



## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

# Interfaces PaaS

**Cécile Cavet**

`cecile.cavet at apc.univ-paris7.fr`

Centre François Arago (FACe), Laboratoire APC, Université Paris Diderot



29 Avril 2016



# Plan

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

**1** Contexte

**2** Outils pour le cloud

**3** TP1 : Docker

**4** TP2 : cluster avec Ansible

**5** TP3 : cluster avec SlipStream



# PaaS

## Contexte

### Cas d'utilisation

- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque



Figure: <http://fr.slideshare.net/clintedmonson/windows-azure-jumpstart>.



# Cas d'utilisation du cloud à l'APC

## Contexte

### Cas d'utilisation

Docker  
Cluster Beowulf  
Cluster Hadoop

### Outils pour le cloud

#### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

#### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

#### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## Projets scientifiques :

- LISAPathfinder (2015) / eLISA (2034) / PTA (2006) : machines virtuelles collaboratives, cluster Beowulf à la demande, R&T Docker.
- Euclid/LSST (~2020) : cluster Hadoop à la demande (prospective Big Data).

➔ **Besoin de cloud Infrastructure-as-a-Service (IaaS) et Plateform-as-a-Service (PaaS).**



# Cas d'utilisation du cloud à l'APC

## Contexte

### Cas d'utilisation

Docker

Cluster Beowulf

Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration

Service

Registre privé

Docker-compose

### TP2 : cluster avec Ansible

Configuration

Play

Torque

### TP3 : cluster avec SlipStream

Configuration

Exécution

Torque

## Ressources IaaS :

- Docker : 1 MV.
- Cluster Beowulf :
  - 1 MV master + x MV workers.
  - stockage : NFS sur un volume de master ou GlusterFS sur des disques externes (éphémères ou persistents).
- Cluster Hadoop :
  - MV → même chose que Beowulf.
  - stockage : HDFS sur des disques externes.



# Docker



- **Conteneurisation des applications.**

- **Services :**

- docker, docker-machine (gestion de MV), docker-compose (multi-conteneur), docker-swarm (cluster de conteneur)...
- [Docker Hub](#), le registre public de Docker (partage des images avec leurs Dockerfiles).

➔ **voir TP1**

## Contexte

Cas d'utilisation

### Docker

Cluster Beowulf

Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration

Service

Registre privé

Docker-compose

### TP2 : cluster avec Ansible

Configuration

Play

Torque

### TP3 : cluster avec SlipStream

Configuration

Exécution

Torque



# Fonctionnement de Docker

## Contexte

Cas d'utilisation

### Docker

Cluster Beowulf

Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration

Service

Registre privé

Docker-compose

### TP2 : cluster avec Ansible

Configuration

Play

Torque

### TP3 : cluster avec SlipStream

Configuration

Exécution

Torque

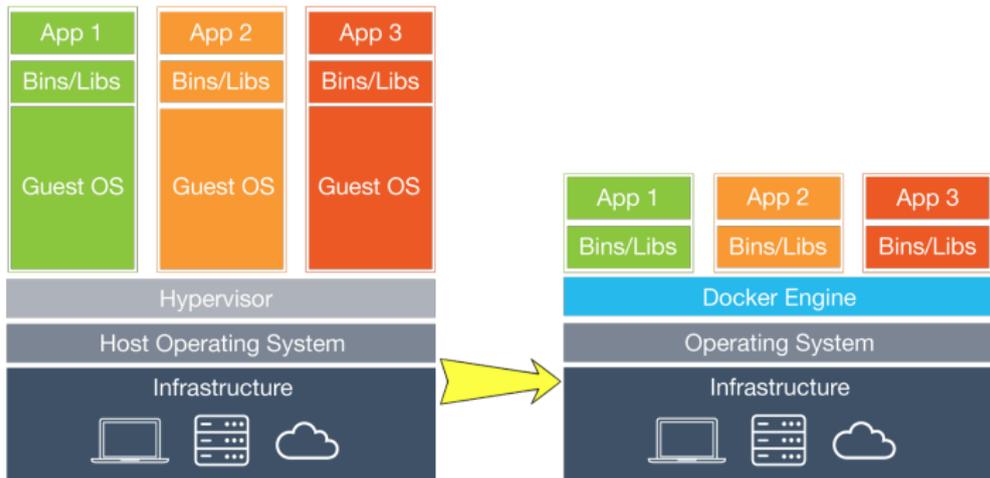


Figure: MV vs conteneur @Docker.

