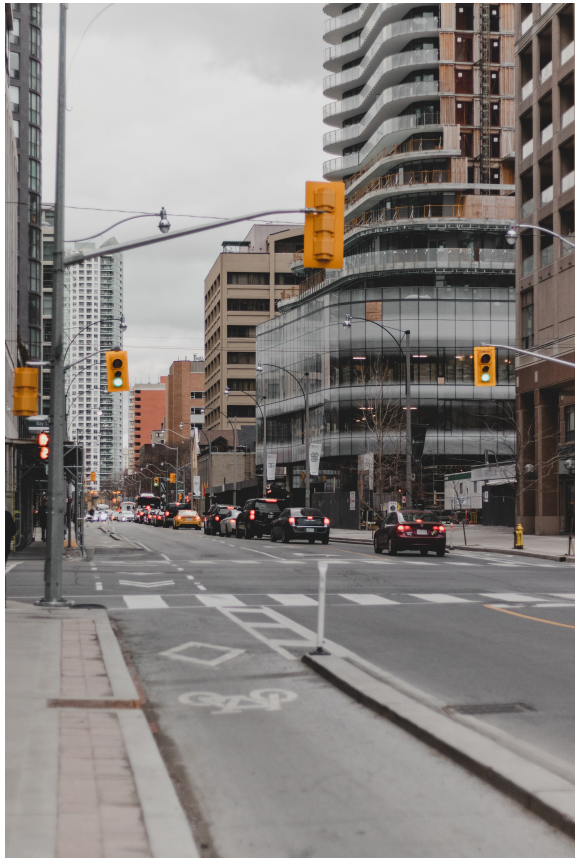


Photo by Omar Quazi on Unsplash

Privileged

Photo by Connor McSheffrey on Unsplash

Pod Security Standards

@PuDiJoglekar

* Opinionated set of Pod Security Configurations
* From lowest to highest risk:
    * Restricted, Baseline, Privileged

Pod Security Admission

@PuDiJoglekar

Lightweight built-in pod security (standards) admission controller

- No Extra installation steps needed
- Feature *Enabled* by default but…
- No PSS is *configured* by default
- Two ways to configure: *cluster* level and *namespace* level

Setting up Cluster Level Pod Security

@PuDiJoglekar

```
apiVersion: apiserver.config.k8s.io/v1
kind: AdmissionConfiguration
plugins:
- name: PodSecurity
  configuration:
    apiVersion: pod-security.admission.config.k8s.io/v1
    kind: PodSecurityConfiguration
    defaults:
      enforce: "baseline"
      enforce-version: "latest"
      audit: "restricted"
      audit-version: "latest"
      warn: "restricted"
      warn-version: "latest"
    exemptions:
      usernames: []
      runtimeClasses: []
      namespaces: [kube-system]
```

Enforce, Warn and Audit Modes

@PuDiJoglekar

Enforce (blocking): Pod Creation Fails if the configured Pod Security Standards are unmet

Warn (non-blocking): Warns the User about violations of configured Pod Security Standards but creates the pod

Audit (non-blocking): Violations of configured Pod Security Standards are logged in Kubernetes API Audit logs

Pod Security Exemptions

@PuDiJoglekar

Covers special circumstances:

- Where the pods, users, runtimes can not conform to cluster level pod security
- We will use namespace exemption in this tutorial

Creating a Cluster with Pod Security

@PuDiJoglekar

```
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
  kubeadmConfigPatches:
  - |
    kind: ClusterConfiguration
    apiServer:
        extraArgs:
          admission-control-config-file: /etc/config/
cluster-level-pss.yaml

</snip>
```

