7th UCAAT User Conference on Advanced Automated Testing

ETSI

DEPLOYING FUNCTIONAL TESTING TOOLS ON CONTINUOUS INTEGRATION

Presented by Zineb Regragui & Eric Merisier

sogeti Part of Capgemini

7th UCAAT User Conference on Advanced Automated Testing

ETSI

AGENDA

sogeti Part of Capgemini

Agenda

- How to accelerate testing tools deployment ?
- Dockers concepts and benefits
- Tutorial Docker Desktop
- Question ?

7th UCAAT User Conference on Advanced Automated Testing

ETSI

HOW TO ACCELERATE TESTING TOOLS DEPLOYMENT ?

sogeti Part of Capgemini

Business Expectations towards the projects

- **High Responsiveness**
  - Responding quickly to the ***customers needs*** to differentiate and gain market shares
- **High Productivity**
  - Reducing the ***Time to Market*** of new functionalities to take the advantage over the competitors
- **High Quality**
  - Optimizing ***End User Experience*** to retain and win customers

The pillars of success

Whatever the methodology, its success relies on :

- Measurements
- Sharing
- Automation

Measurements

Containers provide the ability to run a **process** in an isolated environment.

This isolation allows you to run **many services** simultaneously **on a given host**.

Unlike virtual machines, physical resources are not reserved but **allocated dynamically**

Docker lets you increase your test volume

Sharing

Containers allow testers to package up a testing tool

Images are pushed in a **repository** and made available to the rest of the team

Several images, public or private, can be assembled to compose a complete testing solution, a **service**

Docker encourages sharing of components

Automation

It takes only a few minutes to run **containers** from an **image** into a testing environnement.

Containers can be created and removed on demand

Container executions can be managed by CI

Docker leverages automation to increase testing frequency

7th UCAAT User Conference on Advanced Automated Testing

ETSI

DOCKERS CONCEPTS AND BENEFITS

sogeti Part of Capgemini

Docker benefits for testing

Configuration work done only once

Quick deployment on multiple environments

Parallelization of various configurations on the same server

High frequency execution via an automation server

Increasing speed and reproducibility, Docker improves the productivity of the test team

7th UCAAT User Conference on Advanced Automated Testing

ETSI

TUTORIAL DOCKER DESKTOP

sogeti Part of Capgemini

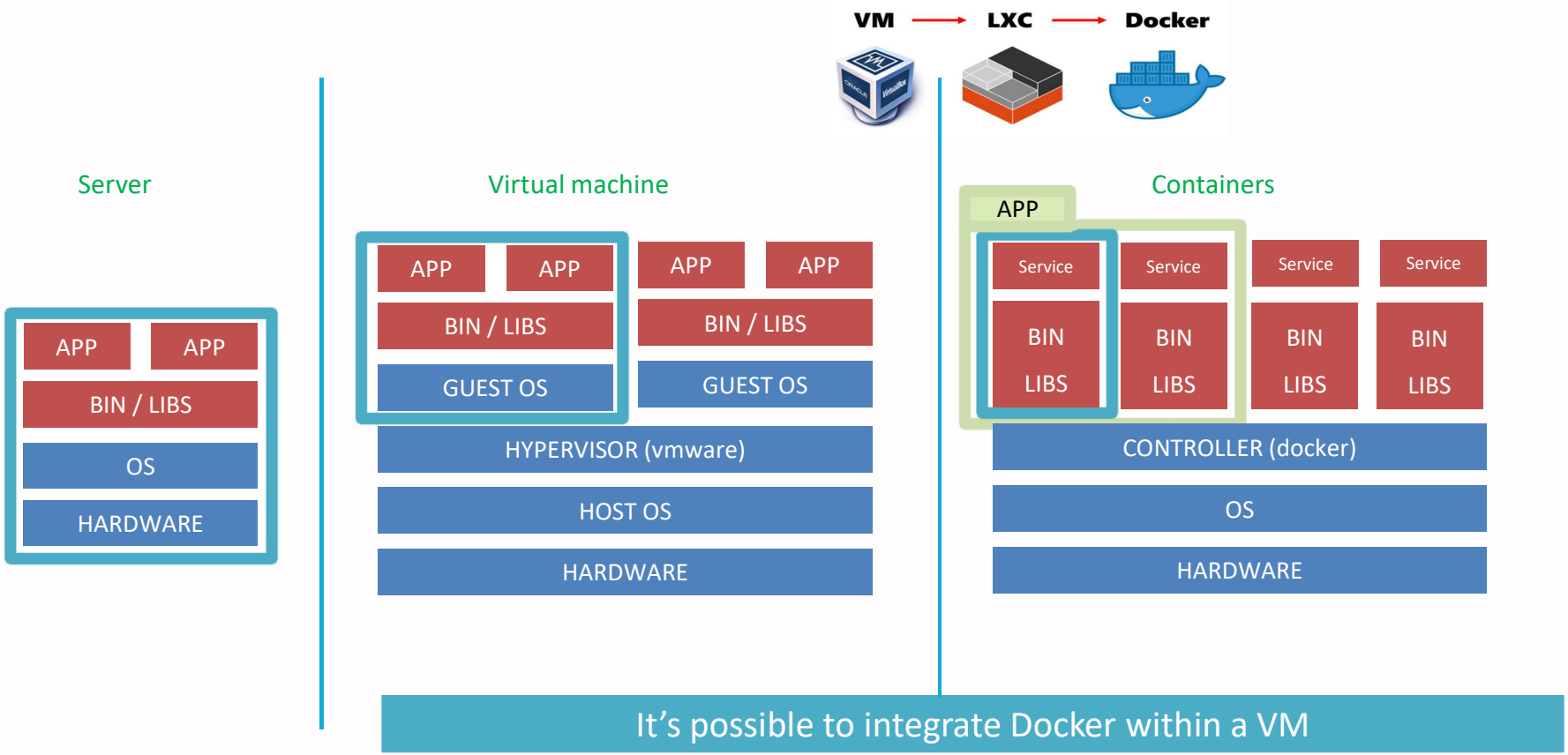
Question ?

In your company is the use of containers a common practice?

The use of containers concerns :

1. Applications ?
2. Tools ?
3. Both ?
4. None ?

Docker benefits for testing



What you will learn

- Docker Toolbox or Docker Desktop ?
- How to setup a Selenium-Grid instance
- How to build your own docker image
- How to manage a service with docker compose
- How to integrate your execution in a CI/CD pipeline

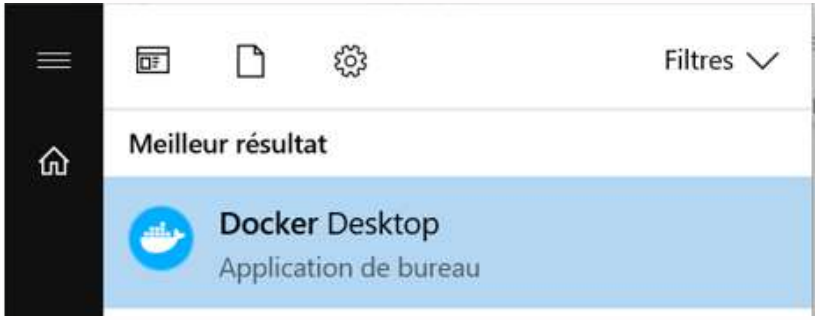
Windows Docker installation

- Docker Toolbox
  - For old configuration (Hyper-V not supported by system and hardware)
  - Needs to install Virtualbox to install a VM based on Linux

- Docker Desktop
  - For new configuration with Hyper-V
  - A server docker engine Linux/windows is integrated
  - It’s easy !!