# Cluster Beowulf



## Contexte

Cas d'utilisation

Docker

**Cluster Beowulf**

Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration

Service

Registre privé

Docker-compose

### TP2 : cluster avec Ansible

Configuration

Play

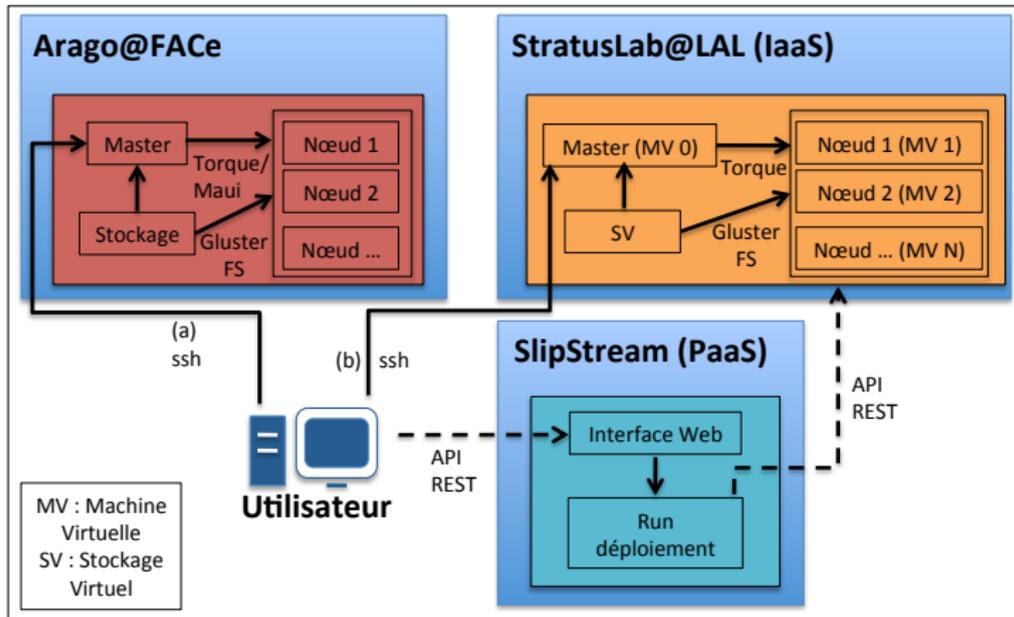
Torque

### TP3 : cluster avec SlipStream

Configuration

Exécution

Torque



➔ voir TP2 et TP3

# Cluster Hadoop



## Contexte

Cas d'utilisation

Docker

Cluster Beowulf

**Cluster Hadoop**

## Outils pour le cloud

### TP1 : Docker

Configuration

Service

Registre privé

Docker-compose

### TP2 : cluster avec Ansible

Configuration

Play

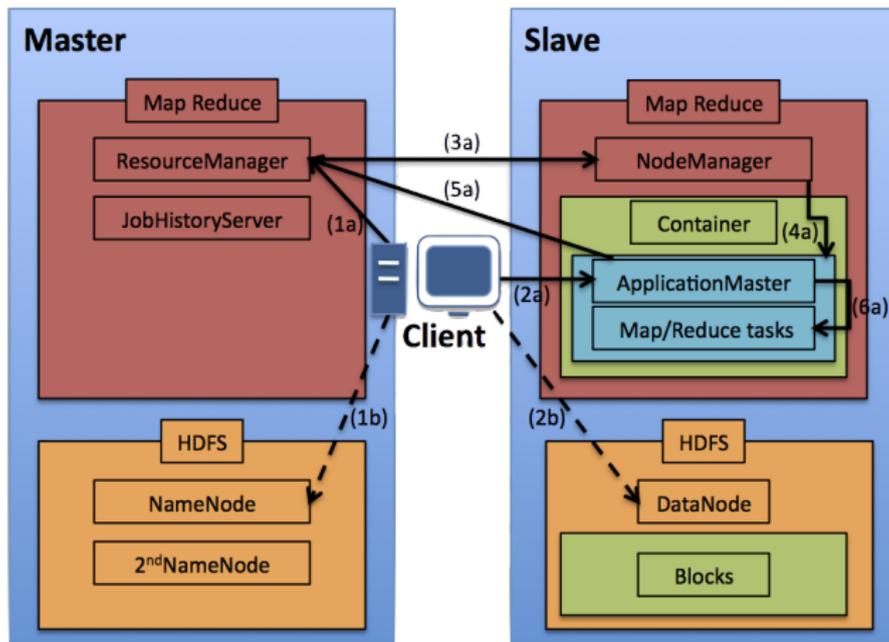
Torque

### TP3 : cluster avec SlipStream

Configuration

Exécution

Torque





# Outils pour le cloud

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## But :

- Déploiement automatique.
- Passage à l'échelle ("scalability") des infrastructures virtuelles :
  - Horizontal : variation du nombre de VMs.
  - Vertical : variation des ressources individuels des VMs.
- Interopérabilité des infrastructures de cloud :
  - 1 seule (Web) interface conviviale (API REST).
  - Multi-connecteurs : commerciaux et open-source.
  - Multi-site : en cours...



# Outils pour le cloud

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

**Table 1** Virtual Infrastructure deployment tools comparison

	Wrangler	Whirr	Claudia	Cloudinit.d	SlipStream	Recont. Broker	Cloud Formation	Vagrant
Systems support	EC2, Eucalyptus, OpenNebula	EC2	OpenNebula	EC2, Nimbus	EC2, OpenStack, OpenNebula, Azure, ...	EC2, Nimbus	EC2	VirtualBox, VMWare, EC2
VMs Context.	Yes	Yes	No	Yes	Yes	Yes	No	Yes
High level Context. Lang.	No	No	-	No	No	Yes	-	Yes
VM pre-config.	Yes	Yes	Yes	Yes	No	No	No	Yes
Simple deploy lang.	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Extensible	Yes	Yes	Yes	No	No	No	No	Yes
Catalog of VMIs	No	No	No	No	No	No	No	No
Elasticity Mgmt	Yes	No	Yes	No	No	Yes	Yes <sup>1</sup>	Yes

Figure: Dynamic Management of Virtual Infrastructures, M. Caballer, I. Blanquer, G. Molto, C. de Alfonso, J. Grid Computing (2015).