Minimal Pod Spec

@PuDiJoglekar

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
    - image: nginx
      name: nginx
      ports:
        - containerPort: 80
```

Minimal Pod Spec - Warnings

@PuDiJoglekar

Warning: would violate PodSecurity "restricted:latest":

- allowPrivilegeEscalation != false (container "nginx" must set *securityContext.allowPrivilegeEscalation=false*),
- unrestricted capabilities (container "nginx" must set *securityContext.capabilities.drop=["ALL"]*),
- runAsNonRoot != true (pod or container "nginx" must set securityContext.runAsNonRoot=true),
- seccompProfile (pod or container "nginx" must set *securityContext.seccompProfile.type* to "RuntimeDefault" or "Localhost")

Test that passes “restricted” PSS

@PuDiJoglekar

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-r-pss
spec:
  containers:
    - image: nginx
      name: nginx-r-pss
      securityContext:
        runAsNonRoot: true
        seccompProfile:
          type: RuntimeDefault
        allowPrivilegeEscalation: false
        capabilities:
          drop: ["ALL"]
      ports:
        - containerPort: 80
```

Demo

5 mins…
Break / Catch Up

Stretch, Drink Water, Relax your eyes

Third Task:

Enable “RuntimeDefault” SecComp Profile

What is SecComp?

@PuDiJoglekar

- It’s a system call filter
- One of the fundamental isolation technologies used in containers
- Blocking system calls that would break isolation boundaries

