

# Advanced Quinta Usage

# Things you probably know:

## - Accessing the frontal machine

```
outside:~$ ssh user@XX.XX.XX.XX
```

## - Verifying machines' status

```
r11:~$ quinta status -a
```

## - Choosing deployment images

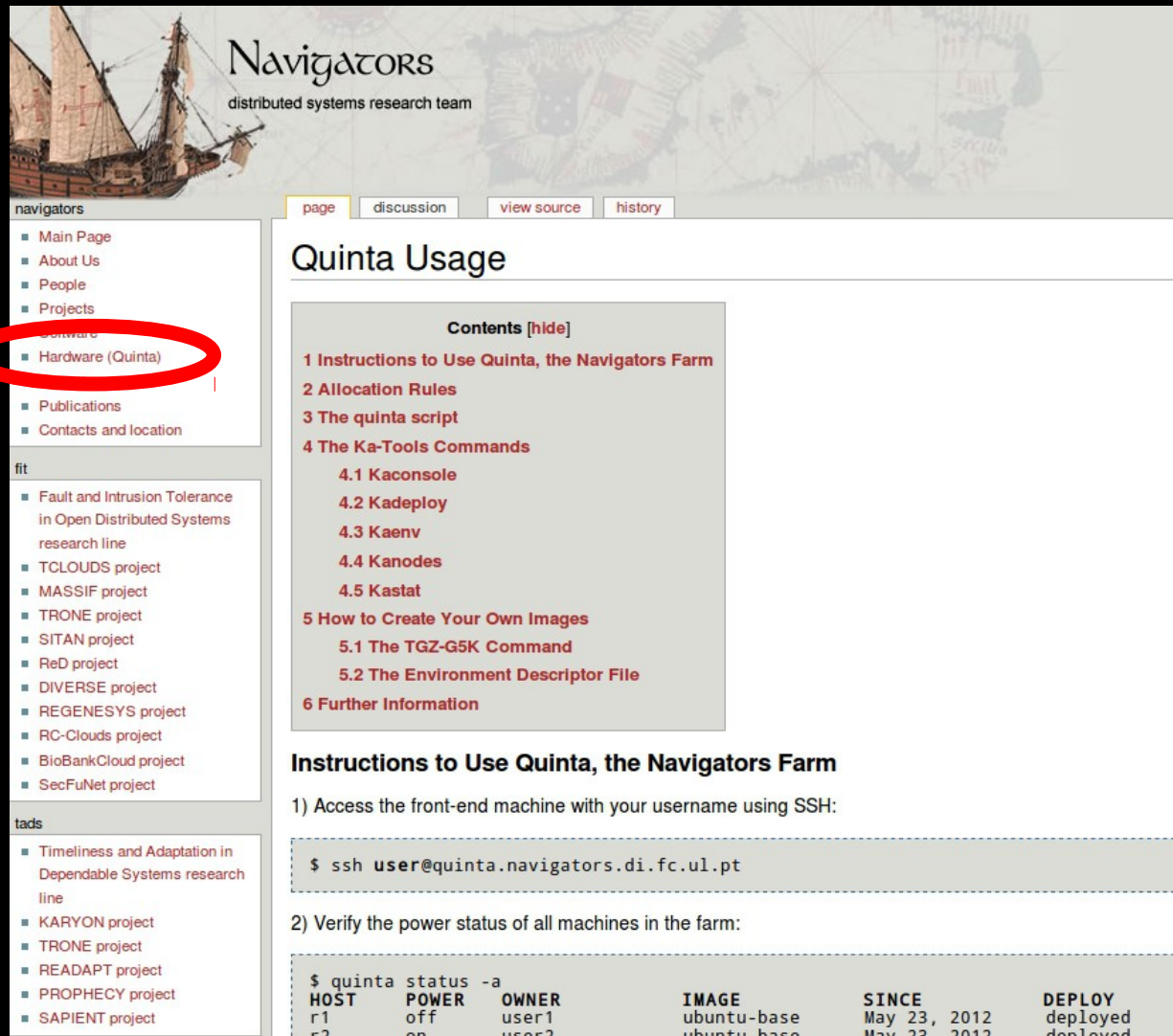
```
r11:~$ kaenv3 -l
```

## - Deploying nodes

```
r11:~$ kadeploy3 -e env-name -f nodes_file
```

# If you still don't know it:

## - Quinta usage tutorial



**Navigators**  
distributed systems research team

page | discussion | view source | history

### Quinta Usage

**Contents [hide]**

- 1 Instructions to Use Quinta, the Navigators Farm
- 2 Allocation Rules
- 3 The quinta script
- 4 The Ka-Tools Commands
  - 4.1 Kaconsole
  - 4.2 Kadeploy
  - 4.3 Kaenv
  - 4.4 Kanodes
  - 4.5 Kastat
- 5 How to Create Your Own Images
  - 5.1 The TGZ-G5K Command
  - 5.2 The Environment Descriptor File
- 6 Further Information

### Instructions to Use Quinta, the Navigators Farm

1) Access the front-end machine with your username using SSH:

```
$ ssh user@quinta.navigators.di.fc.ul.pt
```

2) Verify the power status of all machines in the farm:

```
$ quinta status -a
HOST    POWER  OWNER  IMAGE          SINCE          DEPLOY
r1      off    user1  ubuntu-base   May 23, 2012  deployed
r2      on     user2  ubuntu-base   May 23, 2012  deployed
```

**Left Sidebar:**

- navigators
  - Main Page
  - About Us
  - People
  - Projects
  - Software
  - Hardware (Quinta)**
  - Publications
  - Contacts and location
- fit
  - Fault and Intrusion Tolerance in Open Distributed Systems research line
  - TCLOUDS project
  - MASSIF project
  - TRONE project
  - SITAN project
  - ReD project
  - DIVERSE project
  - REGENESYS project
  - RC-Clouds project
  - BioBankCloud project
  - SecFuNet project
- tads
  - Timeliness and Adaptation in Dependable Systems research line
  - KARYON project
  - TRONE project
  - READAPT project
  - PROPHECY project
  - SAPIENT project