➔ **Multitude d'outils.**



# Outils pour le cloud : **SlipStream**



## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

- Compagnie privée **Sixsq**.
- Modèle commercial : connecteurs payants pour clouds payants (cad AWS...).
- Communautés : toutes.
- Connecteurs : AWS EC2, StratusLab, OpenStack, OCCl, CloudSigma, CloudStack...
- Language : script shell.

➔ voir TP3



# Déploiement automatique avec **SlipStream** : cluster virtuel

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

The screenshot shows the SlipStream web interface. The top navigation bar is red and contains the SlipStream logo, 'Dashboard', 'App Store', 'Workspace', 'Help', and 'cavet'. The main content area is dark grey and displays the 'Torque' project page. The page title is 'Torque Project' with a sub-header 'Version: 44 - Initial version of this project.' Below the title is a breadcrumb trail: 'Home / test / Torque / 44'. On the right side, there are buttons for 'New project', 'New component', and 'New application'. Below the main content area, there is a table with the following data:

Name	Description	Owner	Version
torque		cavet	48
torque-master		cavet	57
torque-worker		cavet	58

Below the table, there is a section for 'Authorizations' which is currently collapsed.



# Passage à l'échelle avec **SlipStream** : workflow complexe

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

The screenshot displays the SlipStream web interface. The top navigation bar is red and contains the SlipStream logo, 'Dashboard', 'App Store', 'Workspace', 'Help', and 'cavet'. Below the navigation bar is a breadcrumb trail: '/ Images / centos7 / 85'. The main content area is divided into several sections, each with a dropdown arrow on the right:

- Summary
- Cloud Image Identifiers and Image Hierarchy
- Operating System Details
- Cloud Configuration
- Application Parameters
- Application Workflows

The 'Application Workflows' section is expanded, showing a list of workflow steps on the left and a script editor on the right:

- 1. Pre-install
- 2. Install packages
- 3. Post-install
- 4. Deployment
- 5. Reporting
- 6. On VM Add
- 7. On VM Remove

The script editor contains the following text:

```
This script is executed on the deployed machine before SlipStream client and packages are installed.
```

```
1
```

Below the script editor are two more sections with dropdown arrows:

- Runs
- Authorizations



# Interopérabilité avec **SlipStream** : Cloud@FG

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

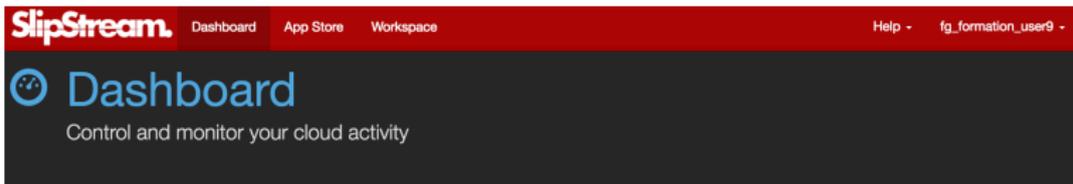
- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque



## Usage



Deployments Virtual Machines

Include inactive runs

ID	Application / Component	Service URL	State	Start Time	Clouds	User	Tags
c467c9a8	centos7 v92		Initializing	27 Apr 2016, 13:46:10 UTC	LAL2	fg_formation_user@	

C. Cavet

Interface PaaS



# SlipStream dans France Grilles

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## Serveur

- Service France Grilles (FG).
- Hébergé au LAL@Orsay :  
<https://slipstream.lal.in2p3.fr>
- Nouvelle version (compatible OpenStack) :  
<https://slipstream-v3.lal.in2p3.fr>
- Documentation :
  - [Sixsq](#).
  - [FG](#).



# Outils pour le cloud : **VM-DIRAC**

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque



- Extension de l'interware DIRAC (Distributed Infrastructure with Remote Agent Control).
- Communauté : Fed Cloud EGI, service FG.
- Fonctionnement : au niveau du job.
- Connecteurs : AWS EC2, OpenStack, OpenNebula, CloudStack, OCCl.



# VM-DIRAC

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

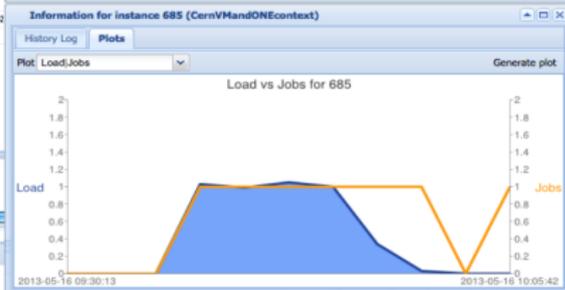
Image	EndPoint	Status	Endpoint VM ID
<input type="checkbox"/> CernVM-voihzoutkay	nova-1-1-r2p3	halted	82142f6b64-4e81-ae18-411c54ed5b
<input type="checkbox"/> CernVMandONEcontext	ooz-0-9-pic	halted	675
<input type="checkbox"/> CernVMandONEcontext	ooz-0-9-pic	halted	676
<input type="checkbox"/> CernVM-voihzoutkay	nova-1-1-r2p3	halted	8085916-e83a-4147-be81-c9c9d77

