Secure Your Ship:

Navigating the Seas of Container Security

The Hackers Meetup

Mehul Patel,
@nomadicmehul

About me:

- Open Source Software Consultant
- Mozilla Reps Council
- Mozilla Reps Mentor
- Auth0 Ambassador
- EMS @Auth0 by Okta
- AWS Community Builder - Container
- GDG Nashik - Organizer
- AWS & GCP - Cloud Solution Architect
- Podcast Host @TACOS (Talk About Community & Open Source)

Aim of the Game

Get up to Speed

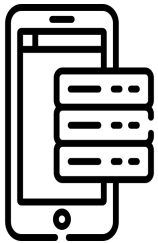
Give Directions

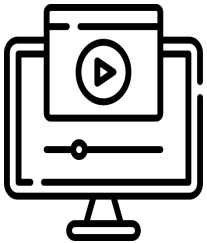
Cover Fundamentals

Less Than 45 mins

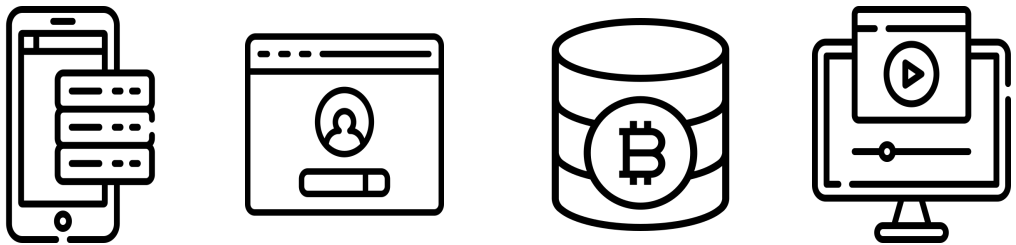
Agenda

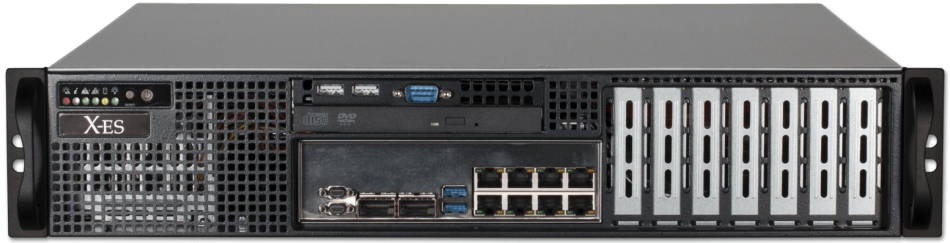
- Introduction
- Why container security is so important ?
- The Importance of Container Security
- Common Container Security Risks
- Security Tools and Techniques
- Q&A