# Next Steps:

- 1 - Improving access
- 2 - Sharing nodes
- 3 - Improving deployments
- 4 - Using NFS

# 1 - Improving access:

## - Reducing command line

```
outside:~$ ssh username@XX.XX.XX.XX
```

```
outside:~$ #Create a ~/.ssh/config file:
```

```
Host quinta
```

```
  User username
```

```
  HostName quinta.navigators.di.fc.ul.pt
```

```
  StrictHostKeyChecking no
```

```
outside:~$ ssh quinta
```

# 1 - Improving access:

## - Accessing nodes directly

```
outside:~$ ssh quinta
```

```
r11:~$ ssh root@s9
```

```
s9:~$
```

```
outside:~$ #Add a rule to the ~/.ssh/config file:
```

```
Host *.quinta
```

```
    User root
```

```
    ProxyCommand ssh username@quinta.navigators.di.fc.ul.pt "nc -q
```

```
10 \$(basename %h .quinta) %p"
```

```
    StrictHostKeyChecking no
```

```
outside:~$ ssh s9.quinta
```

```
s9:~$
```

# 1 - Improving access:

## - “Passwordless” root in nodes

```
outside:~$ ssh s9.quinta  
root@s9.quinta's password: *****  
s9:~$ logout
```

```
outside:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub s9.quinta  
root@s9.quinta's password: *****  
s9:~$ logout  
outside:~$ #same as scp + cat id.pub >> auth_keys
```

```
outside:~$ ssh s9.quinta  
s9:~$ logout
```

# 1 - Improving access:

- “Passwordless” root in nodes  
2<sup>nd</sup> case

```
r11:~$ kadeploy3 -e img-name -f nodes-file  
-k ~/.ssh/id_rsa.pub
```

```
r11:~$ ssh root@s9  
s9:~$ logout
```

```
r11:~$ #It will use the public key from your user in R11
```



## 2 - Sharing nodes:

```
r11:~$ quinta status s12
```

HOST	POWER	OWNER	IMAGE	SINCE	DEPLOY
s12	on	user1	ubuntu-12.04	Oct 28, 2013	deployed

```
r11:~$ quinta update -u user2 s12
```

```
r11:~$ quinta status s12
```

HOST	POWER	OWNER	IMAGE	SINCE	DEPLOY
s12	on	user2	ubuntu-12.04	Oct 30, 2013	deployed

## Problems:

- **user2 need to ask user1 every time he want to use the node**
- **Process from user1 can still be running**  
(should we reboot the machine with the update?)