Update time	Status	Load
2013-05-10 00:35:35	New	0
2013-05-10 00:35:34	Wait_ssh_context	0
2013-05-10 00:37:32	Contextualizing	0
2013-05-10 00:37:57	Running	0
2013-05-10 00:40:35	Running	0.02
2013-05-10 00:41:59	Halted	0

Page 1 of 1 | Items displaying per page: 50 | Show VMs in status: All

Virtual Machines > Browse instances

Update time	Status	Load	Jobs	Files transferred	Bytes transferr...
2013-05-16 09:...	New	0	0	0	0
2013-05-16 09:...	Submitted	0	0	0	0
2013-05-16 09:...	Running	0	0	0	0
2013-05-16 09:...	Running	0.34	1	0	0
2013-05-16 09:...	Running	0.98	1	0	0
2013-05-16 09:...	Running	0.99	1	0	0
2013-05-16 09:...	Running	0.99	1	0	0
2013-05-16 09:...	Running	1	1	0	0
2013-05-16 10:...	Running	0.01	1	0	0
2013-05-16 10:...	Running	0	1	0	0
2013-05-16 10:...	Halted	0	0	0	0





# Outils pour le cloud : **Infrastructure Manager**



## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

- Laboratoire [GRyCAP](#).
- Communauté : Fed Cloud EGI.
- Connecteurs : commerciaux (AWS , Google Cloud, Microsoft Azure), open-source (OpenNebula , OpenStack), Federated (EGI FedCloud (OCCI), FogBow).
- Language : RADL (Resource and Application Description Language) basé sur Ansible.



# IM

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque



Infrastructures

Credentials

RADLs

**Recipes**



List [Add +](#)

Show 10 entries

imuser01 is connected | [Sign out](#) | [Change password](#)

Infrastructure Manager > Recipes

Application	Version	View
ganglia	3.1.0	
gmetad	3.1.0	
MAFFT	7.157	
torque-mom	2.4	
torque-server	2.4	

Showing 1 to 5 of 5 entries

[Previous](#) [Next](#)



# Outils pour le déploiement : **Ansible**



ANSIBLE

- Configuration automatique tous type de serveur.
- Playbook : fichier YAML (format de sérialisation de données).
- Modules et Tasks.
- Rôle : ressources réutilisables.

➔ voir TP2

## Contexte

Cas d'utilisation

Docker

Cluster Beowulf

Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration

Service

Registre privé

Docker-compose

### TP2 : cluster avec Ansible

Configuration

Play

Torque

### TP3 : cluster avec SlipStream

Configuration

Exécution

Torque



# Ansible : play

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

```
[root@ansible-master ~]# ansible-playbook cluster.yml
[
PLAY [master] *****

GATHERING FACTS *****
ok: [134.158.75.235]

TASK: [micafer.torque | include_vars {{ansible_os_family}}.yaml] *****
ok: [134.158.75.235]

TASK: [micafer.torque | Create User {{item.name}}] *****
ok: [134.158.75.235 -> 127.0.0.1] => (item={'password': '$6$Bw64PcHcJj70hGmY$p/DAZc1RsNA7/.8
AfgSvvunSK.NnapwV27wpl02BDfsrsPzRAt6mKbwBwDwZj4IjrjrmQQvPOTP7RFnyHYYSjs/', 'id': 2001, 'name'
: 'user1'})

TASK: [micafer.torque | shell cp /home/{{item.name}}/.ssh/id_rsa.pub /tmp/{{item.name}}_id_r
sa.pub creates=/tmp/{{item.name}}_id_rsa.pub] ***
ok: [134.158.75.235 -> 127.0.0.1] => (item={'password': '$6$Bw64PcHcJj70hGmY$p/DAZc1RsNA7/.8
AfgSvvunSK.NnapwV27wpl02BDfsrsPzRAt6mKbwBwDwZj4IjrjrmQQvPOTP7RFnyHYYSjs/', 'id': 2001, 'name'
: 'user1'})
```



# Et maintenant, les TP...

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque



Figure: @Software Carpentry.



# TP1 : Docker



## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

#### Configuration

Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ Conteneurisation des applications.

### ■ But du TP :

- Créer une MV avec Docker installé.
- Récupérer une image Docker modifiée d'un registre privé.
- Construire un ensemble de conteneurs pour fournir le service Jenkins (intégration continue).



# TP1 : Configuration de Docker

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

#### Configuration

Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ TODO :

- Lancer une MV CentOS7 qui exécute le script `ici` :

```
$ chmod u+x custom_MV.sh
```

```
$ nova boot --key-name=cloudkey --image=centos7  
--flavor=m1.large --user-data ./custom_MV.sh  
docker
```

```
$ ssh -Y centos@ip
```

- Note : la connection se fait avec l'option graphique afin de lancer plus tard un navigateur local à la MV.



# TP1 : Configuration de Docker

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

#### Configuration

Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ TODO :

- Se connecter à la MV et vérifier les logs de Cloud-init jusqu'à la fin de l'installation :

```
$ sudo su
```

```
$ tail -f /var/log/cloud-init-output.log
```

```
...
```

```
Cloud-init v. 0.7.5 finished at ...
```

- Note : l'installation dure 5 min.