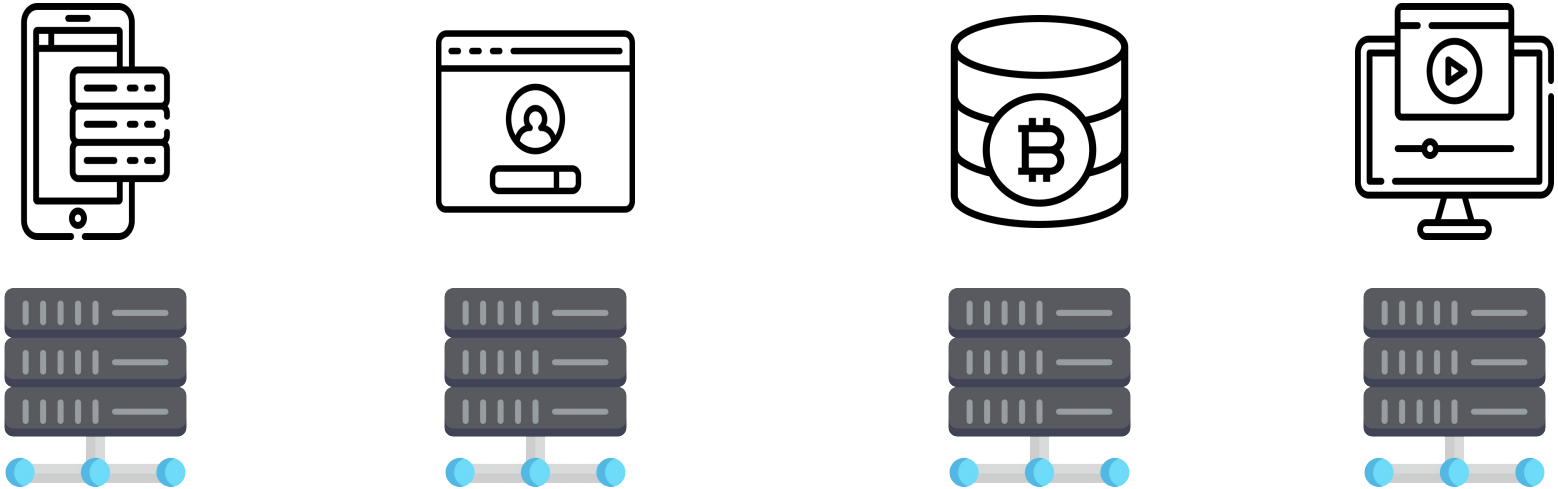


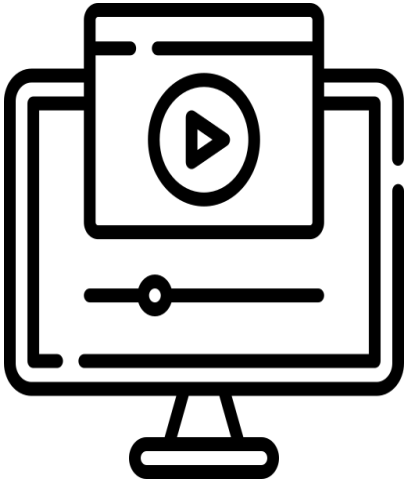


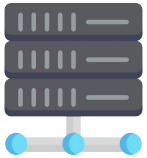
No Applications,
No business!