Docker installation

- After installation and reboot, start the docker Desktop



- A whale icon appears in the task bar

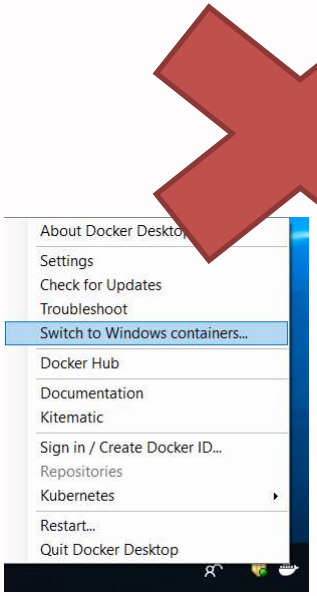
Docker Installation

- Use from a Powershell command (administrator)

#>docker version

```
PS D:\temp\GRID> docker version
Client: Docker Engine - Community
 Version:           19.03.2
 API version:       1.40
 Go version:        go1.12.8
 Git commit:        6a30dfc
 Built:             Thu Aug 29 05:26:49 2019
 OS/Arch:           windows/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.2
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.8
  Git commit:       6a30dfc
  Built:            Thu Aug 29 05:32:21 2019
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          v1.2.6
  GitCommit:        894b81a4b802e4eb2a91d1ce216b8817763c29fb
 runc:
  Version:          1.0.0-rc8
  GitCommit:        425e105d5a03fabd737a126ad93d62a9eeede87f
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
```

Don’t switch to windows container agent




```
temp\GRID> docker version
ocker Engine - Community
                    19.03.2
 sion:              1.40
 sion:              go1.12.8
 Git commit:        6a30dfc
 Built:             Thu Aug 29 05:26:49 2019
 OS/Arch:           windows/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.2
  API version:      1.40 (minimum version 1.24)
  Go version:       go1.12.8
  Git commit:       6a30dfc
  Built:            Thu Aug 29 05:39:49 2019
  OS/Arch:          windows/amd64
  Experimental:     false
```

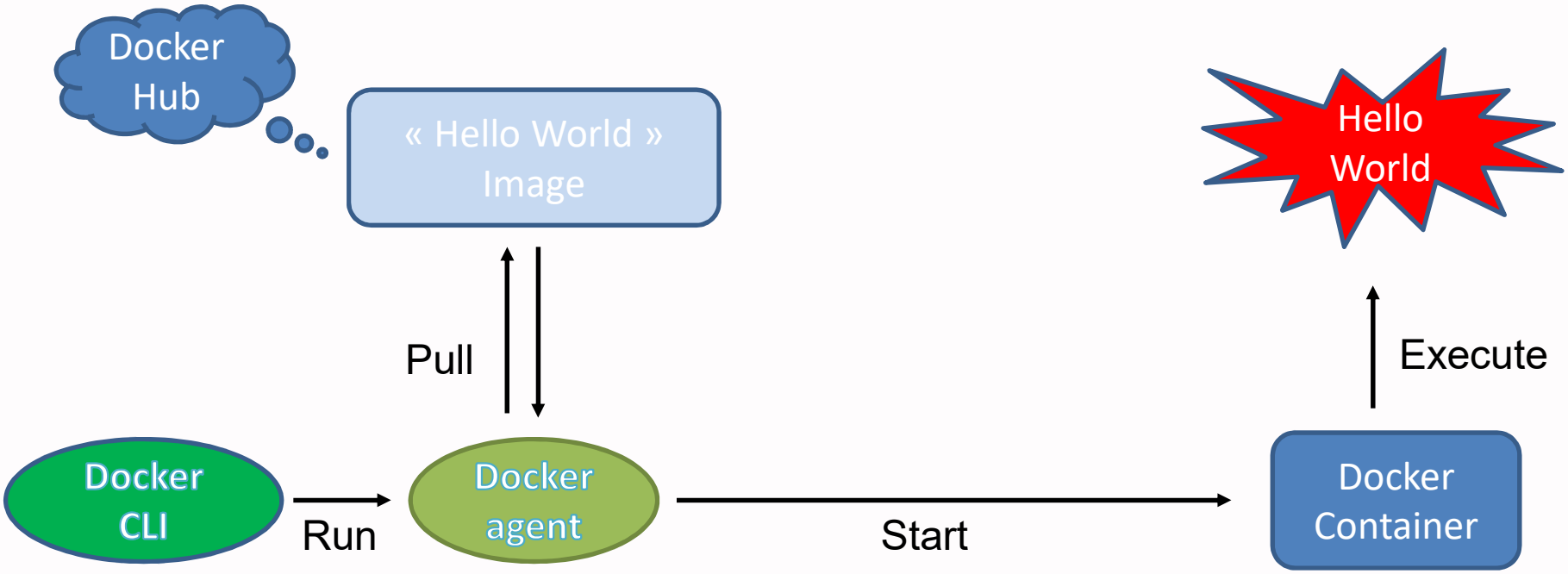
https://hub.docker.com/search?q=&type=image

2,693,531 available images, Only 19,482 available images tagged for engine Windows

