Version : **2024.01**

Last update : 2024/12/17 13:47

DOE600 - Course Presentation

Content

- **DOE600 - Course Presentation**.
    - Content
    - Prerequisites
        - Hardware
        - Software
        - Internet
    - Using the Infrastructure
    - Curriculum

Prerequisites

Hardware

- One computer (MacOS, Linux, Windows™ or Solaris™),
- AZERTY FR or QWERTY US keyboard,
- 4 GB RAM minimum,
- 2-core processor minimum,
- Headphones or earphones,
- A microphone (optional).

Software

- Web Chrome version 72+ or
- Microsoft Edge version 79+ or
- Firefox version 65+.

Internet

- **Fast** Internet access (4G minimum) **WITHOUT** using a proxy,
- **Unblocked** access to ports 80 and 443 at: https://www.ittraining.team **and its sub-domains**.

Curriculum

- **DOE600 - Course Presentation**
  - Prerequisites
    - Hardware
    - Software
    - Internet
  - Use of the Infrastructure
  - Curriculum

- **DOE601 - Virtualisation by Isolation**
  - Presentation of Virtualisation by Isolation
    - History
  - Presentation of Namespaces
  - Presentation of CGroups
    - LAB #1 - cgroups v1
      - 1.1 - Preparation
      - 1.2 - Presentation
      - 1.3 - Memory Limitation
      - 1.4 - The cgcreate command
      - 1.5 - The cgexec command

- 1.6 - The cgdelete command
- 1.7 - The /etc/cgconfig.conf file
- 1.8 - The cgconfigparser command
  - LAB #2 - cgroups v2
    - 2.1 - Preparation
    - 2.2 - Overview
    - 2.3 - Limiting CPU Resources
    - 2.4 - The systemctl set-property command
  - Introducing Linux Containers
    - LAB #3 - Working with LXC
      - 3.1 - Installation
      - 3.2 - Creating a Simple Container
      - 3.3 - Starting a Simple Container
      - 3.4 - Attaching to a Simple Container
      - 3.5 - Basic LXC Commands
        - The lxc-console Command
        - The lxc-stop Command
        - The lxc-execute Command
        - The lxc-info Command
        - The lxc-freeze Command
        - The lxc-unfreeze Command
        - Other Commands
      - 3.6 - Creating an Ephemeral Container
        - The lxc-copy Command
      - 3.7 - Saving Containers
        - The lxc-snapshot Command

- **DOE602 - Getting started with Docker**
  - Introduction to Docker
    - Virtualisation and Containerisation
    - The AUFS File System
    - OverlayFS and Overlay2
    - Docker Daemon and Docker Engine
    - Docker CE and Docker EE

- Docker CE
- Docker EE
- Docker and Mirantis
- LAB #1 - Working with Docker
  - 1.1 - Installing docker on Linux
    - Debian 11
    - CentOS 8
  - 1.2 - Starting a Container
  - 1.3 - Viewing the list of Containers and Images
  - 1.4 - Searching for an Image in a Repository
  - 1.5 - Deleting a Container from an Image
  - 1.6 - Creating an Image from a Modified Container
  - 1.7 - Deleting an Image
  - 1.8 - Creating a Container with a Specific Name
  - 1.9 - Executing a Command in a Container
  - 1.10 - Injecting Environment Variables into a Container
  - 1.11 - Modifying a Container Host Name
  - 1.12 - Mapping Container Ports
  - 1.13 - Starting a Container in Detached mode
  - 1.14 - Accessing Container Services from the Outside
  - 1.15 - Stopping and Starting a Container
  - 1.16 - Using Signals with a Container
  - 1.17 - Forcing the deletion of a running Container
  - 1.18 - Simply using a Volume
  - 1.19 - Downloading an image without creating a Container
  - 1.20 - Attaching to a running Container
  - 1.21 - Installing software in a Container
  - 1.22 - Using the docker commit command
  - 1.23 - Connecting to the container from the outside

- **DOE603 - Managing and Storing Docker Images**
  - LAB #1 - Re-creating an official docker image
    - 1.1 - Using a Dockerfile
    - 1.2 - FROM

