

Part 2.

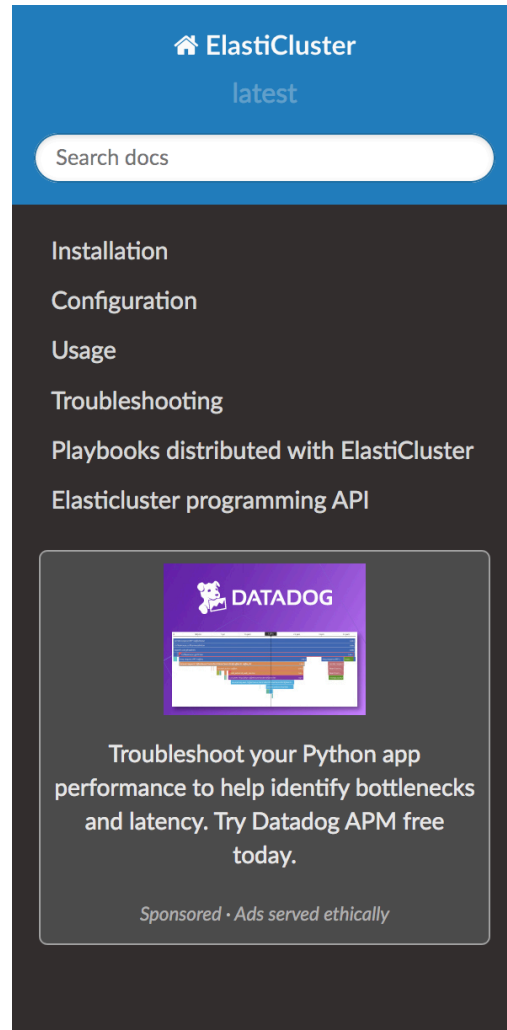
Create a cluster on the cloud

Elasticcluster tutorial

Learn how to create your own computer cluster

Norwegian Research and Education Cloud (NREC) Openstack cloud

Elasticcluster



[Docs](#) » Welcome to elasticcluster's documentation!

[Edit on GitHub](#)

Welcome to elasticcluster's documentation!

Introduction

ElasticCluster aims to provide a user-friendly command line tool to create, manage and setup computing clusters hosted on cloud infrastructures (like [Amazon's Elastic Compute Cloud EC2](#), [Google Compute Engine](#), or a private [OpenStack](#) cloud). Its main goal is to get a private cluster up and running with just a few commands; this [video](#) demoes ElasticCluster setting up a computational batch-queueing cluster.

Complete documentation for ElasticCluster is available on the [Read The Docs](#) website. General discussion over ElasticCluster's usage, features, and bugs takes place on the elasticcluster@googlegroups.com mailing-list (only subscribers can post).

The **ElasticCluster** project is an effort of the [Services and Support for Science IT \(S3IT\)](#) unit at the [University of Zurich](#), licensed under the [GNU General Public License version 3](#).