Running out our first container

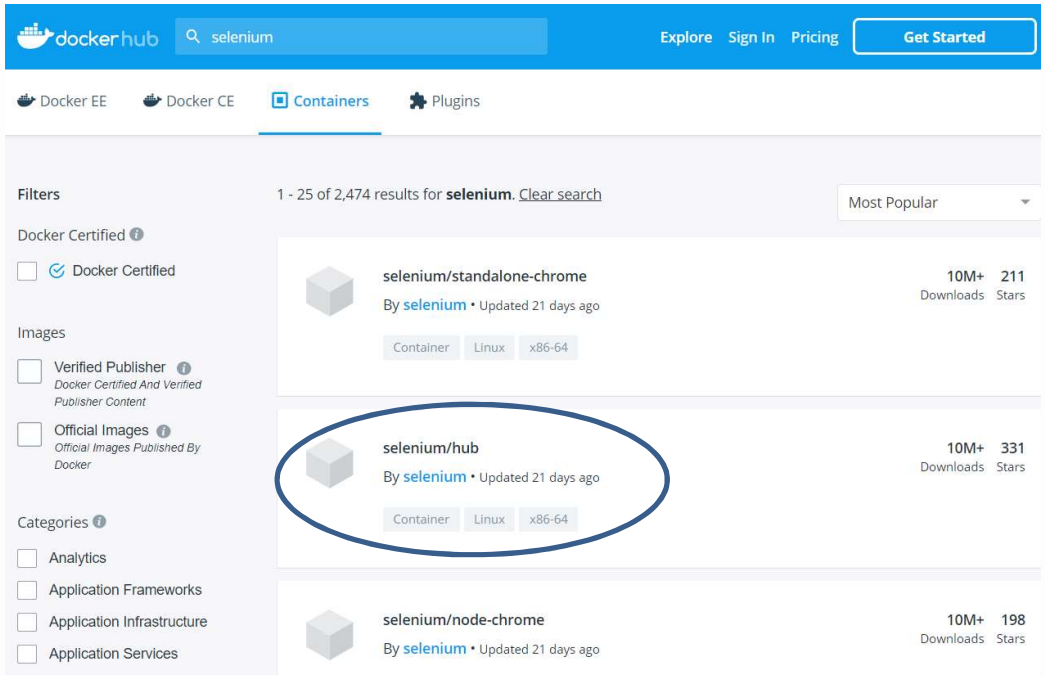
```
PS D:\temp\GRID> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:b8ba256769a0ac28dd126d584e0a2011cd2877f3f76e093a7ae560f2a5301c00
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

What happened ?



Docker repositories for images

- Docker images can be found localy, or pulled from a repository
- Repository can be hosted in the cloud or on premise on a dedicated server.



7th UCAAT User Conference on Advanced Automated Testing

ETSI

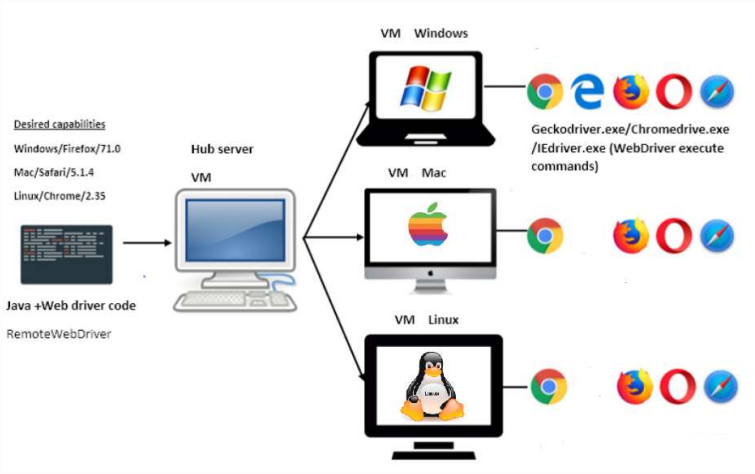
DOCKER FOR WINDOWS

Selenium-Grid setup

sogeti Part of Capgemini

Lab N°1 : Selenium-Grid

- Selenium-Grid allows distributed test execution.
- It consists of a hub and one or more nodes



- Your tests can be run against
  - multiple browsers
  - multiple versions of browser
  - browsers running on different operating systems.

Lab N°1 : Selenium-Grid

- The problem is that :
  - Setup is hard and long-time consuming
  - Configuration management needs efforts
  - Execution requires large resources

- How Docker can help us manage this ?

  →Example : Create a HUB with a NODE based on existing image.

Lab N°1 : Selenium-Grid

1. Create a network

   docker network create grid

   network

2. Create a Hub

   Expose port

   Name container

   Run background

   docker run --detach --publish 4444:4444 --net grid --name **selenium-hub** selenium/hub:3.141.59-titanium

   images

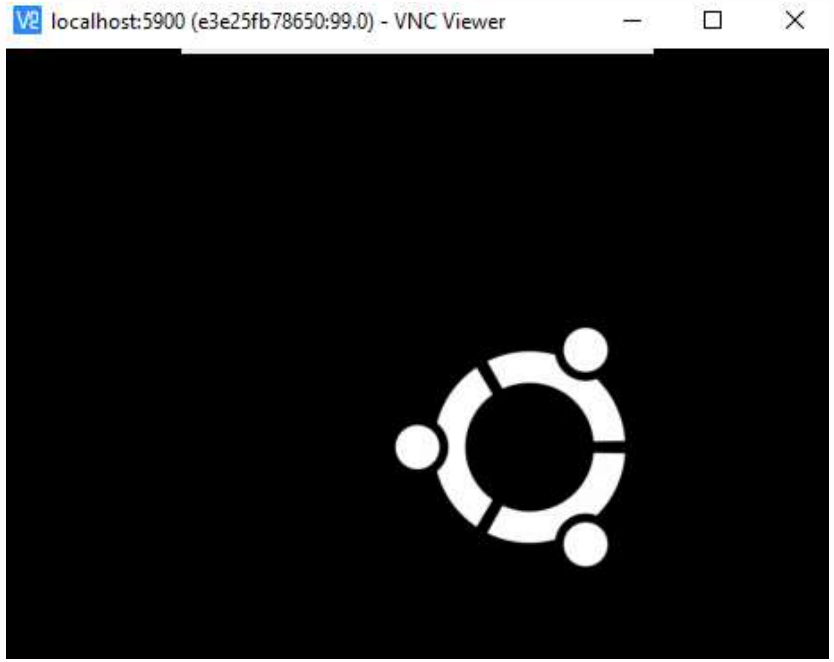
3. Create a Node (Chrome + Vnc)

   docker run -d -p 5900:5900 --net grid --name selenium-node-chrome -e HUB_HOST=selenium-hub -e NODE_MAX_INSTANCES=5 -e NODE_MAX_SESSION=1 -e VNC_NO_PASSWORD=1 -v /dev/shm:/dev/shm selenium/node-chrome-debug:3.141.59-titanium