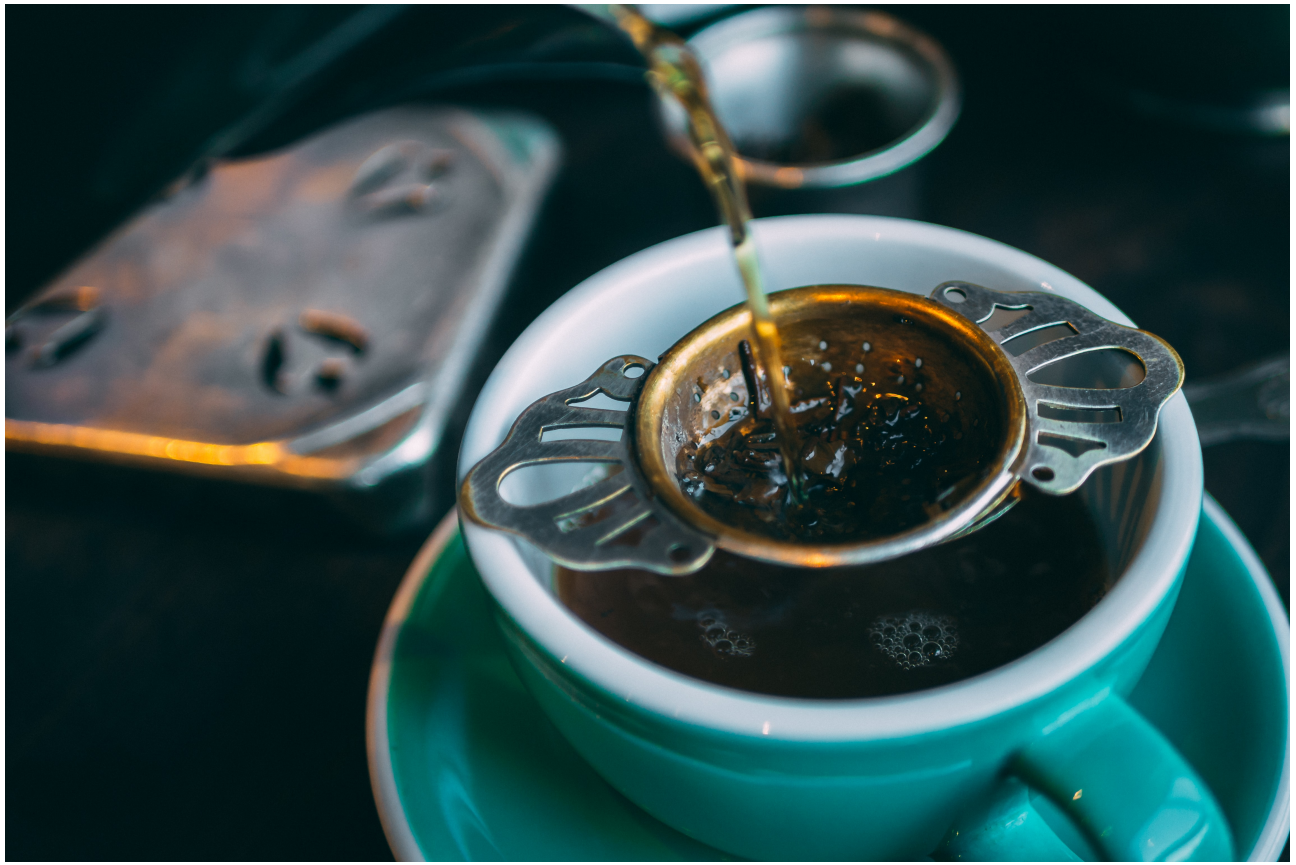


Photo by Matt Seymour on Unsplash

What is SecComp?

@PuDiJoglekar

How many system calls exist in a typical linux system?

300 and counting

What is RuntimeDefault?

@PuDiJoglekar

Tells Kubernetes to use the *seccomp* profile of the underlying container runtime e.g. containerd, cri-o

Enabling SecComp for your Cluster

@PuDiJoglekar

Three choices:

- `Unconfined` i.e. bypass container runtime seccomp profile: Default for many years in Kubernetes
- Stricter profiles:
  - `RuntimeDefault`: inherit container runtime default profile
  - `LocalHost`: custom seccomp profile for your pod

Default SecComp Profile

@PuDiJoglekar

Containerd: https://github.com/containerd/containerd/blob/main/contrib/seccomp/seccomp_default.go#L56-L479

CRI-O: https://github.com/cri-o/cri-o/blob/main/vendor/github.com/containers/common/pkg/seccomp/default_linux.go#L45-L887