# TP1 : service Docker

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration

#### Service

Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

- **TODO** :
- Tester Docker :

```
$ docker ps
$ docker search ubuntu
$ docker pull ubuntu
$ docker images
$ docker run -it ubuntu bash
$ root@id $ cat /etc/issue
Ubuntu 14.04.4 LTS \n \l
```

- Note : l'image Ubuntu provient du [Docker Hub](#), le registre public de Docker. Le *pull* récupère l'image par couche (données et méta-données).



# TP1 : registre privé Docker

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
**Registre privé**  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ TODO :

- Utiliser un registre privé Docker (sur une MV) pour récupérer une image buildée de Jenkins :

```
$ curl -k -X GET \  
https://myregistrydomain.com:443/v2/_catalog  
$ docker pull \  
myregistrydomain.com:443/myjenkins  
$ docker images  
myregistrydomain.com:443/myjenkins latest id...
```

- Note : la fonction *search* n'est pas encore implémentée pour les registres privés.



# TP1 : lancement de conteneurs

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé

#### Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ TODO :

- Lancer les différents conteneurs depuis le répertoire du fichier `*.yml` :

```
$ docker-compose up -d
```

```
$ docker-compose ps
```

Name	Command	State	Ports
-----			
myjenkins	/bin/tini -- /usr/...	Up	0.0.0.0:8080
mysql	/entrypoint.sh mysqld	Up	
sonar	./bin/run.sh	Up	0.0.0.0:9000

- Note : le port de l'interface Web de Jenkins est 8080 et celui de SonarQube est 9000.



# Interface Web de Jenkins

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

Outils pour le cloud

TP1 : Docker

Configuration  
Service  
Registre privé

Docker-compose

TP2 : cluster avec Ansible

Configuration  
Play  
Torque

TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

■ **TODO :**

■ Lancer un navigateur dans la MV :

\$ `firefox &`

Dashboard [Jenkins] x +

localhost:8080 Search

Jenkins

ENABLE AUTO REINDEX

add description

New Item

People

Build History

Manage Jenkins

Credentials

**Welcome to Jenkins!**

Please [create new jobs](#) to get started.

**Build Queue** -

No builds in the queue.

**Build\_Executor\_Status** -

1 idle

2 idle



# Interface Web de SonarQube

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
**Docker-compose**

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

Dashboard [Jenkins] x SonarQube x

0.0.0.0.9000

sonarqube Dashboards Issues Measures Rules Quality Profiles Quality Gates More Log in

Home

Welcome to SonarQube Dashboard  
Since you are able to read this, it means that you have successfully started your SonarQube server. Well done!  
If you have not removed this text, it also means that you have not yet played much with SonarQube. So here are a few pointers for your next step:

- Do you now want to [run analysis](#) on a project?
- Maybe start [customizing dashboards](#)?
- Or simply browse the [complete documentation](#)?
- If you have a question or an issue, please visit the [Get Support](#) page.

PROJECTS

QG	NAME	VERSION	LOC	TECHNICAL DEBT	LAST ANALYSIS
No data					

PROJECTS

No data

- Note : si le conteneur *sonar* est en erreur, stopper *mysql*, relancer *sonar* puis *mysql*.

`$ docker-compose stop/start ...`



# Interface Web de Jenkins

## Contexte

Cas d'utilisation

Docker

Cluster Beowulf

Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration

Service

Registre privé

**Docker-compose**

### TP2 : cluster avec Ansible

Configuration

Play

Torque

### TP3 : cluster avec SlipStream

Configuration

Exécution

Torque

## ■ TODO :

- Pour rendre l'interface publique (inutile pour les IPs publiques non flottantes), créer une interface virtuelle :

```
$ ifconfig eth0:1 134.158.151.XX \  
    netmask 255.255.255.0
```

```
$ docker-compose stop && docker-compose rm -f
```

```
$ vi docker-compose.yml
```

```
myjenkins:
```

```
    ...
```

```
    ports:
```

```
      - "134.158.151.XX:8080:8080"
```



# Interface Web de Jenkins

## Contexte

Cas d'utilisation

Docker

Cluster Beowulf

Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration

Service

Registre privé

**Docker-compose**

### TP2 : cluster avec Ansible

Configuration

Play

Torque

### TP3 : cluster avec SlipStream

Configuration

Exécution

Torque

## ■ TODO :

- Pour rendre l'interface publique, relancer le *docker-compose* :

```
$ docker-compose up -d
```

```
$ docker-compose ps
```

Name	State	Ports
-----		
myjenkins	Up	134.158.151.XX:8080



# TP2 : Ansible

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque



ANSIBLE

- **Automatisation du déploiement post instantiation.**
- **But du TP :**
  - Créer deux MVs (master et worker) avec Ansible installé sur master.
  - Récupérer un rôle Ansible.
  - Exécuter un playbook pour fournir le service Torque (gestionnaire de job).