   Hostname = name of hub container

Lab N°1 : Selenium-Grid

Your first grid is already up and running !



Lab N°1 : Selenium-Grid

7th UCAAT User Conference on Advanced Automated Testing

ETSI

DOCKER FOR WINDOWS

Docker image building

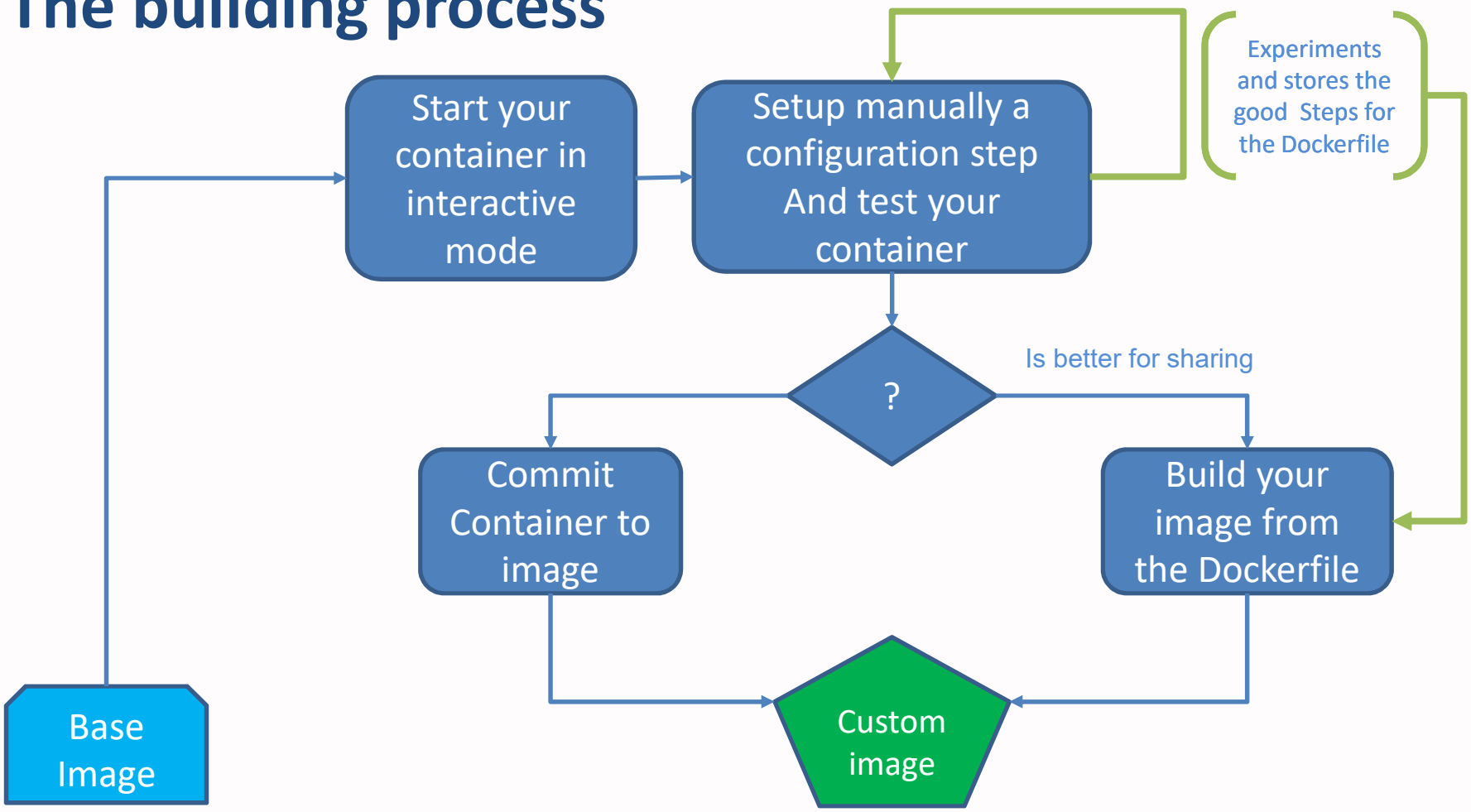
sogeti Part of Capgemini

Docker image

- A Docker image is a pre-built environment for a certain service or tool
- It's a collection of files, libraries and configuration files that build up an environment
- The main source of docker images is the **docker store**

But what if you need to create your own image ?

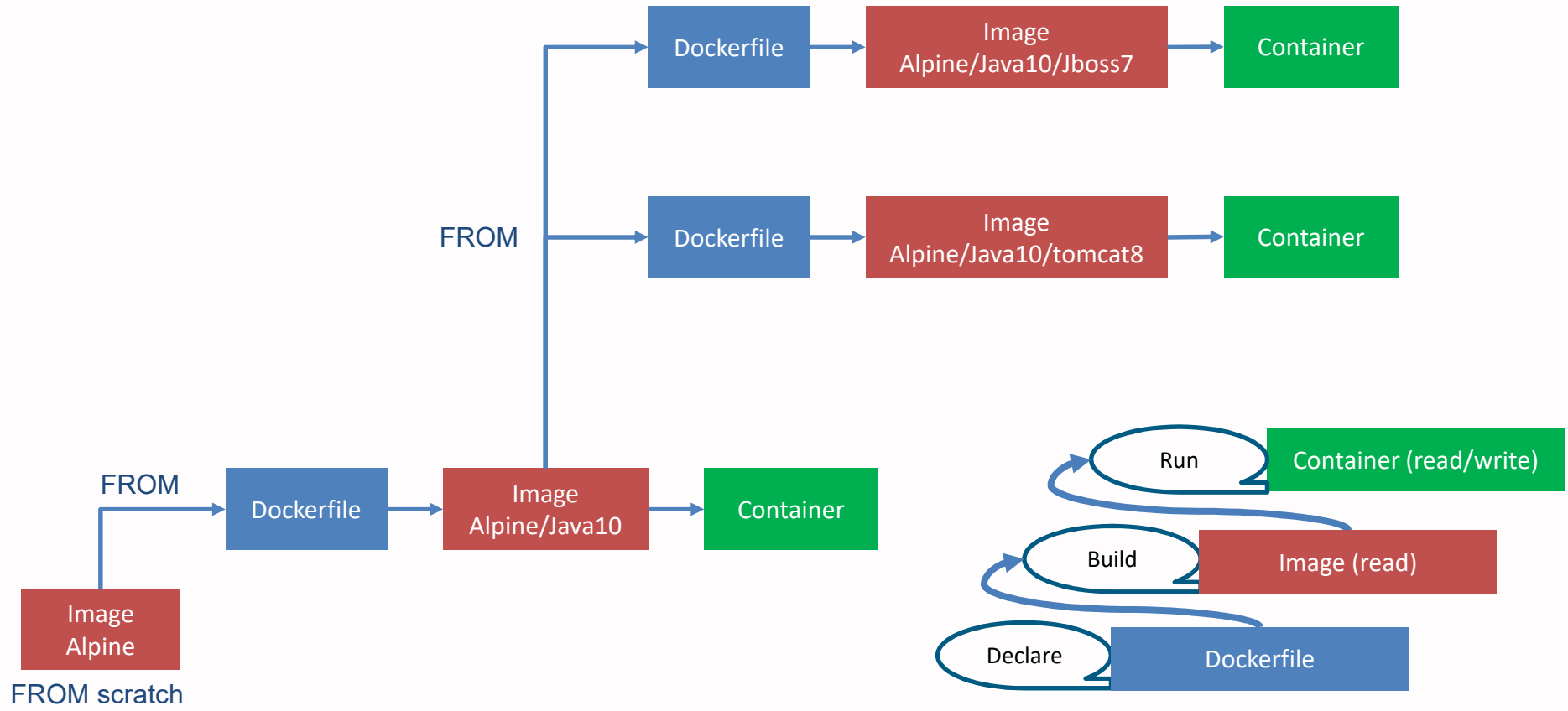
The building process



Dockerfile

- Dockerfile content : command instructions
- Docker CLI to build image from Dockerfile

Scalability, modularity



7th UCAAT User Conference on Advanced Automated Testing

ETSI

DOCKER FOR WINDOWS

Robot Framework image building

sogeti Part of Capgemini

Lab N°2 : Robot FW container experimenting

- How to create a robot Framework container

- We have to :
  1. choose a base image (alpine : minimalist linux librarie)
  2. Identify all the required steps to meet our objective

Lab N°2 : Robot FW container experimenting

- The **alpine image** will be our starting point
  - With an access to RFW tests files
  - With an access to GRID Selenium

- 4 identified steps needed to have an instance available
  - Install python3
  - Upgrade PIP
  - Install RobotFramework
  - Install SeleniumLibrary

Lab N°2 : Robot FW container experimenting

Run an alpine container in **interactive mode**

