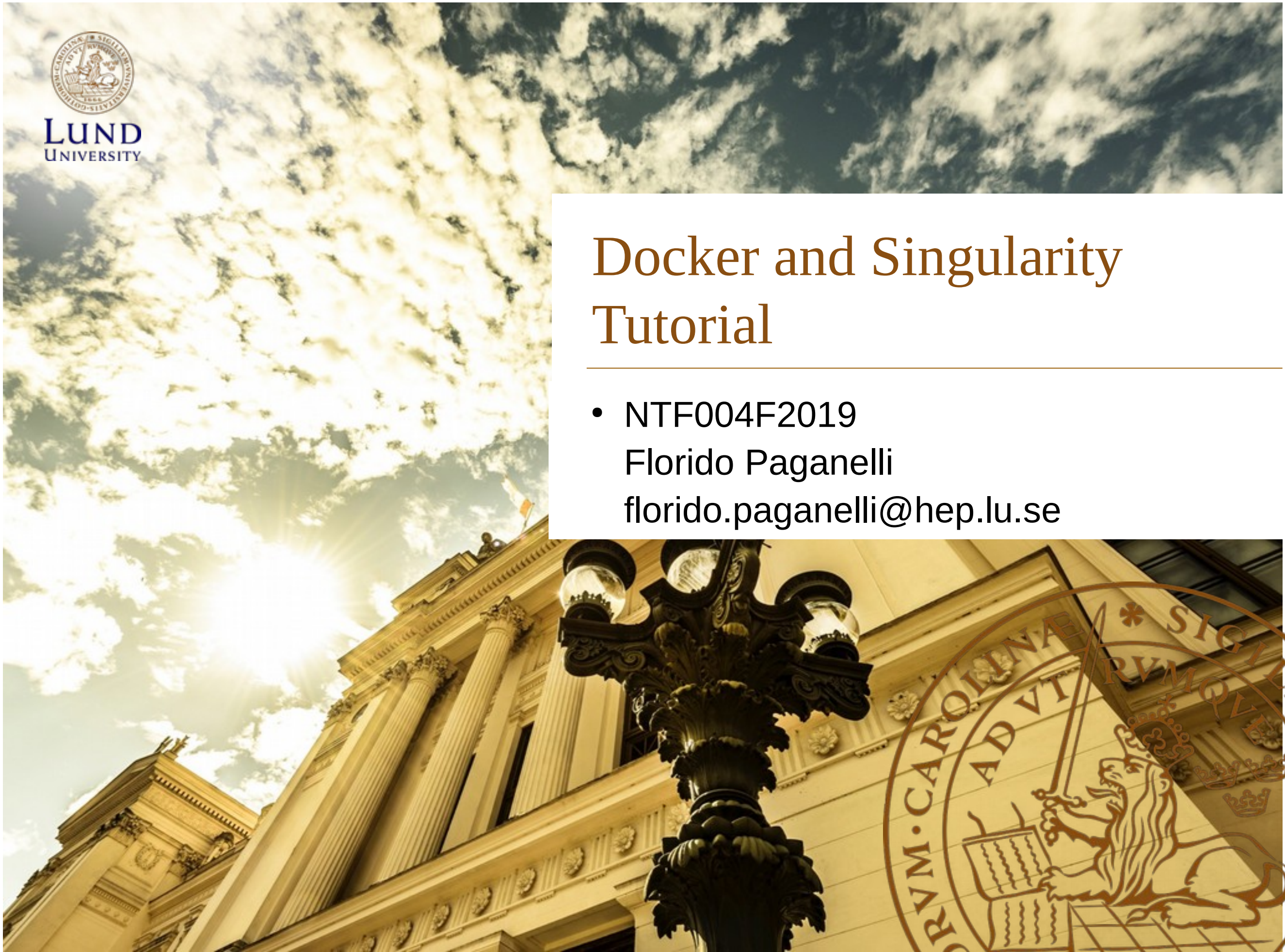




LUND
UNIVERSITY

Docker and Singularity Tutorial

- NTF004F2019
Florido Paganelli
florido.paganelli@hep.lu.se



Problem of the day

- I was given some python code that checks the requirements for some scientific software
- I run it in my institution cluster and notice that the cluster is **not** up to the requirement
- After a chat with the sysadmin, it is **not possible to provide such requirements** by the means of the existing operating system on the cluster
- The sysadmin suggests to provide a **linux container** that can be run with **singularity** and it has all the required dependencies.

Task 0 – Learn Singularity

- Play a bit with singularity. It is only installed on n12. Login to n12 **after** you login to the iridium frontend. Commands:

```
ssh -X yourusername@pptest-iridium.lunarc.lu.se  
byobu # to keep your session alive  
ssh -X n12
```

- Follow this introductory tutorial

https://www.sylabs.io/guides/3.1/user-guide/quick_start.html#overview-of-the-singularity-interface

- There are mistakes in the tutorial. I suggest you use ' (single quote) instead of " (double quotes)
- Some containers will not work, because Iridium kernel is too old. For example the lolcow container from *sylabsed* will not work, but this one will:

```
library://godlovedc/demo/lolcow
```

Task 1 - investigation

- Understand what are the requirements in the python script `projrequirements.py`

It is located on github here:

https://github.com/floridop/NTF004F-2019/blob/master/docker_sing_tutorial/projrequirements.py

- For your convenience, I copied it on the Iridium cluster at

`/nfs/shared/pp/dccourse2019/docker_singularity_190423/`

- Try to run it, look at the code.

Task 2 – creating a docker image

- **On your laptop**, get the python script and create a *Dockerfile* to *build an image* containing all the required software to run it.
- You can pass the python script to the image in various ways, by copying it inside the image or by mounting a folder inside with the `-v` option:

```
docker run -it -v \  
/path/where/script/is/on/host:/path/in/container \  
dockerhubaccount/containerNTF04F:0.1
```

- Iridium is an old cluster with an old kernel. A container runs with the kernel of the HOST OS, that means you cannot use an OS image that expects a newer kernel. For example an Ubuntu 18.04 will not work, but an Ubuntu 16.04 will.

Task 2 – creating an image: strategy

- Start the built image in interactive mode and test that it runs the python script with no errors.
- Check for mistakes, fix the Dockerfile and repeat.
- Hints:
 - It's always better to have a workdir defined in the image, as shown by various tutorials.
 - if you're using an Ubuntu machine, some commands might be helpful:

```
apt-cache search nameofpackage
```

searches for the package *nameofpackage*

```
apt-get install nameofpackage [nameofpackage1]...
```

installs a list of packages named *nameofpackageX*

Task 3 – upload to docker hub

- Upload your image to docker hub.
 - You will need a docker hub account
 - Verify that the image is publicly available using docker hub website
 - **SECURITY WARNING IF YOU USE THIS IN REAL LIFE:**
the docker password is stored in clear text on your machine. This is really **bad**. Follow docker's documentation suggestion to secure your password. This is not part of this course and you don't have to do it, but if you decide to use it for serious business you should definitely do.
<https://hackernoon.com/getting-rid-of-docker-plain-text-credentials-88309e07640d>
<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Task 4 - use your Docker image with Singularity on Iridium

- Download your image from docker hub with the command

```
singularity pull docker://your-docker-id/your-docker-image-name:tag
```

- Test that the image works on Iridium by simply starting it as a command:

```
./myimagename.sif
```

- Test that the image can run *projrequirements.py* without problems:

```
./myimagename.sif ./projrequirements.py
```




LUND
UNIVERSITY

Errata:

- Containers CANNOT run a kernel other than the system kernel, as I erroneously stated during the lecture 16/Apr.