[SHARED WITH UMICH]  
Elasticluster and GCP - Getting Started

This document provides getting started instructions for using [Elasticluster](https://elasticluster.readthedocs.org) to create clusters of Google Compute Engine instances running job management software, such as Grid Engine, SLURM, or Hadoop.

The Elasticluster online documentation provides installation and getting started instructions which are currently incomplete in general and incomplete with regards to Google Compute Engine. We will be submitting patches and suggestions.

# What you will do

By following the instructions in this document, you will

1. Install Elasticluster software
2. Configure Elasticluster to access your Google Cloud Project
3. Configure a cluster of Compute Engine virtual machines running Grid Engine
4. Start a cluster of Compute Engine virtual machine running Grid Engine
5. List nodes of your cluster
6. SSH to the instances of your Cluster
7. Destroy the cluster

# Install Elasticluster

It is highly recommended that you install elasticluster in a [python virtualenv](http://docs.python-guide.org/en/latest/dev/virtualenvs/). This will allow you to contain your Elasticluster install and dependent libraries in one place.

The instructions here explicitly use a Python virtualenv and have only been tested in this environment.

1. If you have not installed virtualenv, then do so with:  
     
   [sudo] pip install virtualenv
2. Change directory to where you want to install Elasticluster. You don't need to explicitly create an elasticluster directory (that will happen next).
3. Create the elasticluster virtualenv:  
     
   virtualenv elasticluster  
     
   This creates a directory named elasticluster and populates it with the necessary Python library files and shell scripts to contain the Elasticluster installation. No Elasticluster software has yet been installed.
4. Activate the elasticluster virtualenv for the current shell session:  
     
   source elasticluster/bin/activate  
     
   This script will set environment variables necessary for the virtualenv:  
     
   VIRTUAL\_ENV: path to the elasticluster virtualenv directory  
   PATH: adds ${VIRTUAL\_ENV}/bin to the head of the PATH  
   PYTHONHOME: unset if currently set  
     
   The script also saves away changed environment variables and installs a "deactivate" function into the bash environment.
5. Install setuptools into the virtualenv  
     
   pip install setuptools==9.1  
   This will uninstall the current version of setuptools and install an older version. Without this step, the Elasticluster installation will fail with:  
     
    Traceback (most recent call last):  
    File "<string>", line 20, in <module>  
    File "/tmp/pip-build-uwB7Cn/elasticluster/setup.py", line 31, in <module>  
    del sdist.finders[:]  
    AttributeError: 'module' object has no attribute 'finders'  
      
    ----------------------------------------  
    Command "python setup.py egg\_info" failed with error code 1 in /tmp/pip-build-uwB7Cn/elasticluster
6. Install ansible into the virtualenv  
     
   pip install ansible==1.7.2  
     
   This will uninstall the current version of ansible and install an older version. Without this step, the installation of any software during cluster start will fail with:  
     
   ERROR:gc3.elasticluster:the setup provider was not able to setup the cluster,  
   but the cluster is running by now. Setup provider error message: `\_\_init\_\_()  
   got an unexpected keyword argument 'sudo'`  
     
   See <https://github.com/gc3-uzh-ch/elasticluster/issues/156>
7. Install elasticluster (select one):
   1. Using pip   
        
      pip install elasticluster
   2. From github (mbookman fork with Google-specific updates)  
        
      cd elasticluster  
      git clone https://github.com/mbookman/elasticluster.git src  
      cd src  
      python setup.py install  
        
      pip uninstall --yes google-api-python-client  
      pip install google-api-python-client
   3. From github (mainline)  
        
      cd elasticluster  
      git clone git://github.com/gc3-uzh-ch/elasticluster.git src  
      cd src  
      python setup.py install  
        
      pip uninstall --yes google-api-python-client  
      pip install google-api-python-client

**Note**: if you change versions of Elasticluster (from pip install to github install, for example), it is common to get inexplicable "AttributeErrors" when trying to deploy. This is due to Elasticluster saving Python objects to ~/.elasticluster/store/. Removing the contents of this directory may resolve your issues.