- 1.3 - RUN
      - 1.4 - ENV
      - 1.5 - VOLUME
      - 1.6 - COPY
      - 1.7 - ENTRYPOINT
      - 1.8 - EXPOSE
      - 1.9 - CMD
      - 1.10 - Other commands
    - LAB #2 - Creating a Dockerfile
      - 2.1 - Creating and testing the script
      - 2.2 - Good Cache Practices
    - LAB #3 - Installing a Private Registry
      - 3.1 - Creating a Local Registry,
      - 3.2 - Creating a Dedicated Registry Server
        - Configuring the Client

- **DOE604 - Volume, Network and Resource Management**
  - LAB #1 - Volume Management
    - 1.1 - Automatic management using Docker
    - 1.2 - Manual Volume Management
    - 1.3 - Manual management of a Bindmount
  - LAB #2 - Network Management
    - 2.1 - The Docker Network Approach
      - Bridge
      - Host
      - None
      - Links
    - 2.2 - Running Wordpress in a container
    - 2.3 - Managing a Microservices Architecture
  - LAB #3 - Monitoring Containers
    - 3.1 - Logs
    - 3.2 - Processes
    - 3.3 - Continuous Activity

- **DOE605 - Docker Compose, Docker Machine and Docker Swarm**

- LAB #1 - Docker Compose
    - 1.1 - Installation
    - 1.2 - Installing Wordpress with Docker Compose
- LAB #2 - Docker Machine
    - 2.1 - Introduction
    - 2.2 - Creating Docker Virtual Machines
    - 2.3 - Listing Docker VMs
    - 2.4 - Obtaining VM IP addresses
    - 2.5 - Connecting to a Docker VM
- LAB #3 - Docker Swarm
    - 3.1 - Overview
    - 3.2 - Initializing Docker Swarm
    - 3.3 - Leader status
    - 3.4 - Joining the Swarm
    - 3.5 - Viewing Swarm Information
    - 3.6 - Starting a Service
    - 3.7 - Scaling Up and Scaling Down a Service
    - 3.8 - Checking Node Status
    - 3.9 - High Availability
    - 3.10 - Deleting a Service
    - 3.11 - Backing up Docker Swarm
    - 3.12 - Restoring Docker Swarm

- **DOF606 - Overlay Network Management with Docker in Swarm mode**
    - The Docker Network Model
    - LAB #1 - Overlay Network Management
        - 1.1 - Creating a network overlay
        - 1.2 - Creating a Service
        - 1.3 - Moving the Service to another Overlay Network
        - 1.4 - DNS container discovery
        - 1.5 - Creating a Custom Overlay Network
    - LAB #2 - Microservices Architecture Management
        - 2.1 - Implementing Docker Swarm with overlay networks

- **DOF607 - Docker Security Management**

- LAB #1 - Using Docker Secrets
- LAB #2 - Creating a Trusted User to Control the Docker Daemon
- LAB #3 - The docker-bench-security.sh script
- LAB #4 - Securing the Docker Host Configuration
- LAB #5 - Securing the Docker daemon configuration
  - 5.1 - The /etc/docker/daemon.json file
- LAB #6 - Securing Images and Build Files
- LAB #7 - Securing the Container Runtime
- LAB #8 - Securing Images with Docker Content Trust
  - 8.1 - DOCKER_CONTENT_TRUST
  - 8.2 - DCT and the docker pull command
    - The disable-content-trust option
  - 8.3 - DCT and the docker push command
  - 8.4 - DCT and the docker build command
    - Creating a second Repositry
    - Deleting a signature
- LAB #9 - Securing the Docker daemon socket
  - 9.1 - Creating the Certificate Authority Certificate
  - 9.2 - Creating the Docker Daemon Host Server Certificate
  - 9.3 - Creating the Client Certificate
  - 9.4 - Starting the Docker Daemon with a Direct Invocation
  - 9.5 - Configuring the Client

- **DOE608 - Course Validation**
  - Course Materials
  - What this course covered
  - Validation of acquired knowledge
  - Course Evaluation

From:
 https://www.ittraining.team/ - **www.ittraining.team**

Permanent link:
https://www.ittraining.team/doku.php?id=elearning:workbooks:docker3:en:start

Last update: **2024/12/17 13:47**