```
// Get IP of DockerNAT (accessible since localhost and containers) to access on Selenium GRID
$ipGrid=$(Get-NetIPAddress -InterfaceAlias 'vEthernet (DockerNAT)' | %{$_.IPAddress})

// set up container (--interactive –tty)
docker run -it --name devRBF -v d:/temp/RBF:/tmp/RBF --add-host="seleniumgrid:"$ ipGrid alpine

// Copy Robot FW tests files in our local filesystem d:/temp/RBF/tests/
```

testSuiteLog.robot.txt

testSuiteSelenium.robot.txt

Lab N°2 : Robot FW container experimenting

- We are now connected to the container

// From the container, verify the access to grid selenium

more /etc/hosts

apk --no-cache add curl

curl http://seleniumgrid:4444/wd/hub/status

// From the container, verify the access to the robot framewekork tests files

ls /tmp/RBF/tests/

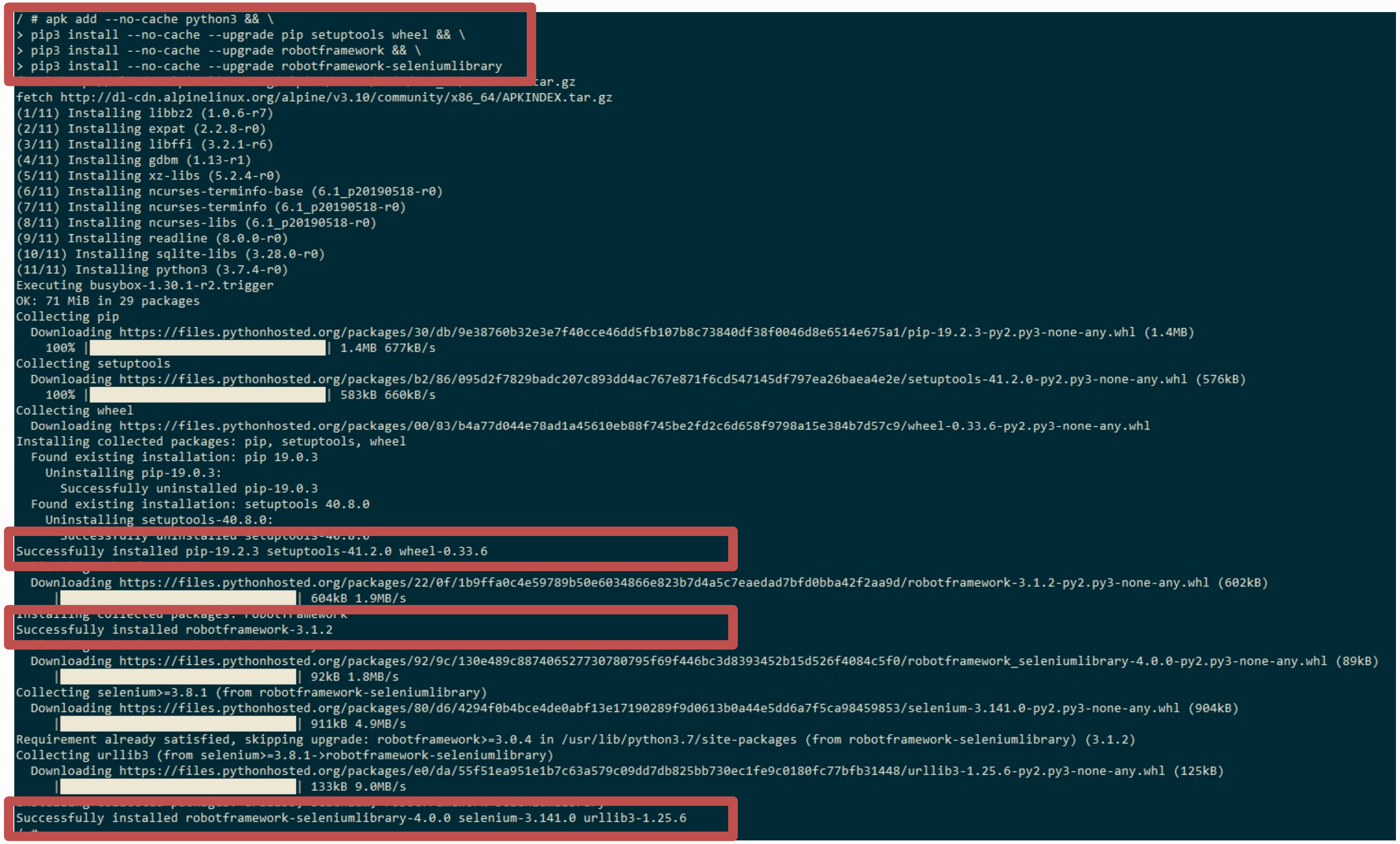
```
/ # ls /tmp/RBF/tests/
testSuiteLog.robot       testSuiteSelenium.robot
/ #
```

Lab N°2 : Robot FW container experimenting

- The next step is the installation of
  - Python3
  - Robot Framework
  - Selenium library for Robot FW