Docker: https://docs.docker.com/engine/security/seccomp/

Customizing SecComp ~~Pizza~~ Profile

@RuDiLoglekar

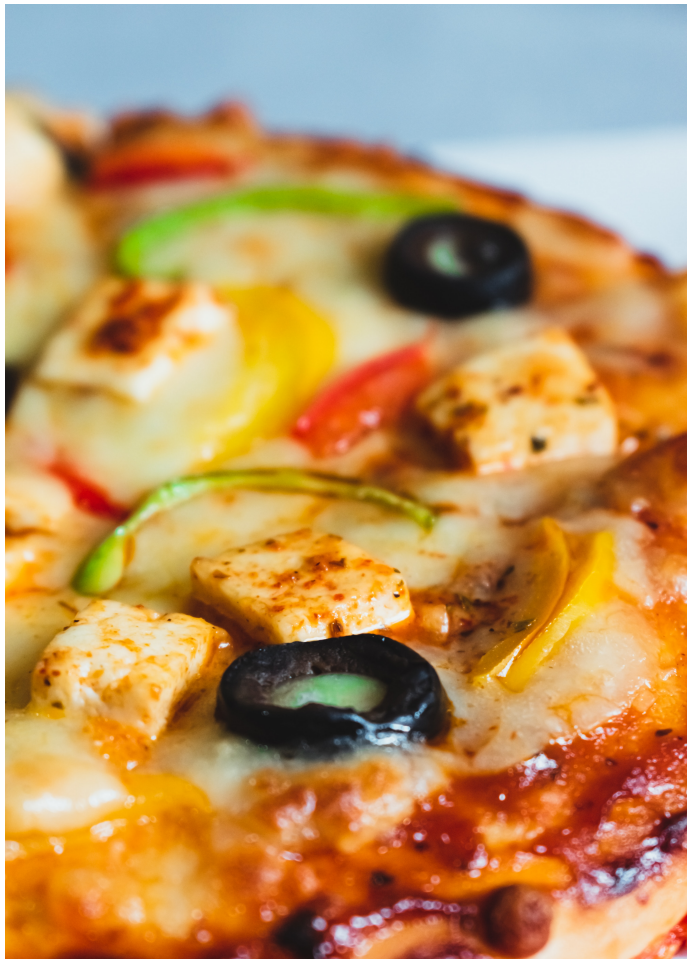


Photo by ABHISHEK HAJARE on Unsplash

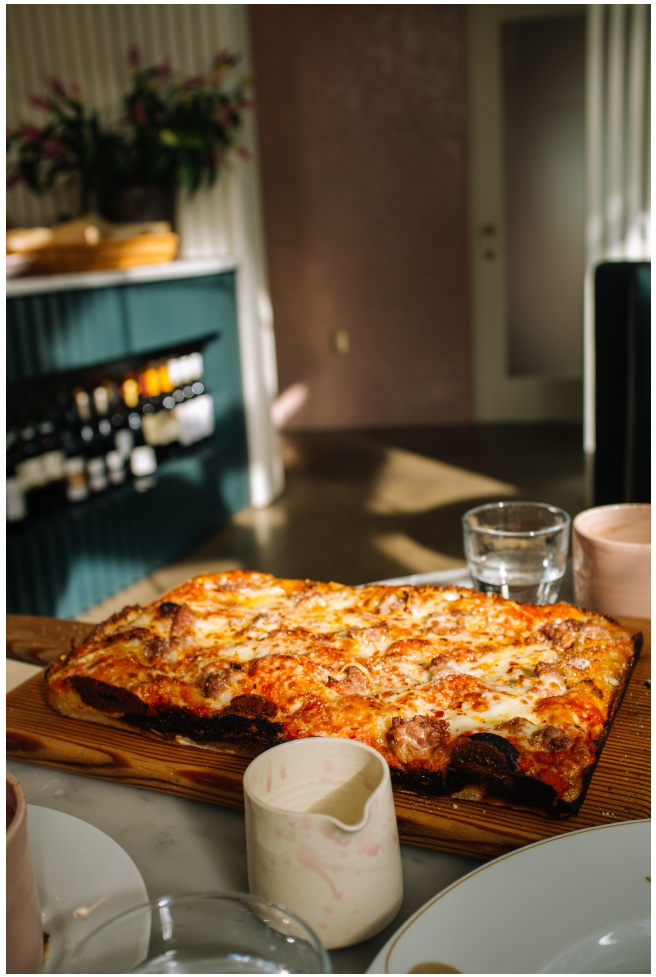


Photo by H Liu on Unsplash

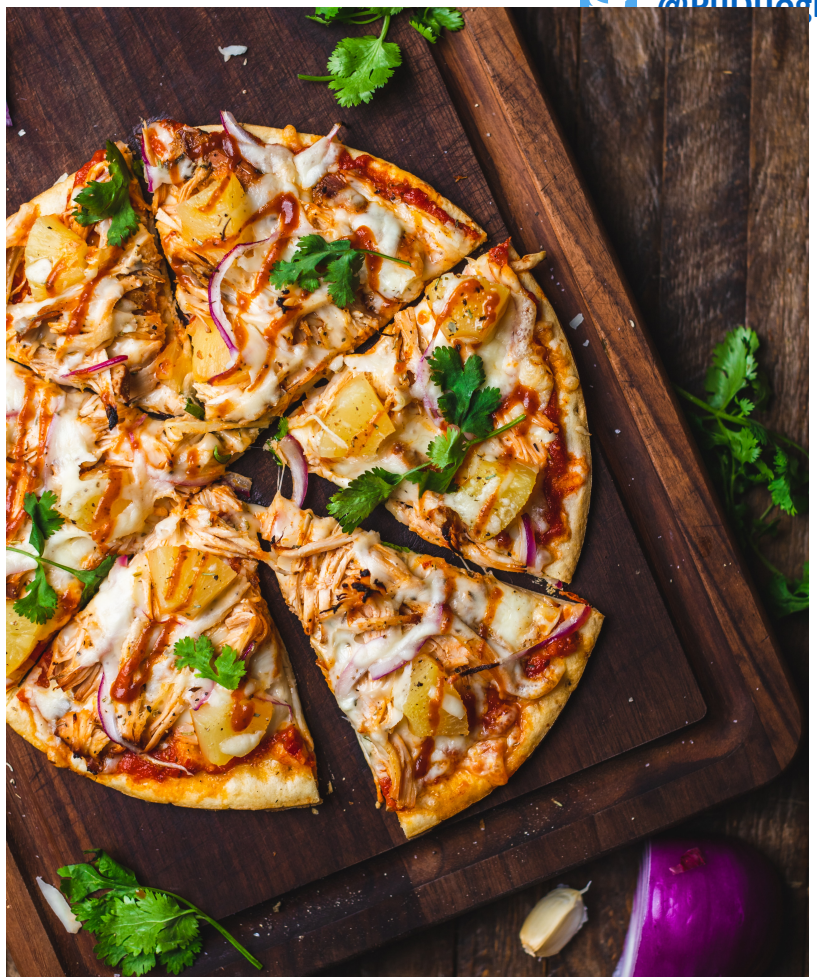


Photo by Chad Montano on Unsplash

Testing Isolation Boundaries

@PuDiJoglekar

- Blocking system calls limits exposure to kernel and thus limits attack surface
- Change (of date and time) in one container should not allow change of date and time for local containers
- This is why SYS_TIME system call and its corresponding capability is blocked in default *seccomp* profile of container runtimes

Set SecComp Default to RuntimeDefault

@PuDiJoglekar

```
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
featureGates:
  SeccompDefault: true
nodes:
  - role: control-plane
    image: kindest/
node:v1.25.0@sha256:6e0f9005eba4010e364aa1bb25c8d7c64f050f744258eb68c4eb40c284c3c0dd
    kubeadmConfigPatches:
      - |
        kind: JoinConfiguration
        nodeRegistration:
          kubeletExtraArgs:
            seccomp-default: "true"
  - role: worker
    image: kindest/
node:v1.25.0@sha256:6e0f9005eba4010e364aa1bb25c8d7c64f050f744258eb68c4eb40c284c3c0dd
    kubeadmConfigPatches:
      - |
        kind: JoinConfiguration
