Version : **2020.01**

Last update: 2020/08/27 16:13

DOE100 - Docker: Implementation

Présentation

Objectives : Master the implementation of Operating-system-level virtualization with Docker.
Who can benefit : Linux Technicians and Administrators.
Prerequisites : One of the following certifications or the equivalent skills: CompTIA Linux+ Powered by LPI or LPIC-1 or SUSE CLA or ITT Debian Linux - Technician or ITT CentOS Linux - Technician.
Learning technique : Clear, theoretical course content divided into lessons and extensive LABS.
Student Progression : Student progression is monitored both in terms of effective attendance and in terms of comprehension using self-assessment tests.
Duration : 2 days (14 hours).

Prerequisites

Hardware

- A computer running MacOS, Linux, Windows™ or Solaris™,
- AZERTY FR or QWERTY US keyboard,
- Minimum 4 GB of RAM,
- Minimum dual-core processor,
- Headphones/Earphones,
- A microphone (optional).

Software

- If Windows™ - Putty and WinSCP,
- Chrome or Firefox web browser.

Internet

- A fast Internet connection (4G minimum) and **no** proxy,
- Unhindered access to the following domains : https://my-short.link, https://itraining.center, https://ittraining.io, https://ittraining.institute, https://ittraining.support.

Curriculum

Day #1

- **DOE100 - Docker : Implementation** - 1 hour.
    - Prerequisites
        - Hardware
        - Software
        - Internet
    - Using the Infrastructure
        - Connecting to the Cloud Server
            - Linux, MacOS and Windows 10 with a built-in ssh client
            - Windows 7 and Windows 10 without a built-in ssh client
        - Starting the Virtual Machine
        - Connecting to the Virtual Machine
    - Course Curriculum

- **DOE101 - Operating-system-level virtualization** - 3 hours.
    - What is Operating-system-level virtualization?
        - A brief history
    - What are Namespaces?
    - What are CGroups?
        - LAB #1 - Working with CGroups
            - 1.1 - Capping memory usage
            - 1.2 - The cgroup-tools package
                - The cgcreate command
                - The cgexec command
                - The cgdelete command
                - The /etc/cgconfig.conf file
    - What are Linux Containers?
        - LAB #2 - Working with LXC
            - 2.1 - Installation
            - 2.2 - Creating a simple container
            - 2.3 - Starting a simple container
            - 2.4 - Attaching a terminal to a running container
            - 2.5 - 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
            - 2.6 - Creating an unprivileged container
                - User Namespaces
                - Creating a dedicated user
                - Setting up the mapping
                - Creating the container
                - Checking out the mapping
            - 2.7 - Creating an unpersistant container
                - The lxc-copy command
            - 2.8 - Backing up containers
                - The lxc-snapshot command

- **DOE102 - Getting Started with Docker** - 3 hours.
    - What is Docker?
    - LAB #1 - Working with Docker
        - 1.1 - Installing Docker
        - 1.2 - Starting a container
        - 1.3 - Viewing a list of containers and images
        - 1.4 - Searching for an image in a repository
        - 1.5 - Deleting a container
        - 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 within a container
        - 1.10 - Injecting Environment Variables into a container
        - 1.11 - Modifying the hostname of a container
        - 1.12 - Port mapping
        - 1.13 - Starting a container in the background
        - 1.14 - Accessing services from outside the container
        - 1.15 - Starting and stopping a container
        - 1.16 - Using Signals with a Container
        - 1.17 - Deleting a running container
        - 1.18 - Using volumes
        - 1.19 - Downloading an image without creating a container
        - 1.20 - Attaching to a running container
        - 1.21 - Installing a package in a container
        - 1.22 - Using the docker commit command
        - 1.23 - Connecting to a running server within the container

Day #2

- **DOE103 - Managing Docker Images** - 3 hours.
    - LAB #1 - Recreating an official Docker image
        - 1.1 - Dockerfiles
        - 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 simple Dockerfile
        - 2.1 - Create and test the script
        - 2.2 - Cache management
- **DOE104 - Managing volumes, the network and resources** - 3 hours.
    - LAB #1 - Managing volumes
        - 1.1 - Automatically
        - 1.2 - Manually
    - LAB #2 - Managing the network

    - 2.1 - Docker networks
      - Bridge
      - Host
      - None
      - Links
    - 2.2 - Wordpress in a container
    - 2.3 - Managing microservices
  - LAB #3 - Monitoring
    - 3.1 - Logs
    - 3.2 - Processus
    - 3.3 - Activity
  - LAB #4 - Managing resources
    - 4.1 - Memory

- **DOE105 - Course completion** - 1 hour.
  - What's next?
    - Training materials
    - What you need
      - Hardware
      - Software
      - Virtual Machine
  - What we covered
    - Day #1
    - Day #2
  - Resetting the course infrastructure
  - Evaluate the training session
  - Thanks