```
apk add --no-cache python3 && \
        pip3 install --no-cache --upgrade pip setuptools wheel && \
        pip3 install --no-cache --upgrade robotframework && \
        pip3 install --no-cache --upgrade robotframework-seleniumlibrary
```

Lab N°2 : Robot FW container experimenting



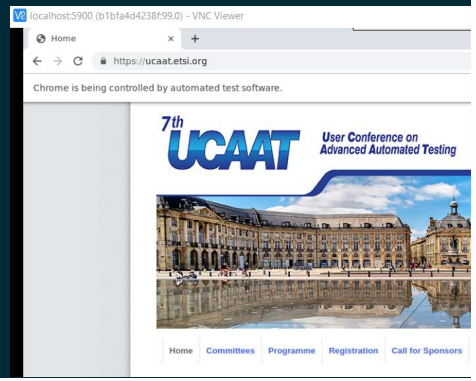
```
/ # apk add --no-cache python3 && \
> pip3 install --no-cache --upgrade pip setuptools wheel && \
> pip3 install --no-cache --upgrade robotframework && \
> pip3 install --no-cache --upgrade robotframework-seleniumlibrary
                                                                            ar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.10/community/x86_64/APKINDEX.tar.gz
(1/11) Installing libbz2 (1.0.6-r7)
(2/11) Installing expat (2.2.8-r0)
(3/11) Installing libffi (3.2.1-r6)
(4/11) Installing gdbm (1.13-r1)
(5/11) Installing xz-libs (5.2.4-r0)
(6/11) Installing ncurses-terminfo-base (6.1_p20190518-r0)
(7/11) Installing ncurses-terminfo (6.1_p20190518-r0)
(8/11) Installing ncurses-libs (6.1_p20190518-r0)
(9/11) Installing readline (8.0.0-r0)
(10/11) Installing sqlite-libs (3.28.0-r0)
(11/11) Installing python3 (3.7.4-r0)
Executing busybox-1.30.1-r2.trigger
OK: 71 MiB in 29 packages
Collecting pip
  Downloading https://files.pythonhosted.org/packages/30/db/9e38760b32e3e7f40cce46dd5fb107b8c73840df38f0046d8e6514e675a1/pip-19.2.3-py2.py3-none-any.whl (1.4MB)
    100% |████████████████████████████████| 1.4MB 677kB/s
Collecting setuptools
  Downloading https://files.pythonhosted.org/packages/b2/86/095d2f7829badc207c893dd4ac767e871f6cd547145df797ea26baea4e2e/setuptools-41.2.0-py2.py3-none-any.whl (576kB)
    100% |████████████████████████████████| 583kB 660kB/s
Collecting wheel
  Downloading https://files.pythonhosted.org/packages/00/83/b4a77d044e78ad1a45610eb88f745be2fd2c6d658f9798a15e384b7d57c9/wheel-0.33.6-py2.py3-none-any.whl
Installing collected packages: pip, setuptools, wheel
  Found existing installation: pip 19.0.3
    Uninstalling pip-19.0.3:
      Successfully uninstalled pip-19.0.3
  Found existing installation: setuptools 40.8.0
    Uninstalling setuptools-40.8.0:
      Successfully uninstalled setuptools-40.8.0
Successfully installed pip-19.2.3 setuptools-41.2.0 wheel-0.33.6
  Downloading https://files.pythonhosted.org/packages/22/0f/1b9ffa0c4e59789b50e6034866e823b7d4a5c7eaedad7bfd0bba42f2aa9d/robotframework-3.1.2-py2.py3-none-any.whl (602kB)
    |████████████████████████████████| 604kB 1.9MB/s
Installing collected packages: robotframework
Successfully installed robotframework-3.1.2
  Downloading https://files.pythonhosted.org/packages/92/9c/130e489c887406527730780795f69f446bc3d8393452b15d526f4084c5f0/robotframework_seleniumlibrary-4.0.0-py2.py3-none-any.whl (89kB)
    |████████████████████████████████| 92kB 1.8MB/s
Collecting selenium>=3.8.1 (from robotframework-seleniumlibrary)
  Downloading https://files.pythonhosted.org/packages/80/d6/4294f0b4bce4de0abf13e17190289f9d0613b0a44e5dd6a7f5ca98459853/selenium-3.141.0-py2.py3-none-any.whl (904kB)
    |████████████████████████████████| 911kB 4.9MB/s
Requirement already satisfied, skipping upgrade: robotframework>=3.0.4 in /usr/lib/python3.7/site-packages (from robotframework-seleniumlibrary) (3.1.2)
Collecting urllib3 (from selenium>=3.8.1->robotframework-seleniumlibrary)
  Downloading https://files.pythonhosted.org/packages/e0/da/55f51ea951e1b7c63a579c09dd7db825bb730ec1fe9c0180fc77bfb31448/urllib3-1.25.6-py2.py3-none-any.whl (125kB)
    |████████████████████████████████| 133kB 9.0MB/s
Successfully installed robotframework-seleniumlibrary-4.0.0 selenium-3.141.0 urllib3-1.25.6
```

Lab N°2 : Robot FW container experimenting

- The container has been successfully upgraded with the RBF components
- Verify the execution of a Robot FW tests on the Selenium GRID

robot --outputdir /tmp/RBF/output --xunit TEST-result.xml --debugfile TEST-debug.txt /tmp/RBF/tests/testSuiteSelenium.robot

```
/ # robot --outputdir /tmp/RBF/output --xunit TEST-result.xml --debugfile TEST-debug.txt /tmp/RBF/tests/testSuiteSelenium.robot
==============================================================================
testSuiteSelenium
==============================================================================
testCaseUCAAT
seleniumGrid=   http://seleniumgrid:4444/wd/hub
capabilities=   {'browserName': 'chrome', 'version': '', 'platform': 'ANY', 'goog:chromeOptions': {'extensions': [], 'args': ['test
STEP GOT TO URL=       https://ucaat.etsi.org/
STEP GET TITLE= Home
STEP SCREENSHOT=       /tmp/RBF/output/UCAAT.png
testCaseUCAAT                                                         | PASS |
------------------------------------------------------------------------------
testSuiteSelenium                                                     | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
==============================================================================
Debug:   /tmp/RBF/output/TEST-debug.txt
Output:  /tmp/RBF/output/output.xml
XUnit:   /tmp/RBF/output/TEST-result.xml
Log:     /tmp/RBF/output/log.html
Report:  /tmp/RBF/output/report.html
/ #
```



Lab N°2 : Robot FW container experimenting

- Test the container behaviour

  Start / Exit container with interactive mode

  Start / Exec command line / Stop container

- You can conserve or delete the experimental container

Lab N°2 : Robot FW container experimenting

Lab N°3 : Robot FW image creation

- Update the dockerfile with the lines:

```
FROM alpine
RUN apk add --no-cache python3 && \
            pip3 install --no-cache --upgrade pip setuptools wheel && \
            pip3 install --no-cache --upgrade robotframework && \
            pip3 install --no-cache --upgrade robotframework-seleniumlibrary && \
            pip3 freeze
WORKDIR /tmp/RBF/tests/
ENTRYPOINT ["robot", "--outputdir", "/tmp/RBF/output", "--xunit", "TEST-result.xml", "--debugfile", "TEST-debug.txt"]
CMD [".."]
```

- Build new image

cd D:/temp/RBF/build/ (directory of the dockerfile)

docker build -t alpine/rbf:latest . (create image from dockerfile)

Lab N°3 : Robot FW image creation

• Run container from new image Robot FW

Commande line = ENTRYPOINT (fixed) + CMD (can be overwrited, by default launch all test in directory)

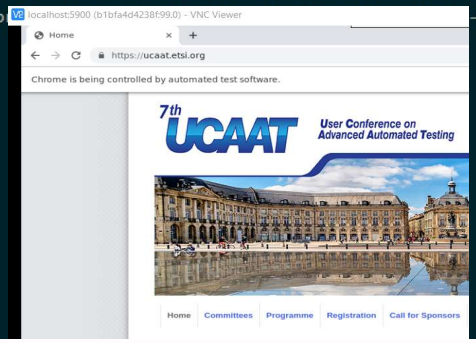
--rm (remove) : delete container after this running

docker run --rm -v d:/temp/RBF:/tmp/RBF --add-host="seleniumgrid:"$ipGrid alpine/rbf

or

docker run --rm -v d:/temp/RBF:/tmp/RBF --add-host="seleniumgrid:"$ipGrid alpine/rbf testSuiteSelenium.robot