        nodeRegistration:
          kubeletExtraArgs:
            seccomp-default: "true"
```

With Seccomp: Change of Date and Time

@PuDiJoglekar

```
kubecon:seccomp tutorial$ kubectl exec --stdin --tty
forever-asleep -- /bin/bash
root@forever-asleep:/# date 010709342000
date: cannot set date: Operation not permitted
Fri Jan  7 09:34:00 UTC 2000


root@forever-asleep:/# date
Thu Oct  6 03:36:24 UTC 2022
```

Without Seccomp: Change of Date and Time

@PuDiJoglekar

```
kubecon:seccomp tutorial$ kubectl exec --stdin --
tty forever-asleep -- /bin/bash
root@forever-asleep:/# date 010709342000

Fri Jan  7 09:34:00 UTC 2000


root@forever-asleep-again:/# date
Fri Jan  7 09:34:00 UTC 2000
```

Demo

5 mins…
Break / Catch Up

Stretch, Drink Water, Relax your eyes

Bringing it all together!

@PuDiJoglekar

- Verify Images
- Create clusters with:
  - SecComp set as RuntimeDefault
  - Baseline Pod Security Standard Enforced

Demo:
May be try this at home?

Further Reading!

@PuDiJoglekar

All part of official Kubernetes Documentation

- https://kubernetes.io/docs/tasks/administer-cluster/verify-signed-images/
- https://kubernetes.io/docs/tutorials/security/ns-level-pss/
- https://kubernetes.io/docs/tutorials/security/cluster-level-pss/
- https://kubernetes.io/docs/tutorials/security/seccomp/

Questions?