Features

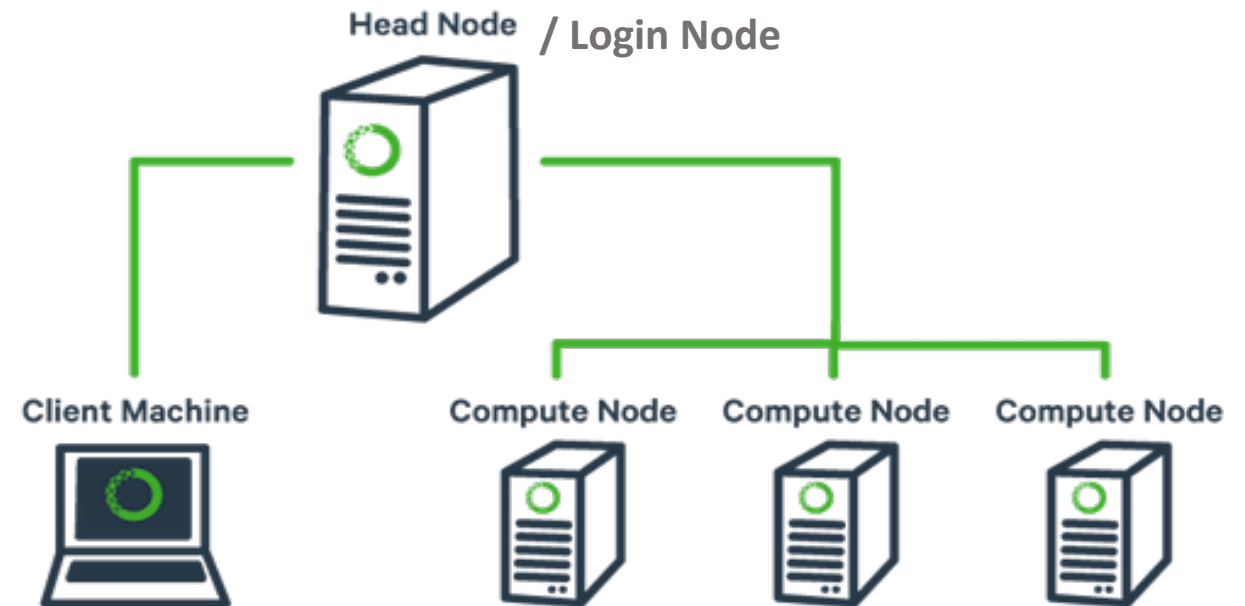
ElasticCluster is in active development, and offers the following features at the moment:

What is Elasticcluster and when to use it

- *[“ElastiCluster](#) aims to provide a user-friendly command line tool to create, manage and setup computing clusters hosted on cloud infrastructures (like [Amazon’s Elastic Compute Cloud EC2](#), [Google Compute Engine](#), or a private [OpenStack](#) cloud).”*
- Uses a combination of python and [ansible](#) to achieve this

Simplified view of a computing cluster

- A cluster is comprised of
 - A master machine and/or login-node
 - A set of compute-nodes/worker-nodes
 - A cluster management and job scheduling system orchestrating the jobs that come in
 - SLURM



Steps needed to create your own cluster

1. Create the virtual machines the cluster needs

- One master machine
- Several compute-nodes

2. Configure the machines

- Install needed software
- Adding any additional storage
- Set up any needed shared storage