# TP2 : Configuration d'Ansible

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ TODO :

- Lancer deux MVs CentOS7 dont le master exécute le script [ici](#) :

```
$ nova boot --key-name=cloudkey --image=centos7  
--user-data ./custom_master.sh \  
--flavor=m1.medium ansible-master  
$ nova boot --key-name=cloudkey --image=centos7  
--flavor=m1.medium ansible-worker  
$ ssh centos@ip-ansible-master && sudo su  
$ tail -f /var/log/cloud-init-output.log
```

- Note : toutes les opérations se déroulent depuis *ansible-master* qui est à la fois le noeud Ansible et le noeud master de Torque.



# TP2 : Configuration d'Ansible

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

- **TODO** :
  - Configurer la MV master :

```
$ vi /etc/ansible/hosts  
[cluster]
```

```
134.158.75.XX
```

```
134.158.75.XX
```

```
[master]
```

```
...
```

- Note : la clé publique de master doit être copiée dans `/root/.ssh/authorized_keys` de worker car Ansible communique avec les noeuds par `ssh`.



# TP2 : Test d'Ansible

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

- **TODO** :
  - Vérifier le bon fonctionnement :

```
$ ansible -m ping all  
134.158.75.XXX | SUCCESS => {...}
```

```
$ ansible -m setup all  
"ansible_facts": {  
    "ansible_all_ipv4_addresses": ...}
```

- Note : l'option `-m` permet d'utiliser les modules Ansible, `all` désigne tous les serveurs référencés dans `hosts` et le module `setup` permet l'affichage des Facts (variables) qui seront utilisés pour configurer les services Torque.



# TP2 : Galaxy d'Ansible

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration

#### Play

Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ TODO :

- Récupérer depuis Galaxy un rôle Torque :

```
$ ansible-galaxy install micafer.torque
$ ls /etc/ansible/roles/micafer.torque
files handlers meta README.md
tasks templates vars
```

- Note : un rôle est une configuration modulaire et **Galaxy** est un hub pour le partage public des rôles.



# TP2 : Galaxy d'Ansible

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
**Play**  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ TODO :

- Remplacer les fichiers issus du dépôt Git dans le rôle micafer.torque (fichiers obsolètes) :

```
$tree Formation_OpenStack/PaaS/TP2/
```

```
Formation_OpenStack/PaaS/TP2/
```

```
|--- cluster.yml
```

```
|--- tasks
```

```
|   |--- main.yml
```

```
...
```

- Note : *cluster.yml* est le playbook utilisé juste après, *hostname.conf* (*/etc/hosts*), *exports.conf* (*/etc/exports*) et *mom.layout* (NUMA) sont de nouveaux templates.



# TP2 : Playbook

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play**
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

## ■ TODO :

- Exécuter le playbook pour installer Torque :

```
$ ansible-playbook cluster.yml
```

```
PLAY *****
```

```
TASK [setup] *****
```

```
ok: [134.158.75.XX]
```

```
ok: [134.158.75.XX]
```

- Note : le playbook exécute des rôles incluant des tâches qui sont différentes pour le master et le worker.



# TP2 : Torque

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
**Torque**

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ TODO :

- Vérifier le fonctionnement de Torque :

```
$ su - user1  
$ pbsnodes -a  
ansible-worker-0  
    state = free  
    np = 2  
    ...
```

- Note : les services *pbs\_server* et *pbs\_sched* sont installés sur le master et le service *pbs\_mom* est installé sur le worker. Les services *trqauthd* et *munge* servent à authentifier les démons de manière sécurisée.



# TP2 : Torque

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
**Torque**

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

## ■ TODO :

- Utiliser Torque pour soumettre un job :

```
$ echo "sleep 30" | qsub
```

```
0.ansible-master
```

```
$ qsub -I
```

```
$ hostname && ls && exit
```

```
$ qstat
```

Job ID	Name	User	Time Use	S	Queue
0.ansible-master	STDIN	user1	00:00:00	C	batch

- Note : la soumission de job est soit interactive (option `-I`) soit automatique, le NFS exporte le `/home` de master.



# TP3 : SlipStream



## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
Torque

- **Automatisation du déploiement.**
- **But du TP :**
  - Créer et configurer un compte sur SlipStream.
  - Lancer une MV avec SlipStream pour tester la configuration.
  - Lancer deux MVs avec SlipStream pour instancier un cluster Torque.



# TP3 : configuration de SlipStream

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

**Configuration**  
Exécution  
Torque

## ■ TODO:

- Se connecter à [l'interface Web](#) de SlipStream.
- Enregistrer les identifiants d'OpenStack :
  - nom / mot de passe / tenant / domain = default.
  - la clé publique SSH renseignée avec `nova keypair-add`.
- Configurer certaines options :
  - Champ *General* :  
Keep running after deployment = **Always**.  
Level of verbosity = 3



# TP3 : identifiants utilisateur

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

Description	Value
Default Cloud	LAL2
Keep running after deployment	Always
Level of verbosity	3 - Debugging
Execution timeout (in minutes)	30
SSH Public Key(s) (one per line)	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACRENUryBTg3oGwrFVniU3GMBJCHoDy JWVXWcuE/RHlAccXy4QtqRsGvx5MefDKk+kQ/SiDRuaah9pMhmvVUDTYpxUetCG /4bSveFKdCjOQX/UBUA1Jwe1rKcJvM7u4mrg6/JMjlliehokGpJ44ixnk22Ae4TYaH GDgoimOaE75ZSKdawaqgetVA1HYQ7vssF5YmomyIPa88UaQoGwxe+x1U83l0+ ZGLUhVlyLVlyeDKMH2pPq+h3yQanFvjKpxziq3HO/YMCP82q+/-8rai4KLa05FnFLFU Y5PsToP9oMjuFdDZm9Jdf6/ybJVeEO+1YNTNYeeu0mNbYMX9s3lqPvb cavet@apcm271.in2p3.fr
Cloud usage email	never