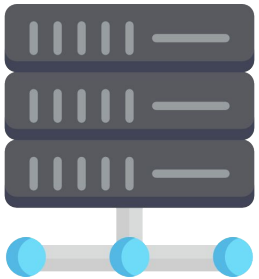



X-ES

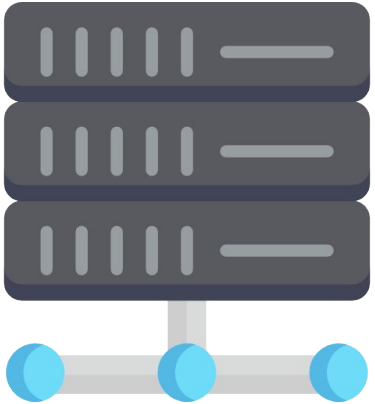


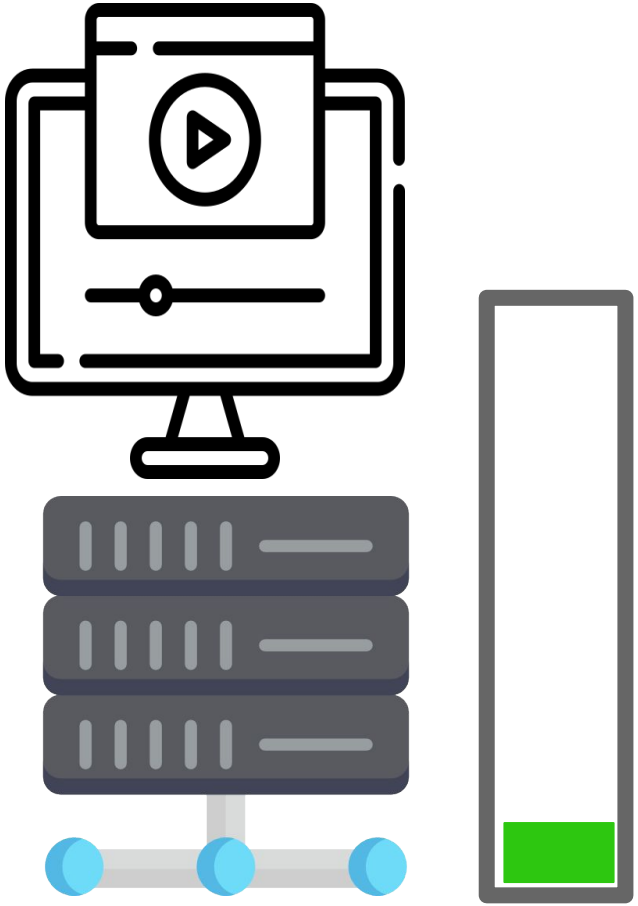




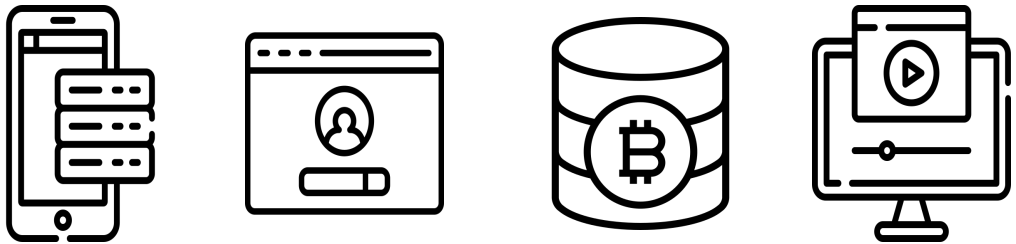


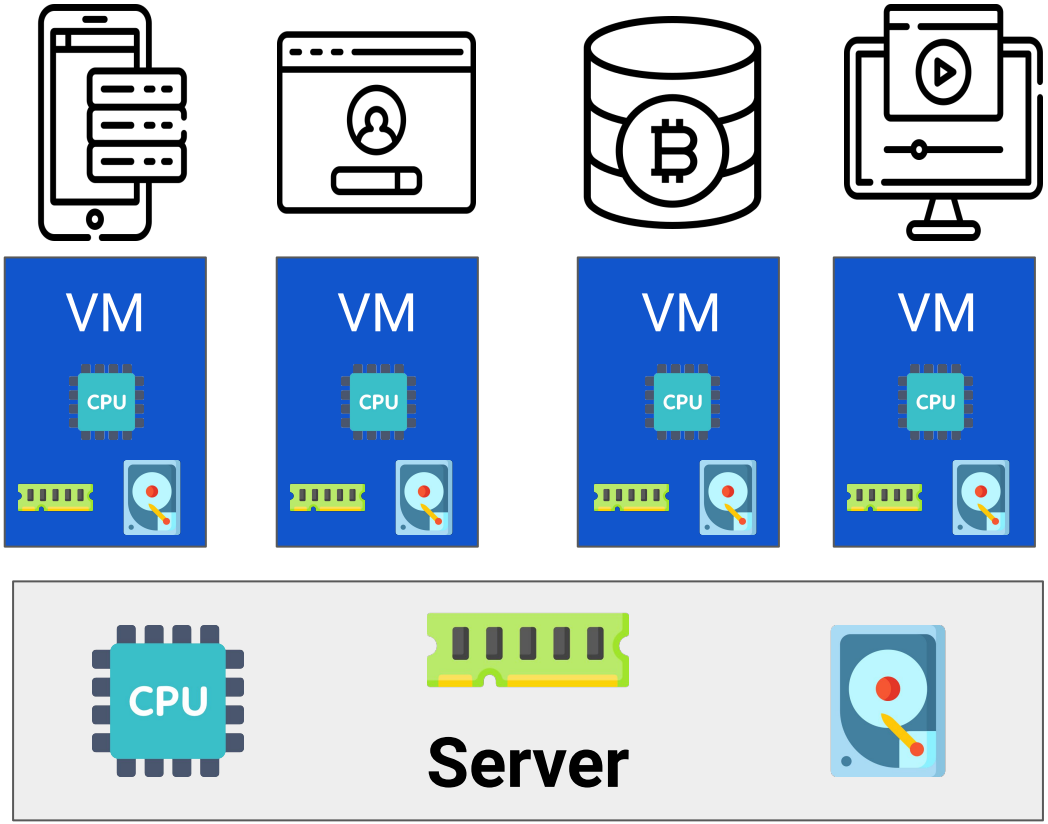
?

