Iaas-level Components

Alfonso Pérez, Carlos de Alfonso, Germán Moltó,

Universitat Politècnica de Valènica

INDIGO-DataCloud is co-founded by the
Horizon 2020Framework Programme

Index

- OneDock
- NovaDocker
- UDocker

Index

- **OneDock**
- NovaDocker
- UDocker

ONEDock in one tweet

INDIGO - DataCloud

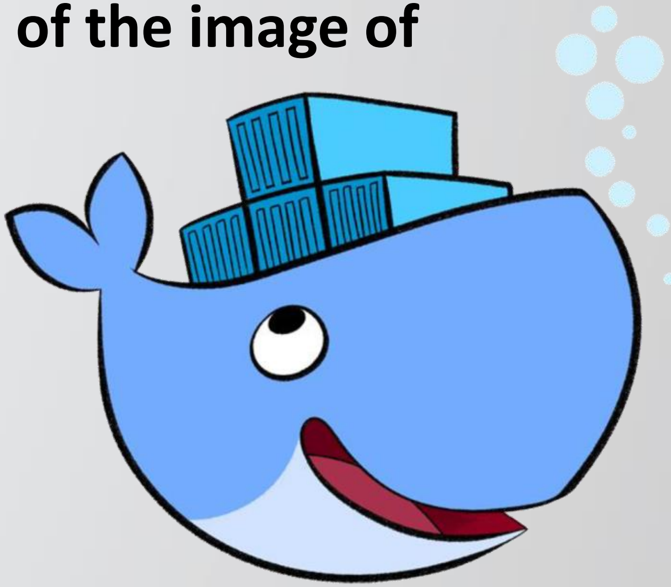
ONEDock is a set of extensions for **OpenNebula** to **use Docker containers** as first-class entities, as if they were **lightweight Virtual Machines**

ONEDock

- ONEDock tries to **integrate Docker as any other hypervisor** available in ONE (KVM, VMWare, Xen, etc.)
  - When ONE is asked for one VM, ONEDock will make that **ONE delivers the user a Docker container**.
  - The lifecycle of the containers is carried out as ONE manages the lifecycle of a VM:
  - Creating, destroying, saving, migrating, etc.

Why Docker for ONE?

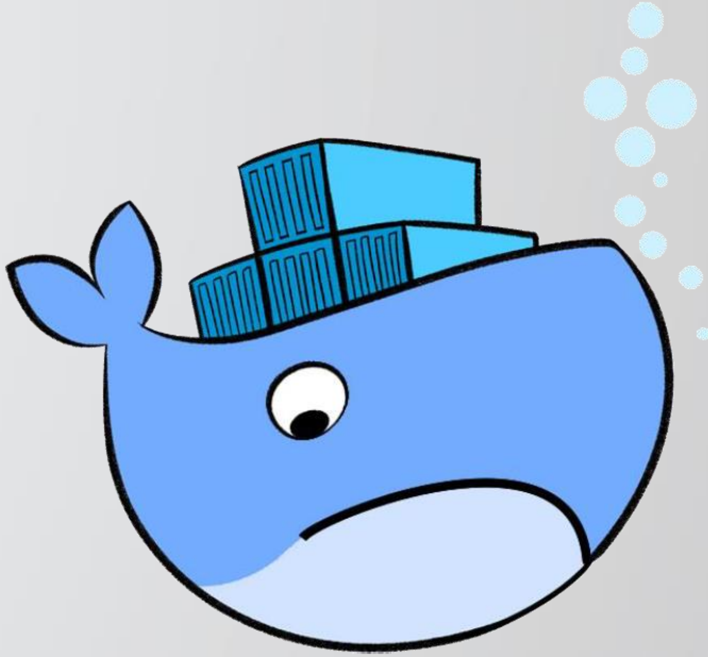
- Container technologies have **gained significant momentum** in the last years.
- Docker includes unique features:
  - **DockerHub** with lots of applications already packed.
  - The Dockerfile mechanism that eases the creation and distribution of images.
  - The usage of layered file systems that **reduce the footprint of the image of the container.**
- Docker seems to be **the winner hype**.



Why not Docker for ONE?

- Docker is mainly **conceived for application delivery** and not for behaving as a long-lasting VM.

- LXC/LXD will probably be a more natural solution.



ONEDock Components (I)

- A ONEDock **datastore** is created to use the features of Dockerfiles and the usage and creation of the container images:
    - ONEDock installs a **private Docker index**.
    - Use **images from DockerHub** or import from exported Docker containers.
- The ONEDock **transfer manager** gets the Docker images from the ONEDock datastore.