# TP3 : identifiants utilisateur

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

The screenshot shows the SlipStream user profile configuration page. The top navigation bar is red and contains the SlipStream logo, "Dashboard", "App Store", "Workspace", "Help", and the user name "fg\_formation\_user9". The main content area has a dark header with a user icon, the text "Your profile", and "Regular user". Below this is a "Save" button and "Cancel" and "Delete" icons. The page is divided into sections: "Summary", "General", "IPHC" (Add credentials), "LAL2" (Configured), and "LUPM" (Add credentials). The "LAL2" section is expanded to show a form with the following fields:

Description	Value
Number of VMs the user can start for this cloud	<input type="text" value="10"/>
Username	<input type="text" value="fg_formation_user9"/>
Password	<input type="password" value="*****"/>
Project name (Tenant name)	<input type="text" value="FG_Formation"/>
Domain	<input type="text" value="default"/>



# TP3 : instantiation de la MV



- **TODO:**
  - Lancer une MV CentOS7 :
    - m1.small (ou équivalent).
    - *Images/centos7/Deploy*
  - Attendre l'état *Ready* pour se connecter à la MV.

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
**Exécution**  
Torque



# TP3 : instantiation de la MV

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

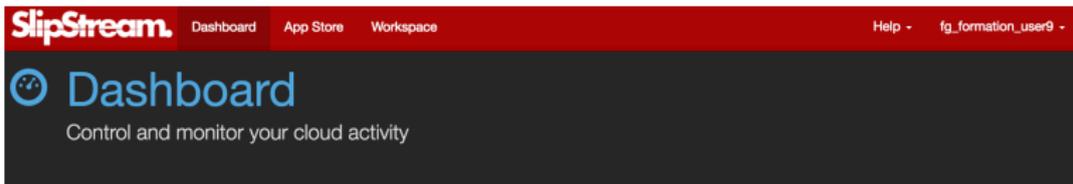
- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque



## Usage



Deployments Virtual Machines

Include inactive runs

ID	Application / Component	Service URL	State	Start Time	Clouds	User	Tags
c467c9a8	centos7 v92		Initializing	27 Apr 2016, 13:46:10 UTC	LAL2	fg_formation_user9	



# TP3 : instantiation de la MV

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

A screenshot of the SlipStream dashboard. The top navigation bar is red and contains the SlipStream logo, "Dashboard", "App Store", "Workspace", "Help", and "fg\_formation\_user9". The main content area has a dark background and displays "Deployment 13af4d08 is Ready" in large blue text. Below this, it says "Application component deployed by you (as 'fg\_formation\_user9') 10 minutes and 58 seconds ago". At the bottom of the main content area, there is a breadcrumb trail: "Home / Images / centos7 / 97 / 13af4d08" and a "Terminate" button with a circular arrow icon.

A light blue banner with a bookmark icon on the left. The text reads "The service is ready" in bold. Below this, there is a command: "ss:url.service - ssh://centos@134.158.75.127" with a right-pointing arrow.

A screenshot of the "Overview" page in the interface. The page title is "Overview" with an upward-pointing arrow. The main content area shows a single VM card. The card has a green icon of a VM and the text: "machine VM is Running CentOS 7 Ready!". To the right of the card is a tooltip box containing the following information: "ip: 134.158.75.127", "instance id: a4ceb962-57c8-43c3-8bf5-34058178a108", and "msg: CentOS 7 Ready!".



# TP3 : instantiation de la MV

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
**Exécution**  
Torque

## ■ TODO :

- Vérifier le fonctionnement de SlipStream :

```
$ nova list
```

```
+-----+-----+-----+-----+
| ID | Name | Status | Task |
+-----+-----+-----+-----+
| id | machine:id_ss | ACTIVE | - | Running
```

- Note : les MVs instanciées par SlipStream ont des noms spécifiques.



# TP3 : instantiation de la MV

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
**Exécution**  
Torque

## ■ TODO :

- Ajouter une adresse IP publique à *master* si besoin.
- Vérifier l'exécution des scripts par SlipStream dans les rapports de déploiement ou dans le log de la MV :

```
$ ssh centos@ip
```

```
$ sudo su
```

```
$ more \  
/var/log/slipstream/client/slipstream-node.log
```

- Note : l'exécution des scripts commence à l'instruction *Executing*.



# TP3 : instantiation du cluster

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
**Exécution**  
Torque



## ■ TODO:

- Lancer un (petit) cluster Torque :
  - 1 master + 1 slave : m1.medium (2 CPU, 6 GB mémoire, 30 GB de disque local).
  - *Torque/torque/Deploy*
- Se connecter au noeud master et vérifier l'exécution.
- Attendre l'état *Ready* pour tester Torque.