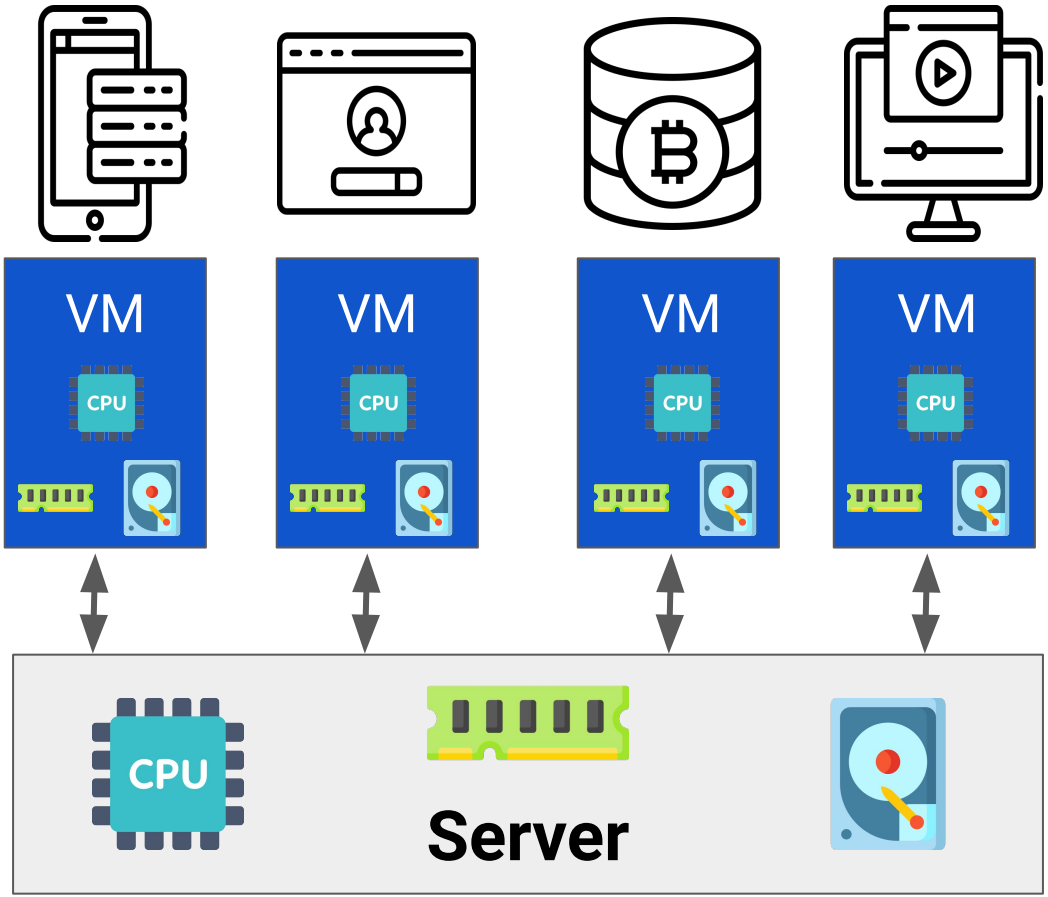


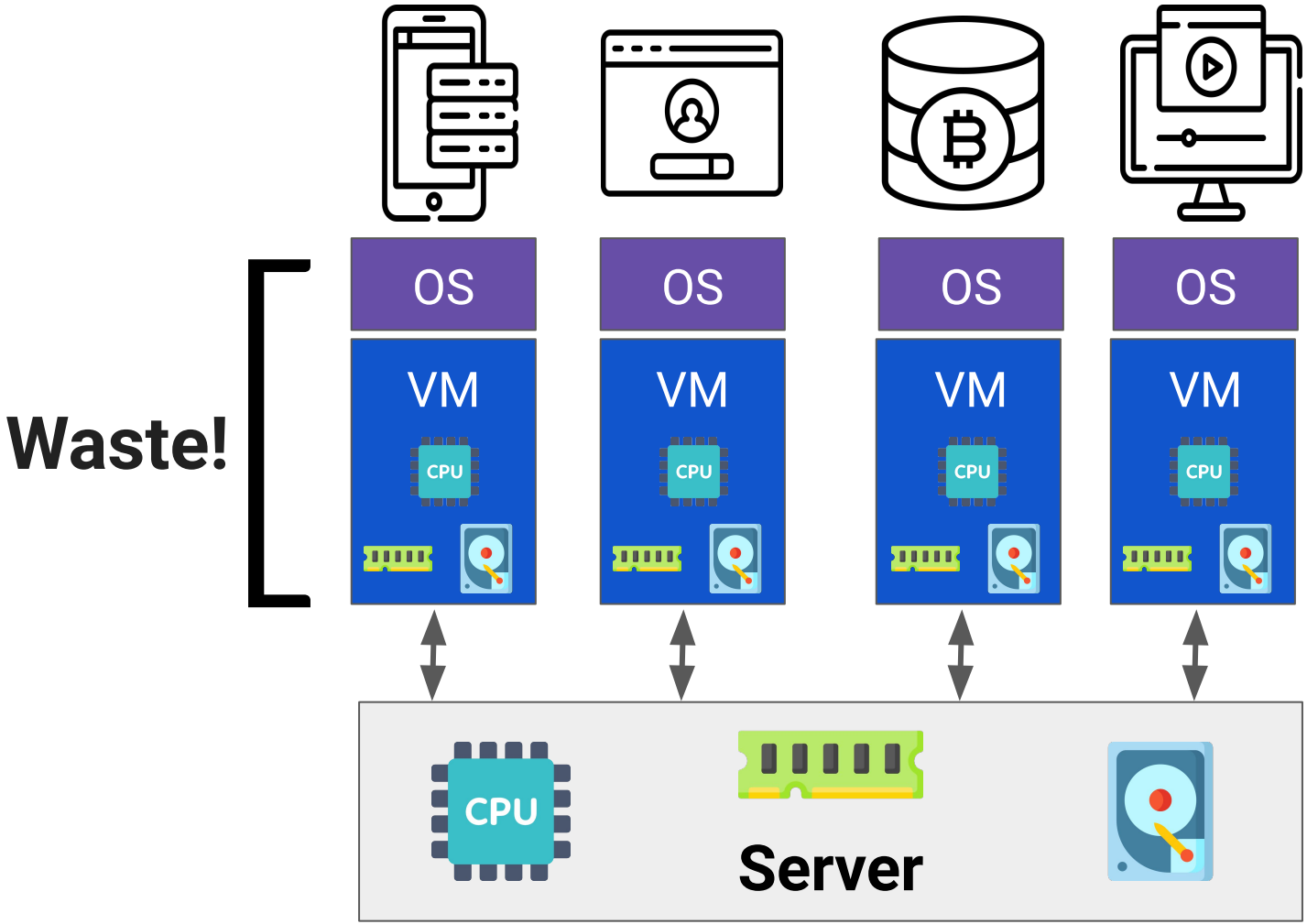


Hello VMware!