# Create your cluster definition file

Elasticluster cluster definitions are driven from a configuration file. By default this file is:

~/.elasticluster/config

Details of the config file can be found at:

<https://elasticluster.readthedocs.org/en/latest/configure.html>

Elasticluster provides a command to automatically create the config file for you, however using this command will create a template configuration file which you cannot immediately use as it has a list of clusters that are not correctly configured. You can either:

1. Install the default template using list-templates and then fix it up, or
2. Install a minimal template

In either case, you will need to configure the ~/.elasticluster/config file for accessing your Google Cloud project.

## Install the default template

If you install the default template using the command:

elasticluster list-templates

It will copy a default file to ~/.elasticluster/config and will emit a number of WARNINGS and ERRORS to the console. To use this configuration file, you must then comment out or remove all of the "cluster" examples. Look for the section:

# Cluster Section

and then comment out or remove everything up to the:

# Cluster node section

You can then copy each element (except setup/ansible-gridengine) of the following minimal template into the config file.

## Install a minimal template

Copy the file into ~/.elasticluster/config and update the fields marked with \*\*\*. Instructions for getting your client\_id and client\_secret can be found below. The instructions provided on the Elasticluster installation site are currently out of date.

# Gridengine software to be configured by Ansible

[setup/ansible-gridengine]

provider=ansible

frontend\_groups=gridengine\_master

compute\_groups=gridengine\_clients

# Create a cloud provider (call it "google-cloud")

[cloud/google-cloud]

provider=google

gce\_project\_id=\*\*\*\*REPLACE WITH YOUR PROJECT ID\*\*\*\*

gce\_client\_id=\*\*\*\*REPLACE WITH YOUR CLIENT ID\*\*\*\*

gce\_client\_secret=\*\*\*\*\*REPLACE WITH YOUR SECRET KEY\*\*\*\*\*

# Create a login (call it "google-login")

[login/google-login]

image\_user=\*\*\*\*\*REPLACE WITH YOUR GOOGLE USERID (just the userid, not email)\*\*\*\*\*

image\_user\_sudo=root

image\_sudo=True

user\_key\_name=elasticluster

user\_key\_private=~/.ssh/google\_compute\_engine

user\_key\_public=~/.ssh/google\_compute\_engine.pub

# Bring all of the elements together to define a cluster called "gridengine"

[cluster/gridengine]

cloud=google-cloud

login=google-login

setup\_provider=ansible-gridengine

security\_group=default

image\_id=\*\*\*\*\*REPLACE WITH OUTPUT FROM: gcloud compute images list | grep ^debian | cut -f 1 -d " " \*\*\*\*\*

flavor=n1-standard-1

frontend\_nodes=1

compute\_nodes=2

image\_userdata=

ssh\_to=frontend

### Obtaining your client\_id and client\_secret

To generate a client\_id and client\_secret to access the Google Compute Engine visit the following page:

<https://console.developers.google.com/project/_/apiui/credential>

1. Select the project to be used for your cluster
2. If a "Client ID for native application" is listed on this page, skip to step 8
3. Under the OAuth section, click "Create new Client ID"
4. Select "Installed Application"
5. If prompted, click "Configure consent screen" and follow the instructions to set a "product name" to identify your Cloud project in the consent screen
6. In the Create Client ID dialog, be sure the following are selected:  
    **Application type**: Installed application  
    **Installed application type**: Other
7. Click the "Create Client ID" button
8. You'll see your Client ID and Client secret listed under "Client ID for native application"

# Deploy your cluster

elasticluster start -v gridengine

# List your cluster instances

elasticluster list-nodes gridengine

# SSH to your instances

Elasticluster provides a convenience routine to connect to your frontend instance:

elasticluster ssh -v gridengine

However, you can connect to other instances using gcloud:

gcloud compute ssh <instance> --zone <zone>

# Destroy your cluster

elasticluster stop -v --yes gridengine

# Exit the virtualenv

The "activate" command creates a function in the bash environment called "deactivate". To exit the virtualenv, just execute the command:

deactivate