Kumar **Ashwin**

Common Misconfigurations In Your Kubernetes Cluster And What Can We Do About It?

About Me

Kumar **Ashwin**

Security Consultant & Program Manager at Payatu

Interested and constantly learning in Web, Cloud and Cloud Native Domains

Reach me out at **0xCardinal.com**

What are we going to cover?

- Wh- Kubernetes?
  - What?
  - Why?
- Common Issues or Misconfigurations in Kubernetes
  - Namespace
  - Over-permissive containers
  - Secret Management Issues
  - Insecure Defaults
  - ...
- What can you do about these misconfigurations?
- Next?

Wh- Kubernetes?

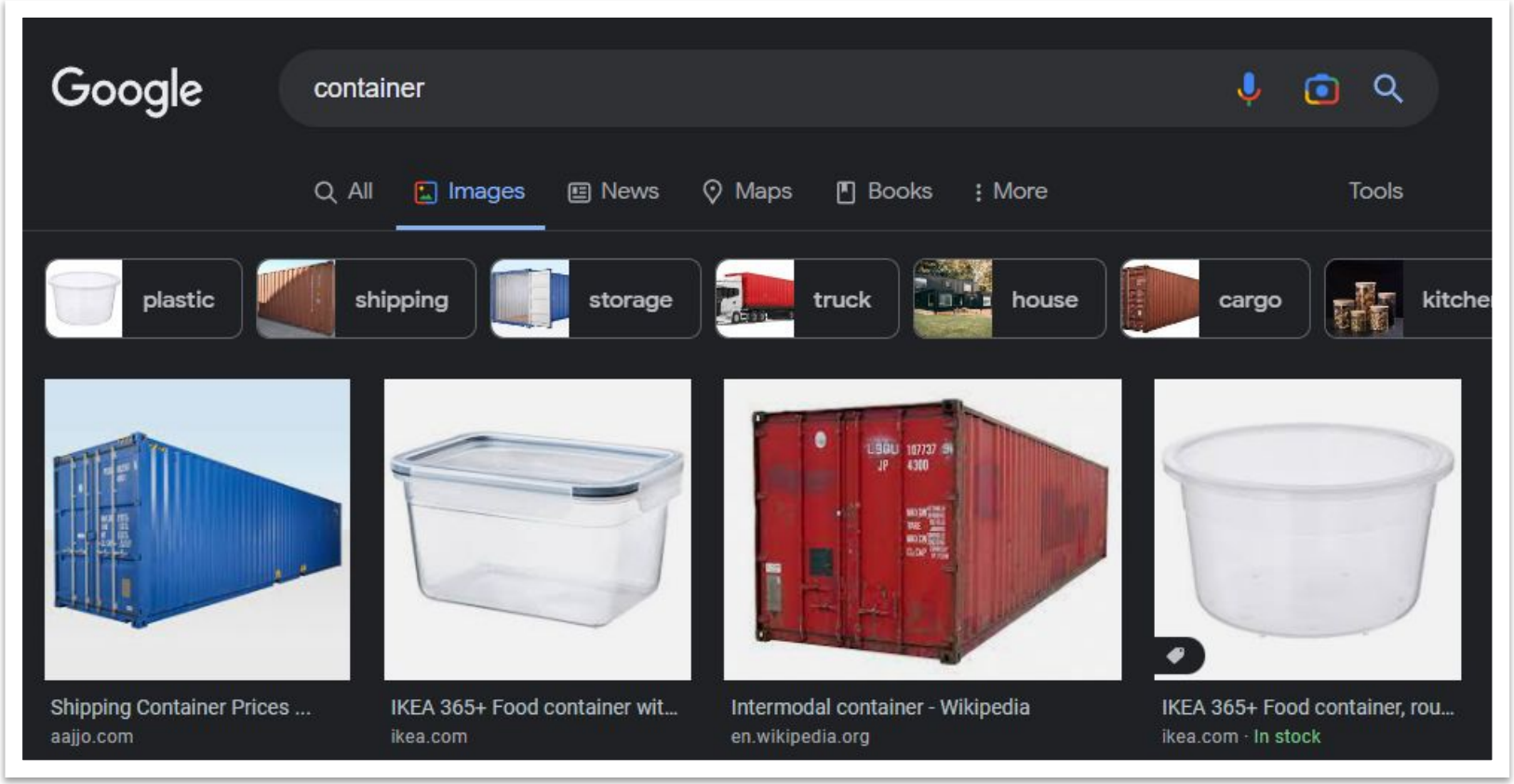
- Why do we need Kubernetes?
- What is Kubernetes?

But before that let’s go through some of the things that comes for Kubernetes to exists –

Containers

What are containers?

If we just go by the name and do a Google search -



It's something that you use to hold stuff.

What are containers?

In software context –

> Containers are a way for developers to easily package and deliver applications, and for operations to easily run them anywhere in seconds, with no installation or setup necessary.
>
> - Amir Jerbi, CTO and co-founder at Aqua Security