CPU
Server


VM
CPU
VM
CPU
VM
CPU
VM
CPU
CPU
Server

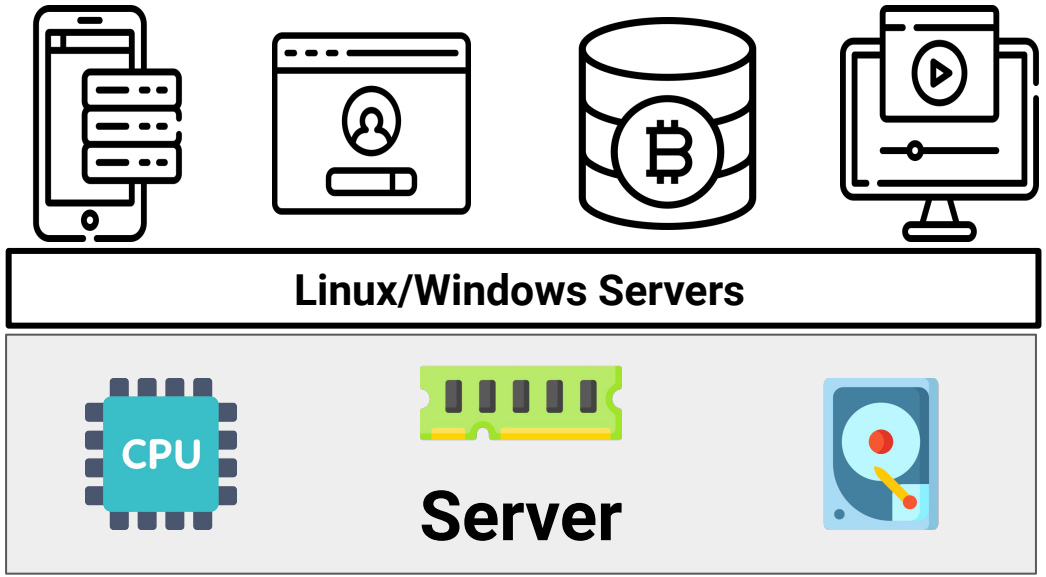

VM
CPU
VM
CPU
VM
CPU
VM
CPU