```
PS D:\temp\RBF\build> docker run --rm -v d:/temp/RBF:/tmp/RBF --add-host="seleniumgrid:"$ipGrid alpine/rbf testSuiteSelenium.robot
==============================================================================
testSuiteSelenium
==============================================================================
testCaseUCAAT
seleniumGrid=   http://seleniumgrid:4444/wd/hub
capabilities=   {'browserName': 'chrome', 'version': '', 'platform': 'ANY', 'goog:chromeOptio
STEP GOT TO URL=        https://ucaat.etsi.org/
STEP GET TITLE= ucaat.etsi.org
STEP SCREENSHOT=        /tmp/RBF/output/UCAAT.png
| PASS |
------------------------------------------------------------------------------
testSuiteSelenium                                                     | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
==============================================================================
Debug:   /tmp/RBF/output/TEST-debug.txt
Output:  /tmp/RBF/output/output.xml
XUnit:   /tmp/RBF/output/TEST-result.xml
Log:     /tmp/RBF/output/log.html
Report:  /tmp/RBF/output/report.html
PS D:\temp\RBF\build>
```



Lab N°3 : Robot FW image creation

- **Result under D:\temp\RBF\output**

```
PS D:\temp\RBF\output> Get-ChildItem -Name
log.html
output.xml
report.html
TEST-debug.txt
TEST-result.xml
UCAAT.png
PS D:\temp\RBF\output>
```

TEST testCaseLog
Full Name: Tmp.testSuite.testCaseLog
Start / End / Elapsed: 20190831 14:14:24.560 / 20190831 14:14:24.562 / 00:00:00.002
Status: PASS (critical)
KEYWORD BuiltIn.Log EXECUTE ROBOFRAMEWORK TEST WITH DOCKER

TEST testCaseUCAAT
Full Name: Tmp.testSuite.testCaseUCAAT
Start / End / Elapsed: 20190831 14:14:24.563 / 20190831 14:14:39.530 / 00:00:14.967
Status: PASS (critical)
KEYWORD testSetup
KEYWORD openUCAAT
Start / End / Elapsed: 20190831 14:14:25.928 / 20190831 14:14:38.960 / 00:00:13.032
KEYWORD SeleniumLibrary.Go To ${url}
Documentation: Navigates the active browser instance to the provided url.
Start / End / Elapsed: 20190831 14:14:25.930 / 20190831 14:14:27.552 / 00:00:01.622
14:14:25.931 INFO Opening url 'https://ucaat.etsi.org/'
KEYWORD ${titre} = SeleniumLibrary.Get Title
KEYWORD BuiltIn.Log ${titre}
KEYWORD SeleniumLibrary.Capture Page Screenshot
Documentation: Takes screenshot of the current page and embeds it into log file.
Start / End / Elapsed: 20190831 14:14:28.069 / 20190831 14:14:28.958 / 00:00:00.889
14:14:28.958 INFO

Lab N°3 : Robot FW image creation

7th UCAAT User Conference on Advanced Automated Testing

ETSI

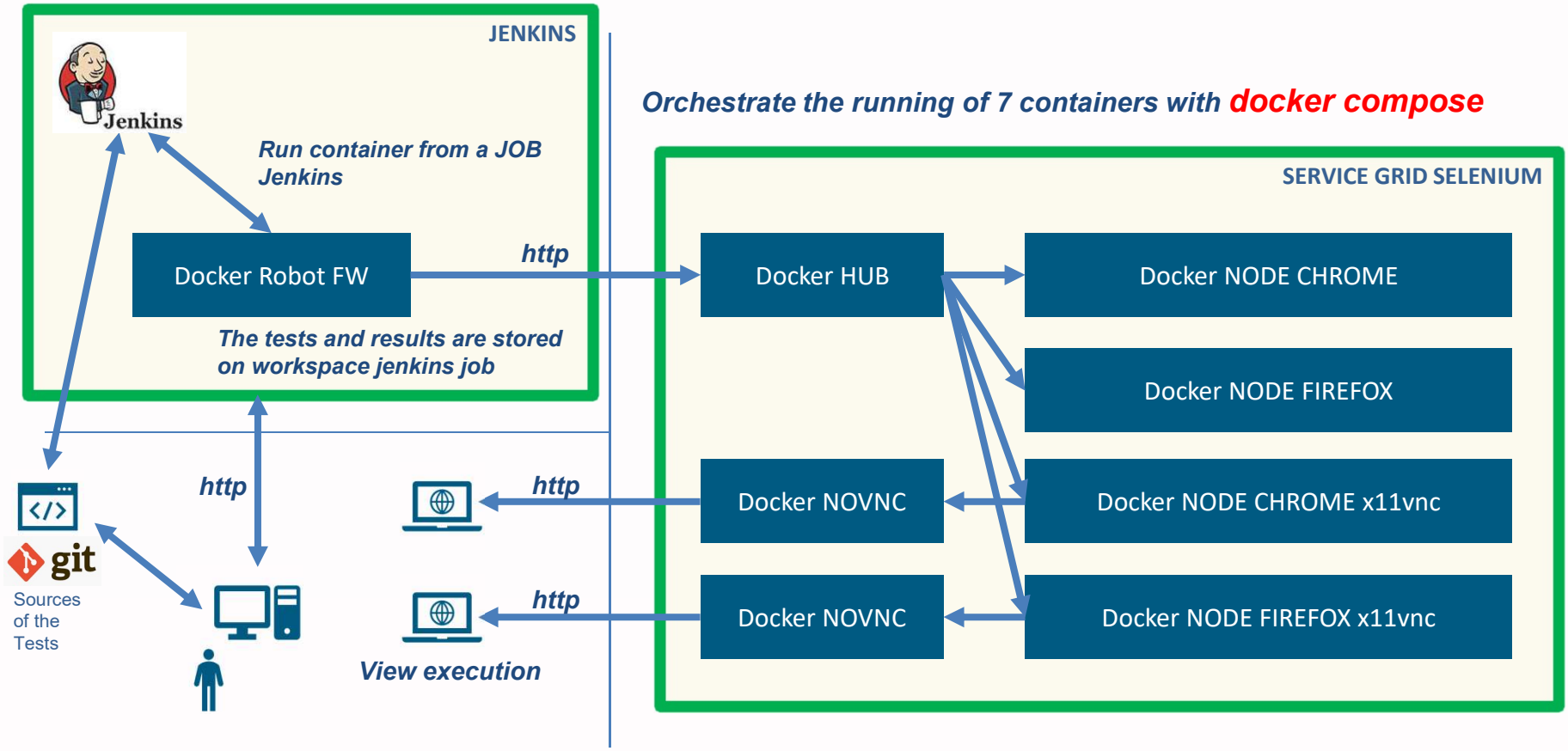
DOCKER COMPOSE

A service needs severals containers

Example for the GRID Selenium

sogeti Part of Capgemini

Lab N°4 : Docker Compose and Jenkins



Lab N°4 : Docker Compose

- Docker compose helps us, through a YAML file, to manage different containers as a service
- It can specifies:
  - Dependencies between containers
  - Volumes, ports

docker-compose.yml.txt