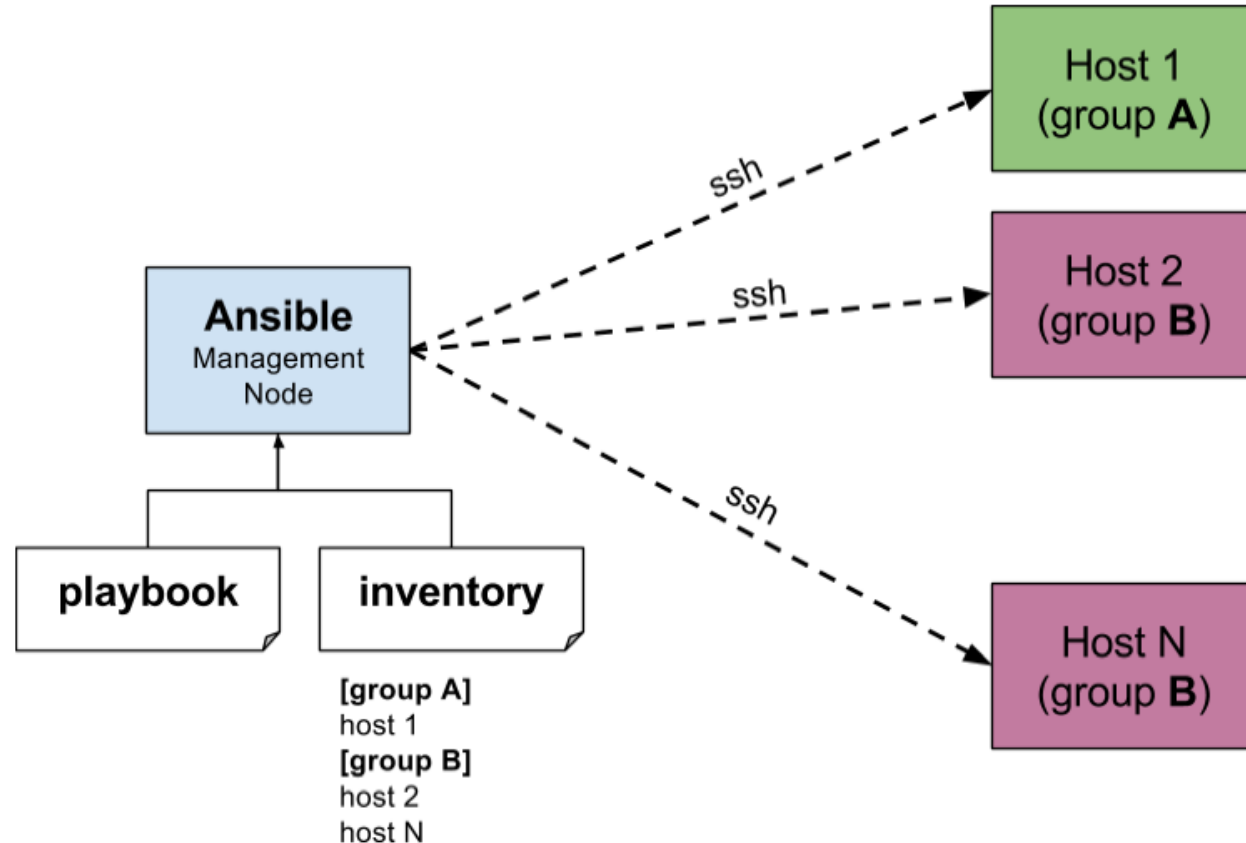
3. Log into master machine and submit a job

Your mini-cluster

- One login/master node
- Two compute/work nodes
- SLURM queueing system

Ansible

- Elasticcluster configures machines using Ansible
- Ansible is an automation tool
 - Instructions are written in yaml-format
 - Skips steps already done, has various fault handling features, and options to run some tasks only by using tags etc, etc.
- You could just write a set of python scripts or a bash scripts that do the same, but that quickly becomes more time-consuming



Ansible terminology

- Ansible inventory file: A list of hostnames and/or ip-addresses in groups
- Ansible playbook: Set of instructions to perform: *"Playbooks are Ansible's configuration, deployment, and orchestration language."*
- Ansible roles: framework for breaking down a playbook into a file-structure
 - Each role contains a set of variables, templates, files and tasks
 - Example: one role for configuring a compute node, and another for the frontend/login node

site.yml - playbook

```
---  
- hosts: webserver  
  roles:  
    - common  
    - webserver
```

Example inventory file

```
mail.example.com  
  
[webserver]  
foo.example.com  
bar.example.com  
  
[dbserver]  
one.example.com  
two.example.com  
three.example.com
```

Example role folder structure

```
site.yml  
webserver.yml  
fooserver.yml  
roles/  
  common/  
    tasks/  
    handlers/  
    files/  
    templates/  
    vars/  
    defaults/  
    meta/  
  webserver/  
    tasks/  
    defaults/  
    meta/
```