CPU
Server

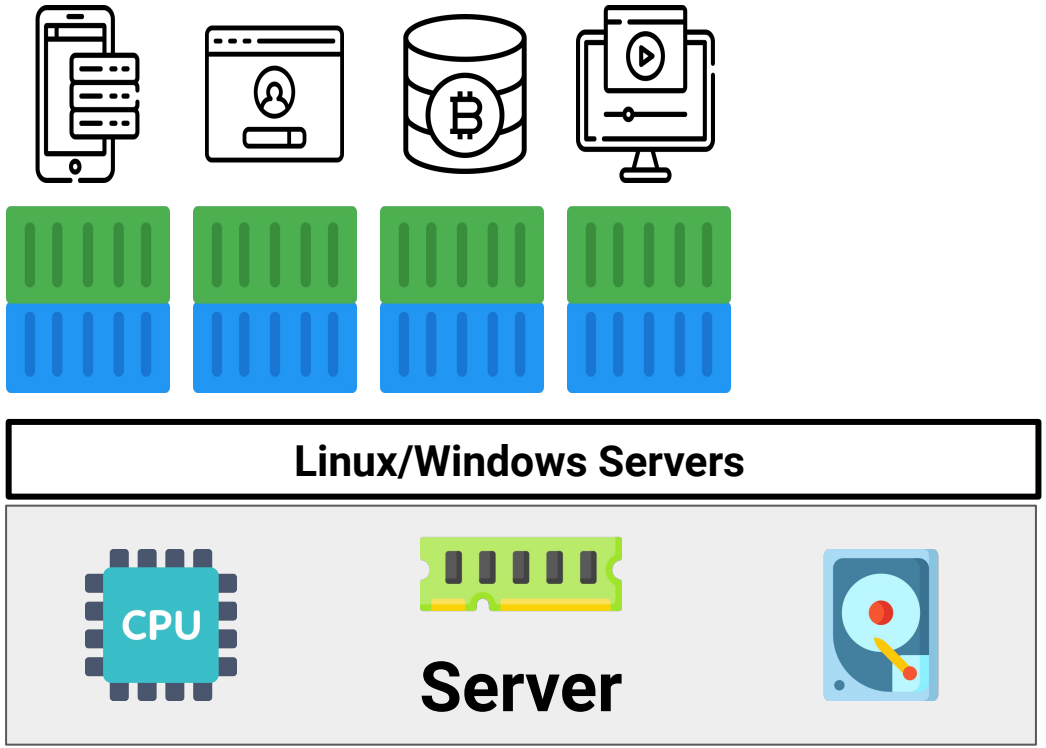


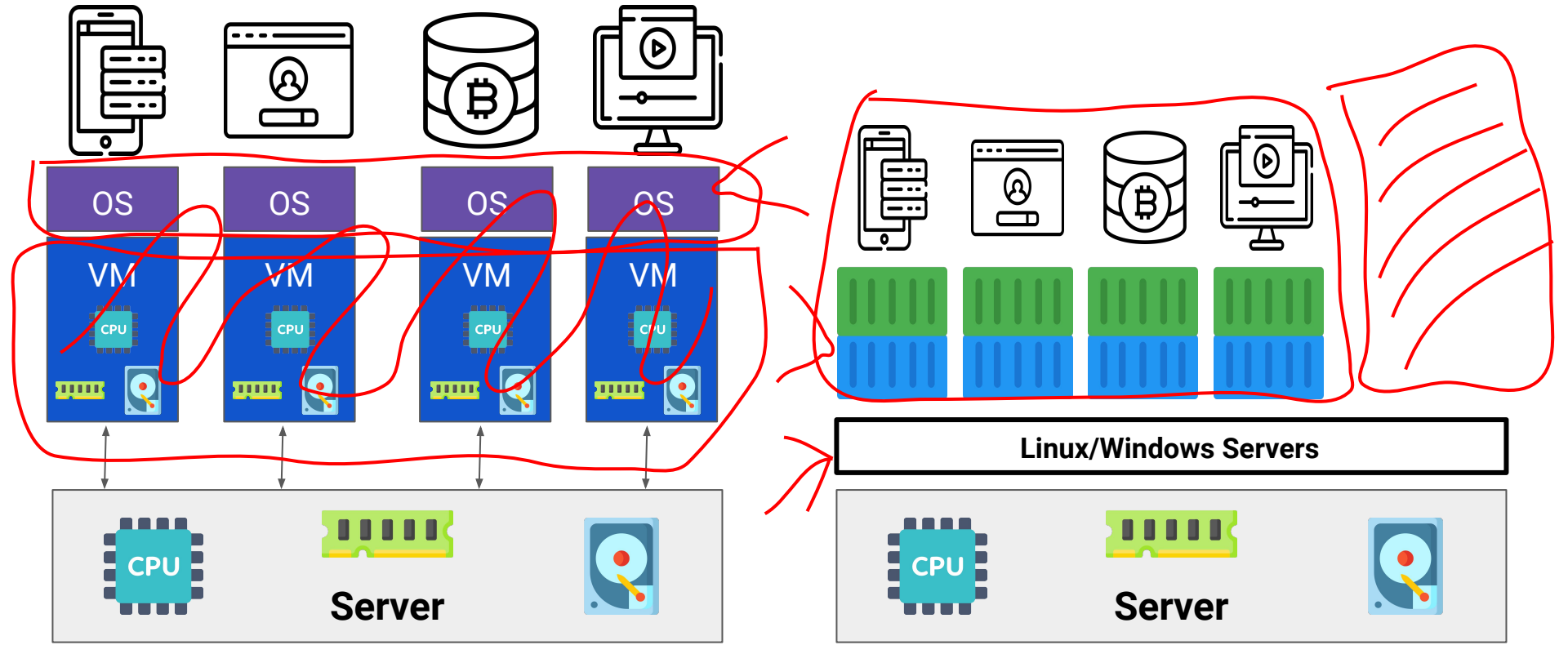
Potential OS Overheads:

- Licence Cost
- Admin
  - Patching
  - Updates
  - AV
  - More..

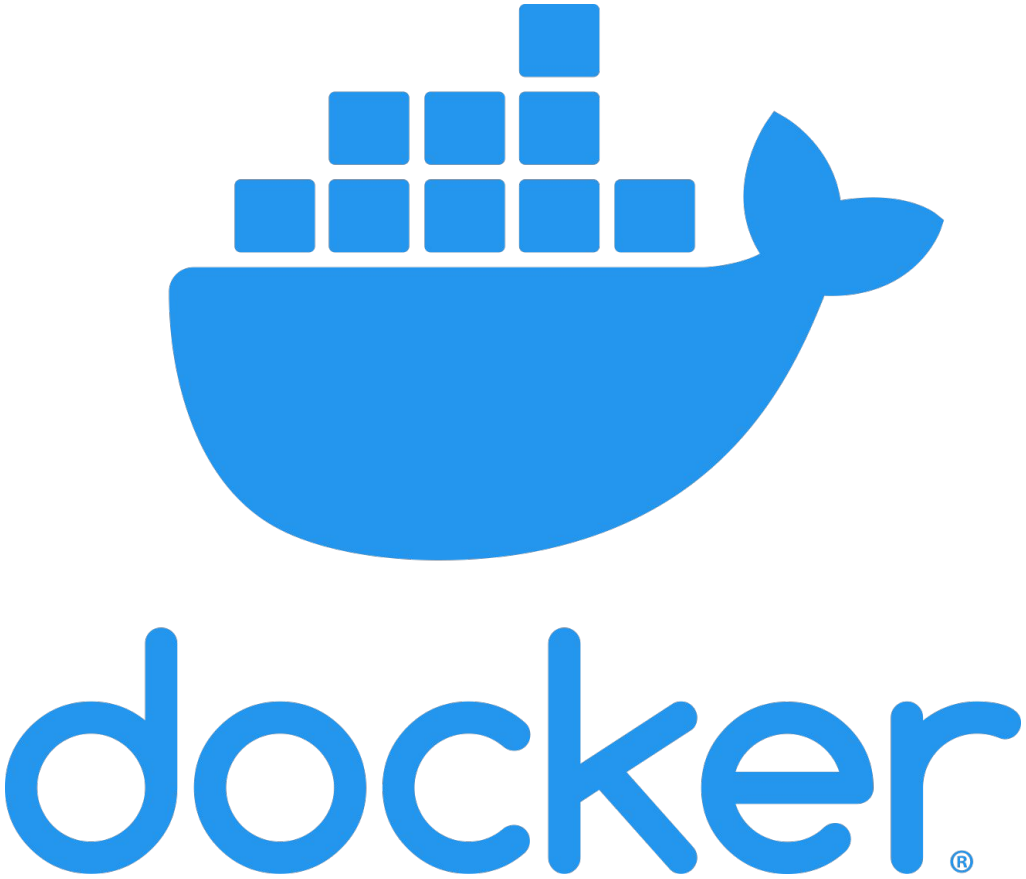
What are Containers ?


