

- Filename: eccouncil-ceh31250-v11-19-2-1-container-basics.md
- Show Name: CEHV11 (312-50)
- Topic Name: Cloud Computing - Cloud Computing
- Episode Name: Container Basics

=====

Container Basics

Objectives:

- What is a container?
 - Portable Software package/bundle
 - Config files
 - Libraries
 - Dependencies
 - Everything needed to run an app
 - Consistent across platforms
 - Scalable
 - Cost effective
 - 5-Tier Container Architecture (As defined by CEH)
 - Tier1: Developer Machines
 - Image Creation, Testing, and Accreditation
 - Tier2: Testing and Accreditation Systems
 - Verification and Validation of image contents
 - Signing Images
 - Sending Images to Registry
 - Tier3: Registries
 - Storing Images
 - Delivering Images to Orchestrators based on requests
 - Tier4: Orchestrators
 - Transforming Images into Containers
 - Deploying Containers to Hosts
 - Tier5: Hosts
 - Operating and managing Containers as instructed by the Orchestrator
- What is Docker?
 - Open source containerization platform
 - Building\
 - Deploying -----> Containerized Apps
 - Managing.... /
 - Terms
 - Images: Basic foundation for building of containers
 - Container: Created from Images and run the actual application
 - Docker Daemon: Background service that listens for Docker API requests and

manages docker objects like Images, Containers, Networks, and Volumes

- Docker Client: Primary way most users interact with Docker
- Docker Registry: aka Docker Hub. Repo of official Images. Private registries are configurable
- Dockerfile: Simple text file that contains a list of commands that the Docker client calls while creating an image

- What is orchestration?

- Automation of container lifecycle

- Provisioning
 - Configuring
 - Deploying
 - Security
 - Monitoring
 - Resource allocation
 - Scaling

- Orchestration Apps

- Docker Swarm
 - Kubernetes
 - OpenShift
 - Ansible

- Container Security Challenges

- Large attack surface

- Increased complexity through many objects
 - Containers
 - Apps
 - Databases

- Container breakout

- Attacker can breach the 'wall' between the container and host
 - Running as root

- Vulnerable source code

- Devs use containers for testing code
 - Could expose org to attack through insecure code

- Insecure storage of secrets

- API Keys
 - Usernames
 - Passwords

- Noisy Neighbor

- Containers may exhaust resources
 - Makes other containers fail due to lack of resources