ONEDock Components (II)

- The ONEDock **monitoring driver** provides ONE with **information about the running containers** and the state of the Docker hypervisor.

- The ONEDock **virtual machine manager** translates the operations of the lifecycle of the“ONE VMs” into Docker operations:
  - Currently **only the basic operations** of the lifecycle of VMs are available(e.g. creating, destroying, etc.).
  - Not all of the operations will be posible, because of the concept of Docker (e.g hot attaching a volume).

ONEDock Workflow

1. **A Docker image is registered** in ONE:
   - The process consists of **downloading the image into the private registry** to reduce the external traffic and to match the concept of image in ONE.
2. A ONEDock **VM is created** and scheduled:
   - The host downloads **the image from the private registry** and **starts the container**.
   - Configures the container to act “like a ONE VM”
   - Configure a VNC console, give full access to the network.
3. The **container is accesible** as if it were a VM.

Technical details (I)

- The concept of **Docker does not match the concept of a VM**.
  - Containers do not have full access to the network
    - ONEDock **hacks the network** to deliver full working Ips instead of exposing individual ports.
  - Docker containers do not have a “console”.
    - ONEDock uses **SVNC term that emulates a VNC console** on a command execution (i.e. bash inside the container).
  - A Docker container executes an application.
    - It is important to **ensure that the application will not end**, to keep the container running.

Technical details (II)

- Some features of the VMs need that **containers are granted with specific privileges when** mapped to Docker (e. g. mounting block devices)
  - It is a security issue that is difficult to solve.
  - Some **workarounds are included** in ONEDock.
  - Mounting the devices in the host and then mapping folders to the container filesystem.
- It is difficult to translate some concepts of the VMs to the containers.
  - e. g. hot attaching volumes or NICs.

More info

- Can be obtained from the public repository **https://github.com/indigo-dc/onedock**

- Distributed under the Apache 2.0 license.

Index

- OneDock
- **NovaDocker**
- UDocker

Overview

- Container support under linux relies on built in kernel features
  - cgroups for limitation and prioritization of resources
  - namespaces to isolate applications regarding process trees, networking, user IDs and mounted filesystems.
- A commonly used abstraction over this functionality is LXC (Linux Containers) which exposes simplified interfaces and libraries.
- Tools like Docker provide an additional abstraction layer simplifying deployment and management of container units, introducing the concepts of images and tags.

Nova-Docker

- OpenStack provides container support via its compute service called Nova.
- Drivers exist to create and manage containers both using LXC (with the libvirt driver) and using Docker.
  - In both cases containers are treated as lightweight virtual machines, offering similar isolation to traditional VMs but using the same kernel as the host.
- Nova-docker aims to ease the container management in OpenStack.

Index

- OneDock
- NovaDocker
- **UDocker**

udocker

- Imagine that Computing & Storage resources are made available as a ***pool**,* in which*:*

  1. *You cannot count on having your very specific software/libraries installed,*
  2. *There is no system software available to run your application encapsulated as a container*

Containers in multi-user environment

- **Adoption of docker is being very slow in computing farms or interactive linux system shared by many users**
- Particularly so in large infrastructures operated for many users (HPC systems)
- The typical situation is that docker **is not installed**, and one cannot run containers without support from the system software.
- The main issue is that docker needs root permissions to run a container.
- Even though the user, within the context of the container is completely isolated from the rest of the machine, it raises all the alarms among security people
- A user with access to docker can own the hosting system

udocker: capabilities

- **It is a tool to execute content of docker containers in user space** when docker is not available
  - enables download of docker containers from dockerhub
  - enables execution of docker containers by non-privileged users
- **It is executed under the regular user id (no root privileges needed anymore).**
  - privileges are not used in any step not for running not for installing
- **It can be used to execute the content of docker containers in Linux batch systems and interactive clusters managed by others**
- **Acts as a wrapper around other tools to mimic docker capabilities**

- **More info and downloads at:**
  - https://www.gitbook.com/book/indigo-dc/udocker/details
  - https://indigo-dc.gitbooks.io/udocker/content/doc/user_manual.html

udocker: basic description

- Everything is stored in the $HOME or some other directory belonging to the user (tunable parameter).
  - Container layers are download to the above specified directory
  - Directory trees can be created/extracted from these container layers
- Uses the **ptrace** mechanism to change pathnames and execute transparently inside a directory tree
- **No impact on read/write or execution**, only impact on system calls using pathnames (open, chdir, etc)
  - **NO IMPACT ON THE PERFORMANCE OF THE CODES**
- **Does not require installation of additional software** in the host system: ***udocker* is a python script**