Linux/Windows Servers
CPU
Server


Linux/Windows Servers
CPU
Server

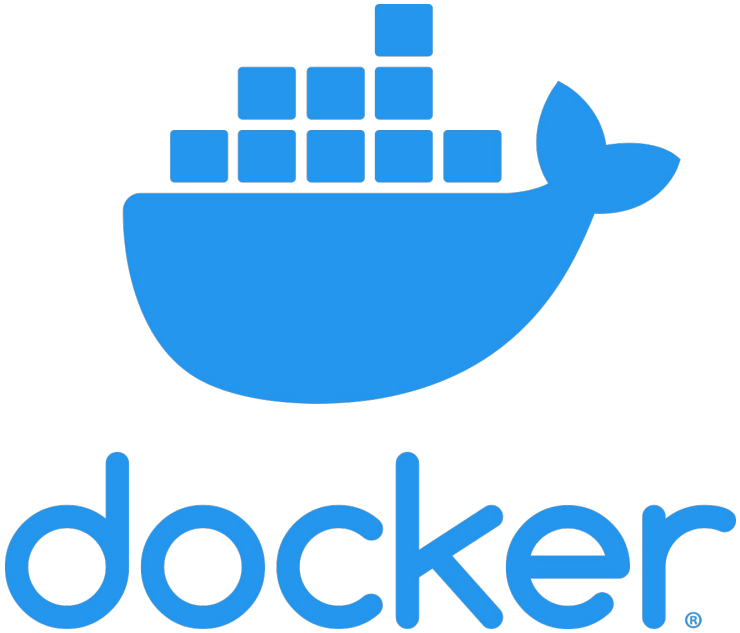

OS
OS
OS
OS
VM
VM
VM
VM
CPU
CPU
CPU
CPU
Linux/Windows Servers
CPU
Server
CPU
Server

Docker


docker

Docker: The Technology

Making containers easy



<Containerizing Apps>

The Importance of Container Security

Common Container Security Risks

Neglecting fundamental security practices

Failing to properly secure and configure tools and environments

Neglecting proper logging, monitoring, and testing procedures

Overlooking security measures throughout the entire CI/CD pipeline

5 Best Practices for Container Security

1. Secure your code and dependencies:

- Regularly update and patch your application code and its dependencies to address any known vulnerabilities.
- Implement secure coding practices to minimize the risk of introducing security flaws.

2. Start with a minimal and trusted base image:

- **Begin with a minimal base image from a trusted source to reduce the attack surface.**
- **Avoid using bloated or outdated base images that may contain unnecessary or vulnerable components.**

3. Manage image layers effectively:

- **Keep your container images lean by minimizing the number of layers and removing unnecessary components.**
- **Regularly scan and monitor the image layers for vulnerabilities and update them as needed.**

4. Implement access management:

- Apply the principle of least privilege by granting only necessary permissions to containerized applications.
- Utilize strong authentication mechanisms and implement role-based access controls to restrict access to sensitive resources.

5. Secure the container infrastructure:

- Regularly update and patch the underlying container platform, orchestrator, and host system.
- Implement network segmentation, firewall rules, and container isolation to prevent unauthorized access and lateral movement.

Security Tools and Techniques

1. Calico

- Calico Open Source is a networking and security solution for containers, virtual machines, and native host-based workloads.
- It supports a broad range of platforms including Kubernetes, OpenShift, Docker EE, OpenStack, and bare metal services.
- Implement network segmentation, firewall rules, and container isolation to prevent unauthorized access and lateral movement.

2. Clair

- Clair carries out static examination of container vulnerabilities. Today, it works with Docker containers and OCI.
- Clair consumes numerous vulnerability information sources, including Red Hat Security Data, Debian Security Bug Tracker, and Ubuntu CVE Tracker.
- Clair ingests a large amount of CVE databases for in-depth auditing.

3. Anchore Engine

- The open-source Anchore Engine is used to analyze container images and provide reporting on CVE-based security vulnerabilities.
- The Anchore Engine also assesses Docker images via custom rules to permit automated certification and validation.

4. OpenSCAP

- **OpenSCAP is a command-line tool used for auditing. It lets users load, scan, edit, export, and validate SCAP documents.**
- **SCAP (Security Content Automation Protocol) is a solution that checks for compliance for enterprise-level Linux infrastructure. It is overseen by NIST.**
- **It utilizes the Extensible Configuration Checklist Description Format (XCCDF), a common way of displaying checklist content, and clarifies security checklists.**

5. Grafeas

- Google and IBM have joined forces with a container security tool known as Grafeas that was made public in late 2017.
- This could help you develop your personal container security scanning plans.

6. Falco

- Falco is a threat detection engine for Kubernetes. It is also an open-source project and a runtime security tool used to identify anomalous behavior in containers and hosts running on Kubernetes.

- It isolates any unusual activity in your application and tells you of the threats at runtime.

7. Dagda

- **Dagda is a security tool to perform static analysis of known vulnerabilities, malware and threats in Docker images and containers.**

Any Questions?

Available on Telegram & Twitter: @nomadicmehul

Thank You!