Demo of running a small ansible script

- Task: Install nano on all the admin nodes
- 1. Get a list of machines (here called hosts) to run on
 - Can be grouped by servers, like e.g. a set of login machines, compute machines, all machines etc etc
 - Can be created in a variety of ways
 - Manually
 - If you set up your machines with elasticcluster, you will find the list of hosts – the so-called inventory file in `~/.elasticcluster/storage/inventory_<your-cluster-name>`
 - Command-like client for the cloud architecture you are using – for us: openstack client
 - openstack server list
 - Needs installation, see e.g. https://docs.openstack.org/mitaka/user-guide/common/cli_install_openstack_command_line_clients.html

```
[admin]
student00 ansible_host=158.39.48.136
student01 ansible_host=158.39.75.98
student02 ansible_host=158.37.63.228
student03 ansible_host=158.39.48.48
student04 ansible_host=158.37.63.242
student05 ansible_host=158.39.48.10
student06 ansible_host=158.39.48.68
student07 ansible_host=158.39.48.18
student08 ansible_host=158.39.48.75
student09 ansible_host=158.39.48.29
student10 ansible_host=158.39.48.54
student11 ansible_host=158.39.48.70
```


Ansible demo contd.

2. Create a playbook

- This is the recipe for ansible to follow – it contains everything that ansible needs to perform the tasks you define
- Our demo will install the nano text editor using yum
 - Example: ansible yum module doc:
https://docs.ansible.com/ansible/latest/modules/yum_module.html
- Run the playbook:

```
ansible-playbook -i hosts_inf9380_simple -l student00 --private-key=~/.ssh/inf9380-ssh demo_role.yml
```

Ansible demo contd.

- Perform the same things, but now using a role

```
- hosts: admin
  become: true

  roles:
    - demo
```

```
site.yml
webservers.yml
fooservers.yml
roles/
  common/
    tasks/
    handlers/
    files/
    templates/
    vars/
    defaults/
    meta/
  webservers/
    tasks/
    defaults/
    meta/
```

Ansible demo contd.

- Run an ad-hoc command

```
ansible -i hosts_inf9380_simple student00 -m yum -a "name=emacs  
state=present" --become
```

Elasticcluster playbooks available

- Setup variables
 - General setup variables
 - Feature variables
 - Azure files
- Compute clusters
 - SLURM
 - GridEngine
 - Hadoop + Spark
 - HTCondor
 - Kubernetes
 - PBSPro
 - TORQUE
- Filesystems and storage
 - CephFS
 - GlusterFS
 - OrangeFS/PVFS2

- Add-on software
 - Anaconda
 - Ansible
 - CUDA
 - Docker CE
 - EasyBuild
 - Ganglia
 - IPython cluster
 - HPC common
 - Julia language
 - JupyterHub
 - OpenCPU
 - R language
 - R Studio Server
 - SAMBA

Elasticluster cluster configuration file (INI format)