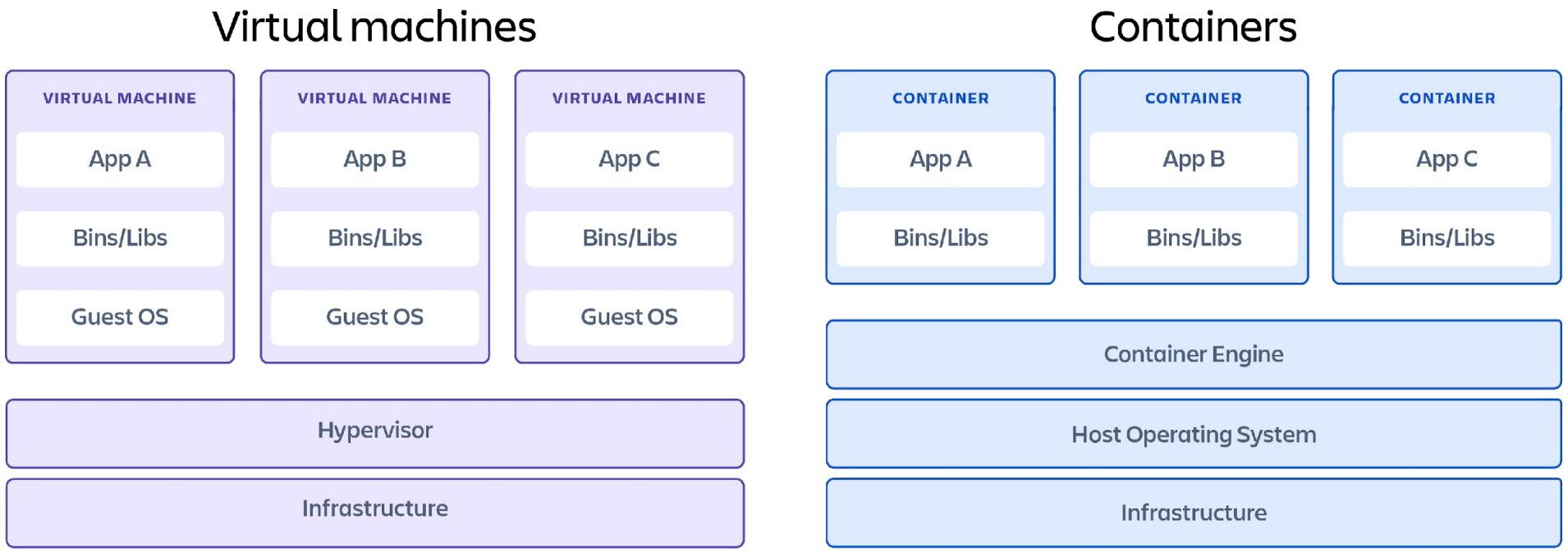
What are containers?

Isolated Linux processes.

The isolation works using two primary concepts in the linux system –

- **cgroups** - It defines what a container **can use or access**.
- **namespaces** - It defines what a container **can see**.

What are containers?



Problems?

Why do we need Kubernetes?

- Large Scale Application Deployment
- Micro service architecture
- Enable computing heavy tasks

Example?

What is Kubernetes?

Container Orchestrator. It helps in managing the containers.

- Service discovery and load balancing
- Horizontal scaling
- Self healing
- Automated rollouts and rollbacks
- Storage orchestration

Common Issues or Misconfigurations in Kubernetes

- Insecure Defaults
- SSRF
- Namespaces Bypass
- Over-permissive Containers
- Exposed Kubernetes API

Insecure Defaults

- Kubernetes and containers, while powerful, were designed for developer productivity, not necessarily security.
- Flat networking schema, allowing cross pods communications.
- No authentication on different Kubernetes components.
- and many more.

For more read - https://docs.guardrails.io/docs/vulnerabilities/kubernetes/insecure_configuration

SSRF

- Due to SSRF vulnerability in the application hosted on any of the containers, it allows the attacker to get access to the **Metadata** of the instance and even at times get access to the instance and can access the **secrets**.

Demo - https://madhuakula.com/kubernetes-goat/docs/scenarios/scenario-3

Namespace Bypass

- By default, Kubernetes has a flat networking schema, so if segmentation is required, it must be created by setting up boundaries like NSPs (network security policies), among others. We can see how to access other namespaces resources in this scenario by navigating around the namespaces.

Demo - https://madhuakula.com/kubernetes-goat/docs/scenarios/scenario-11

Over-permissive Containers

- As the name suggests, the container has excessive permission than required.
- Suppose a container, that only needs a web server has multiple other service running. It increases the attack surface.
- Suppose a container, is running with root privileges and an attacker get access to this container.

- Defense – Use Security Contexts – SELinux, Seccomp and AppArmour profile are a few good practices that can be followed.

Exposed Kubernetes API

- Kubernetes exposes an unauthenticated REST API on port 10250.
- It allows the attacker to control different aspect of the cluster, viewing secrets and checking pods in the namespaces, etc.

Demo - https://madhuakula.com/kubernetes-goat/docs/scenarios/scenario-16

What can you do about the misconfigurations?

- Performing Configuration Review of the Kubernetes Cluster.
- Following best practices while defining how an application should run.
- Container Security
- Runtime audits of the Kubernetes Clusters

Tooling for Kubernetes Security

- **kube-bench** is a tool that checks whether Kubernetes is deployed securely by running the checks documented in the CIS Kubernetes Benchmark - https://github.com/aquasecurity/kube-bench
- **Project Kubescout -** You can now have Scout Suite scan not only your cloud environments, but your Kubernetes clusters - https://github.com/nccgroup/ScoutSuite
- **Project Calico** – Secure Networking in Kubernetes - https://www.projectcalico.org/
- **Istio** - Allows you to control, connect, and secure your services on Kubernetes - https://github.com/istio/istio
- **Falco** – Kubernetes Runtime Security - https://falco.org/

...and more.

Reference and Continue Learning

- https://madhuakula.com/kubernetes-goat/ - Vulnerable by design Kubernetes Environment with multiple scenario.
- https://kubernetes.io/docs/home/ - The Kubernetes Documentation is good for learning and implementing the clusters.
- https://sysdig.com/blog/ - has some great article related to Kubernetes Security.
- https://cheatsheetseries.owasp.org/cheatsheets/Kubernetes_Security_Cheat_Sheet.html - The OWASP Cheat sheet also contains lots of information.

Thank You!

- Questions?
- Reach out to me at 0xCardinal on social media.