```
version: "3"
services:
  selenium-hub:
    image: selenium/hub:3.141.59-titanium
    container_name: selenium-hub
    ports:
      - "4444:4444"
  chrome:
    image: selenium/node-chrome:3.141.59-titanium
    container_name: selenium-node-chrome
    volumes:
      - /dev/shm:/dev/shm
    depends_on:
      - selenium-hub
    environment:
      - HUB_HOST=selenium-hub
      - HUB_PORT=4444
      - NODE_MAX_INSTANCES=4
      - NODE_MAX_SESSION=3
  firefox:
    image: selenium/node-firefox:3.141.59-titanium
    container_name: selenium-node-firefox
    volumes:
      - /dev/shm:/dev/shm
    depends_on:
      - selenium-hub
    environment:
      - HUB_HOST=selenium-hub
      - HUB_PORT=4444
      - NODE_MAX_INSTANCES=4
      - NODE_MAX_SESSION=3
    deploy:
      replicas: 2
```

Lab N°4 : Docker Compose

- Run Service as demon

docker-compose up –d

- Stop Service

docker-compose down

→ It's easy !!!

```
PS D:\temp\GRID> docker-compose up -d
Creating selenium-hub ... done
Creating selenium-node-chrome        ... done
Creating selenium-node-firefox       ... done
Creating selenium-node-chrome-debug  ... done
Creating selenium-node-firefox-debug ... done
Creating vnc-firefox-debug           ... done
Creating vnc-chrome-debug            ... done
PS D:\temp\GRID>
```

```
PS D:\temp\GRID> docker-compose down
Stopping vnc-chrome-debug            ... done
Stopping vnc-firefox-debug           ... done
Stopping selenium-node-firefox-debug ... done
Stopping selenium-node-chrome-debug  ... done
Stopping selenium-node-firefox       ... done
Stopping selenium-node-chrome        ... done
Stopping selenium-hub                ... done
Removing vnc-chrome-debug            ... done
Removing vnc-firefox-debug           ... done
Removing selenium-node-firefox-debug ... done
Removing selenium-node-chrome-debug  ... done
Removing selenium-node-firefox       ... done
Removing selenium-node-chrome        ... done
Removing selenium-hub                ... done
Removing network grid_grid
PS D:\temp\GRID>
```

- Force execution on the NODE Firefox VNC

docker run --rm -v d:/temp/RBF:/tmp/RBF --add-host="seleniumgrid:"$ipGrid alpine/rbf --variable FORCE_VNC:TRUE --variable BROWSER:FIREFOX testSuiteSelenium.robot

The test script sets the capabilities so that the Selenium HUB directs the test to the expected Selenium NODE.

Lab N°4 : Jenkins

- Run Service as demon (see the installation of Jenkins in annex)

docker run -u root -d -p 8080:8080 -p 50000:50000 -v D:/Docker/Jenkins:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock -v D:/temp/RBF:/tmp/RBF --name jenkins jenkinsci/blueocean:1.18.1-bcc31d32159f

- Or else restart the service

docker start jenkins

Lab N°4 : Use Solution

- Finally, the tester use only http connection

Services	Urls
Consult the state of Selenium GRID	http://localhost:4444/grid/console
Jenkins to launch and schedule the tests and consult the results	http://localhost:8080
View in real time test execution on chrome	http://localhost:8081/vnc.html
View in real time test execution on firefox	http://localhost:8082/vnc.html

The final solution is made on servers, the urls will not be on localhost, there will certainly be a reverse proxy (nginx) to provide the various services without the ports being visible.

To improve, it's need to mount a volume between the Grid Nodes and Jenkins if the web application has a transfer file feature. (upload file or download file)

Lab N°4 : Docker Compose and Jenkins

Go further for management and remote deployment

- Kubernetes/Docker-Swarm : are open-source container-orchestration system for automating application deployment, scaling, and management.

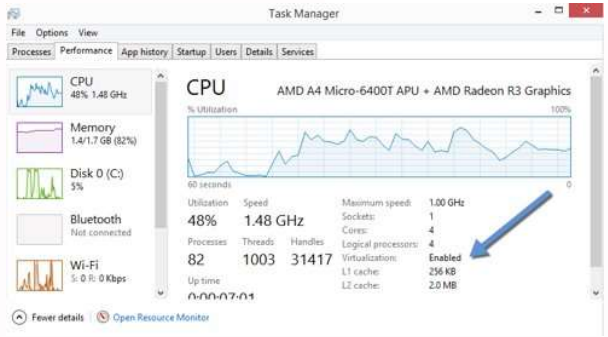
https://thenewstack.io/kubernetes-vs-docker-swarm-whats-the-difference/

- Ansible is an open-source software provisioning, configuration management, and application-deployment tool. (SSH Agent-Less)

Congratulation
We managed to use tools without installing them, but only through docker

ANNEX

Docker Desktop installation



- System requirements
  - **Windows 10 64-bit: Pro**, Enterprise, or Education (Build 15063 or later).
  - Hyper-V and Containers Windows features must be enabled.

- Hardware prerequisites
  - 64 bit processor with Second Level Address Translation (SLAT)
  - 4GB system RAM
  - BIOS-level hardware virtualization support must be enabled in the BIOS settings.

- Documentation https://docs.docker.com/docker-for-windows/install/

Docker best practices and security

https://docs.docker.com/develop/develop-images/dockerfile_best-practices/
https://www.docker.com/blog/intro-guide-to-dockerfile-best-practices/
https://snyk.io/blog/10-docker-image-security-best-practices/
https://cloud.google.com/blog/products/gcp/7-best-practices-for-building-containers

Example for docker file:
Use official/certified image
Package a single application per container
Minimize the number of layers
Don’t use default USER (root)
Use no-cache

Attached : Installation of Jenkins from image

- **Create un local workdir** : *D:/Docker/Jenkins (persistence of installation and data)*
- ***Launch this command***

docker run -u root --rm -d -p 8080:8080 -p 50000:50000 -v D:/Docker/Jenkins:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock -v D:/temp/RBF:/tmp/RBF --name jenkins jenkinsci/blueocean:1.18.1-bcc31d32159f

- ***Copy le password that is in D:\Docker\Jenkins\secrets\initialAdminPassword***
- ***Continue the installation since*** http://localhost:8080/
  - ***Install the suggested plugins***
  - ***Create an admin user***

- https://github.com/auriuki/jenkins-docker
- https://hub.docker.com/r/jenkinsci/blueocean
- https://stackoverflow.com/questions/36607394/error-opening-robot-framework-log-failed