# TP3 : recettes Torque

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution**
- Torque

The screenshot shows the SlipStream web interface. At the top, there is a navigation bar with 'SlipStream.' logo, 'Dashboard', 'App Store', and 'Workspace' tabs. On the right, there are 'Help' and 'cavet' dropdown menus. The main content area displays 'Torque Project' with a version note: 'Version: 44 - Initial version of this project.' Below this, there is a breadcrumb trail: 'test / Torque / 44'. On the right side of the main area, there are buttons for 'New project', 'New component', and 'New application'. The interface also features expandable sections: 'Summary', 'Children', and 'Authorizations'. The 'Children' section is expanded, showing a table of components.

Name	Description	Owner	Version
torque		cavet	48
torque-master		cavet	57
torque-worker		cavet	58



# TP3 : déploiement de Torque

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
**Exécution**  
Torque

- **TODO :**
  - Vérifier le fonctionnement de SlipStream :

```
$ nova list
```

```
+-----+-----+-----+-----+
| ID | Name | Status | Task |
+-----+-----+-----+-----+
| id0 | orchestrator-LAL2:id0_ss | ACTIVE | - |
| id1 | master.1:id1_ss | ACTIVE | - |
| id2 | worker.1:id2_ss | ACTIVE | - |
```

- Note : l'orchestrator disparaît quand le cluster est dans l'état *Ready*.



# TP3 : déploiement de Torque

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

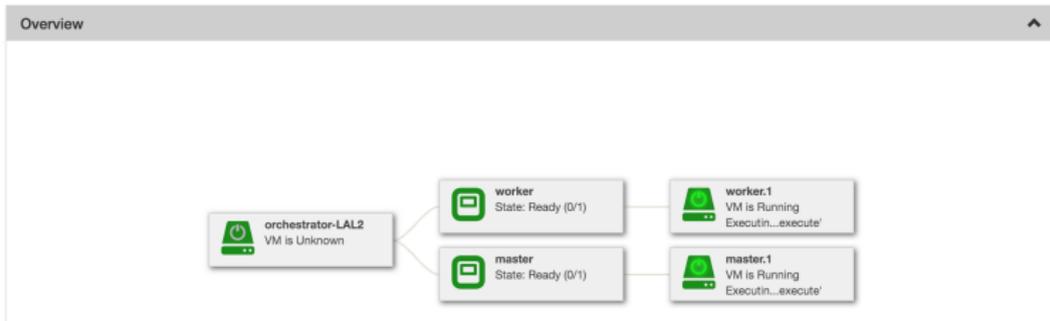
SlipStream Dashboard: Dashboard App Store Workspace Help cavet

## Deployment d0bde792 is Ready

Application deployed by you (as 'cavet') 21 minutes and 19 seconds ago

Home / test / Torque / torque / 48 / d0bde792 Terminate

**The service is ready**  
ss:url.service - ssh://user1@134.158.75.139



C. Cavet

Interface PaaS



# TP3 : déploiement de Torque

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
**Exécution**  
Torque

- **TODO** :
  - Explorer les paramètres de SlipStream :

```
$ ss-abort --cancel
```

```
$ ss-get worker.1:nodename  
worker
```

- Note : la commande *ss-abort -cancel* permet d'exécuter de nouvelles commandes si le déploiement de la MV a échoué.



# TP3 : déploiement de Torque

## Contexte

- Cas d'utilisation
- Docker
- Cluster Beowulf
- Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

- Configuration
- Service
- Registre privé
- Docker-compose

### TP2 : cluster avec Ansible

- Configuration
- Play
- Torque

### TP3 : cluster avec SlipStream

- Configuration
- Exécution
- Torque

**SlipStream** Dashboard App Store Workspace Help ▾ cavet ▾

↑ / test / Torque / torque / 48 / 83c5d24a 🔄 Terminate

master

master master.1

Name	Value	
master.1:LAL2.instance.type	m1.small	?
master.1:LAL2.security.groups	slipstream_managed	?
master.1:abort		?
master.1:cloudservice	LAL2	?
master.1:complete	false	?
master.1:disk.attach.size		?
master.1:disk.attached.device		?
master.1:disk.detach.device		?
master.1:extra.disk.volatile		?
master.1:hostname	134.158.75.245	?
master.1:id	1	?
master.1:image.id	e4325c96-547b-492f-8810-db4451d451a8	?
master.1:image.platform	centos	?
master.1:instanceid	19ddf4b4-4348-4886-a45b-87549f3fcf9a	?
master.1:is.orchestrator	false	?
master.1:master_hostname	onevm-245.lal.in2p3.fr	



# TP3 : Torque

## Contexte

Cas d'utilisation  
Docker  
Cluster Beowulf  
Cluster Hadoop

## Outils pour le cloud

### TP1 : Docker

Configuration  
Service  
Registre privé  
Docker-compose

### TP2 : cluster avec Ansible

Configuration  
Play  
Torque

### TP3 : cluster avec SlipStream

Configuration  
Exécution  
**Torque**

## ■ TODO :

- Vérifier le fonctionnement de Torque :

```
$ pbsnodes -a  
ansible-worker-0  
state = free  
np = 2  
...
```

- Note : les services *pbs\_server* et *pbs\_sched* sont installés sur le master et le service *pbs\_mom* est installé sur le worker. *trqauthd* et *munge* servent à authentifier les démons de manière sécurisée.