KubeCon CloudNativeCon North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

This is a safe space 😊

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

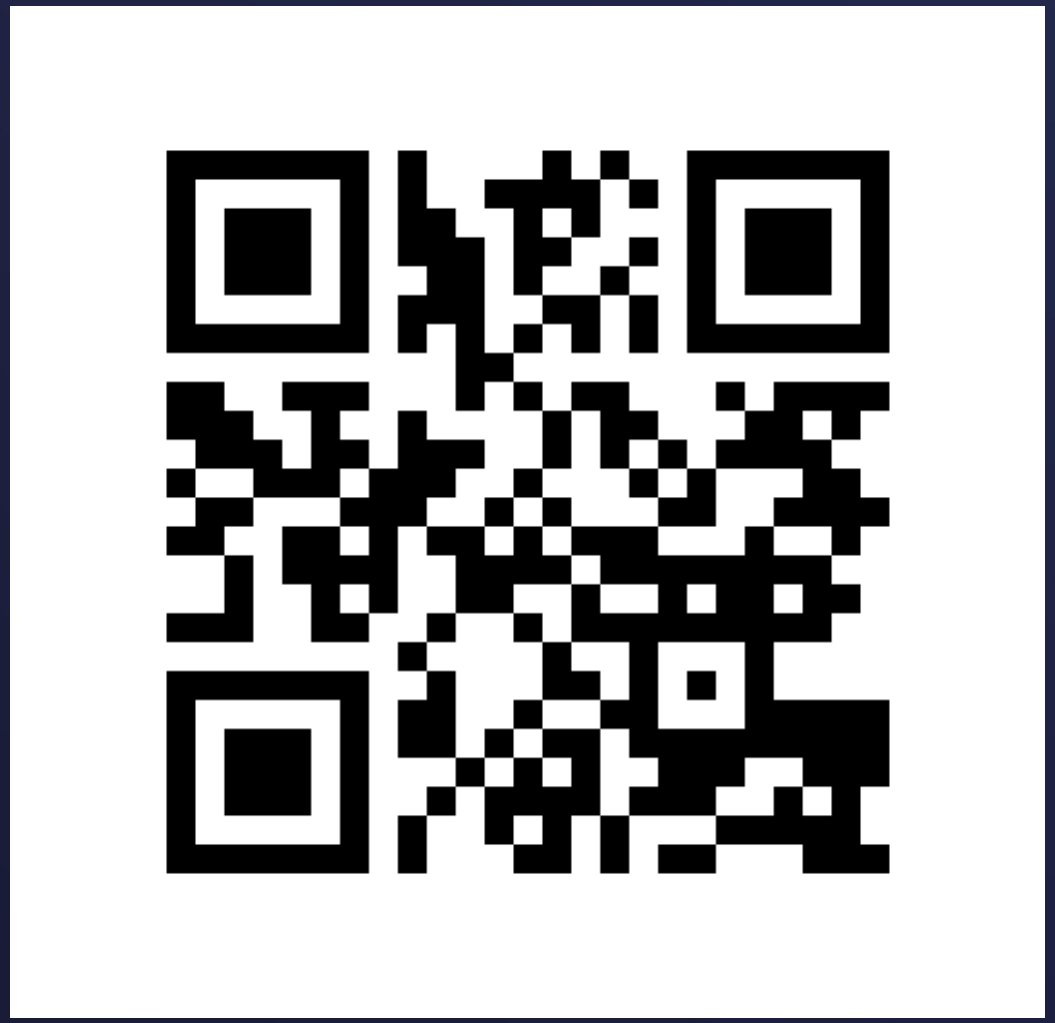
Find me on Twitter: @PuDiJoglekar

Thank You!

KubeCon | CloudNativeCon
North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022



Please scan the QR Code above to leave feedback on this session