# 3 - Improving deployments:

## - Creating a new image

```
user@r11:~$ kadeploy3 -e ubuntu-12.04 -m r5
```

```
user@r11:~$ ssh root@r5
```

```
root@r5:~$ passwd root
```

```
root@r5:~$ apt-get install foo
```

```
root@r5:~$ echo "" > /etc/udev/rules.d/70-persistent-net.rules
```

```
root@r5:~$ tgz-g5k user@r11:~/my-ubuntu-foo.tgz
```

```
root@r5:~$ exit
```

# 3 - Improving deployments:

## - Registering a new image

```
user@r11:~$ vim myDescriptor
```

```
name : my-ubuntu-foo
```

```
version : 1
```

```
description : Ubuntu Lucid Lynx image with Java.
```

```
author : User (user@lasige.di.fc.ul.pt)
```

```
tarball : /home/users/user/my-ubuntu-foo.tgz|tgz
```

```
postinstall : /home/deploy/postinstalls/post4all.tgz|tgz|traitement.ash
```

```
kernel : /boot/vmlinuz-2.6.32-21-server
```

```
initrd : /boot/initrd.img-2.6.32-21-server
```

```
fdisktype : 83
```

```
filesystem : ext3
```

```
environment_kind : linux
```

```
visibility : shared
```

```
user@r11:~$ kaenv3 -a myDescriptor
```

```
Computing the md5sum for /home/users/user/my-ubuntu-foo.tgz
```

```
Computing the md5sum for /home/deploy/postinstalls/post4all.tgz
```

```
user@r11:~$ kadeploy3 -e my-ubuntu-foo -f nodes-file
```

# 3 - Improving deployments:

## - Adding public key:

```
user@r11:~$ kadeploy3 -e my-ubuntu-foo -f nodes-file  
-k ~/.ssh/id_rsa.pub
```

## 3 - Improving deployments:

### - Running a script automatically:

```
user@r11:~$ kadeploy3 -e my-ubuntu-foo -f nodes-file  
-k ~/.ssh/id_rsa.pub -s myScript.sh
```

### Example of automatic tasks:

- Start the service
- Start clients and store results in a local file
- Copy results to R11 (for persistence)
- Execute the previous three steps indefinitely until I decide to shut down the nodes
- Start a test: simply deploy a new cluster (no more software installation and file retrieval)
- Sharing: simply turn off nodes

# 3 - Improving deployments:

## - **Best practices:**

- Deploy a basic image
- Install your software
- Create a new image
- Register this image
- Create a post-deployment script
- Deploy passing -k and -s arguments
- Turn off machines after running the experiments

## - **Example:**

- Script to execute a Hadoop workload in Grid'5000:  
[https://www.grid5000.fr/mediawiki/index.php/Run\\_Hadoop\\_On\\_Grid%275000#Script\\_config.sh](https://www.grid5000.fr/mediawiki/index.php/Run_Hadoop_On_Grid%275000#Script_config.sh)

# 4 - Using NFS:

```
user@r11:~$ kadeploy3 -e ubuntu-12.04-nfs -m r5
```

```
user@r11:~$ ssh user@r5
```

```
user@r5:~$ ls
```

#Your files from R11 appear in your home (~user) at r5

## - Best practices:

- Deploy a NFS image
- Copy your input files to a local folder (e.g.: /opt/\*)
- Run your software with the copied files
- Store output in a local file
- At the end of your experiment, copy again to NFS folder (~user)