- [cloud] section
 - What cloud platform and how to authenticate
- [login] section
 - Login details to the cluster
- [setup] section
 - What playbooks to install
 - Definition of host groups
- [cluster] sections
 - Cluster name, here student00
 - Definition of the host machine types

```
[cloud/iaas]
provider=openstack
auth_url=https://api.uh-iaas.no:5000/v3
username=some-username@some-email
password=some-password-token
project_name=uo-itf-inf9380
user_domain_name=dataporten
project_domain_name=dataporten
region_name=osl
identity_api_version=3

[login/centos]
image_user=centos
image_user_sudo=root
image_sudo=True
user_key_name=inf9380-ssh
user_key_private=~/.ssh/inf9380-ssh
user_key_public=~/.ssh/inf9380-ssh.pub

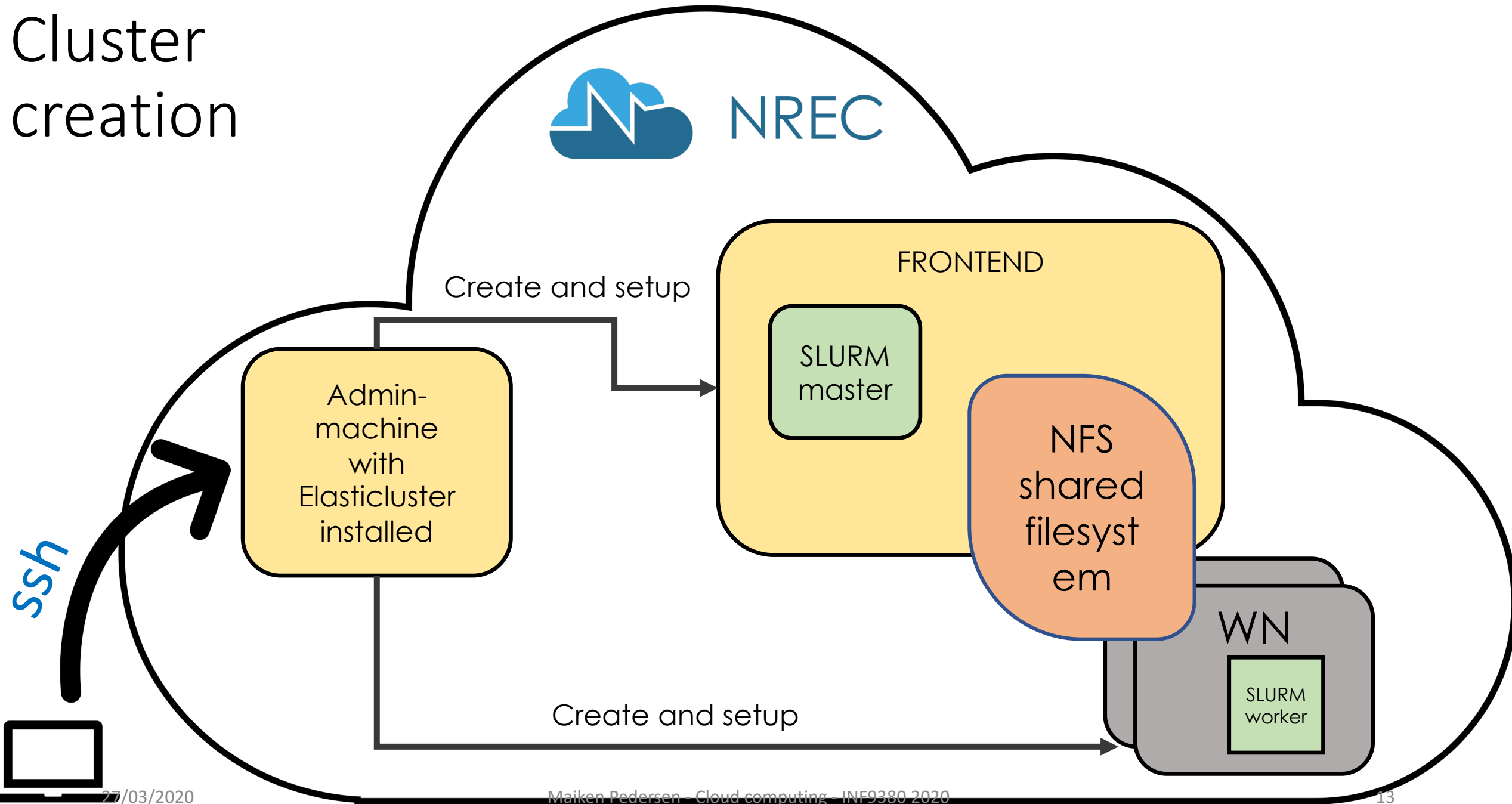
[setup/slurm]
slow_but_safer=True
provider=ansible
global_var_multiuser_cluster=no
login_groups=slurm_master
compute_groups=slurm_worker,julia

[cluster/student00]
cloud=iaas
login=centos
setup=slurm
security_group=inf9380
login_nodes=1
compute_nodes=2
ssh_to=login
network_ids=62421b56-346d-4794-99b0-fc27fe4e700f
image_id=0f6d4a45-043b-4231-872d-4c8f1aee34fc

[cluster/student00/login]
flavor=m1.small

[cluster/student00/compute]
flavor=m1.small
```

Cluster creation



Let's install elasticcluster and create the clusters