ElasticHPC Installation and User Guide v.2.0

Table of Contents:

1. INTRODUCTION

This document describes how to get started using ElasticHPC library, ElasticHPC v.2 supports three cloud service provider which are Amazon Web Services (AWS), Google Cloud (CGE) and Microsoft Windows Azure (Azure). We developed this library to help the user orchestrate their clusters over different cloud service provider, this features enforce the elasticity of cloud computing where every service provider offers different scenarios for billing and usage, for example Google Compute engine provides the user “Sustained Used” and Inferred Instances which keeps the user utilize the computation resources over one month on the other hand AWS offers the spot instance where the user can start specific type of computation power at certain price.

In the following sections, this document shows the installation and configuration for each cloud provider client side then presents how to use these clients to start and manage your cluster.

2. Installation of ElasticHPC

2.a Basic Requirements

This library was developed over Ubuntu 14.04 LTS,so to guarantee the optimum and minimum versioning issues, try to use it over this OS version. According to Windows Azure, user have to create an affinity group to keep all computation and storage resources in the same zone to create new affinity group click here,we recommend the affinity group name to be the same name of Storage account which will be add in configuration file. Before using ElasticHPC over GCE, User have to create new project from Google Compute Engine portal for information about creating new GCE project.

2.b AWS Agent

ElasticHPC Client over AWS requires the following packages:

- Python > 2.5
- Sun Java > 1.5
- PyCrypto >= 2.3
- SimpleJson
- ElementTree
- PrettyTables
- Boto

Then Download ElasticHPC AWS version from AWS Client page or pull the recent development version from bitbucket repo.

2.c Azure Agent

This package is a Linux based OS. As shown in the following list, the packages you should make sure that you have installed on your system to be able to use our package:

```
$sudo apt-get update
$sudo apt-get install Python build-essential openssh-server
  python-dev g++ gpp kcc libssl-dev libxml2-dev libtool
  python-m2crypto python-paramiko make cmake rcconf dialog mercurial
  python-pip libffi-dev swig
```

```
$sudo pip install boto south simplejson m2crypto paramiko
```

After installation Download ElasticHPC Azure version from Azure Client or pull the recent development version from bitbucket repo.

2.d GCE Agent

Google Compute Engine uses API Client library to interface with their services first you have to install all packages in section 2.b then install the following library:

```
$sudo pip install --upgrade google-api-python-client
```

Now we are ready to install GCE Agent/client to start using ElasticHPC over Google Compute Engine, to download ElasticHPC GCE version GCE Client or pull the recent development version of ElasticHPC GCE Client form bitbucket repo.

2.e ElasticHPC Configuration File

This is the most important section in to configure ElasticHPC before using the library. This section shows the configuration file 'cluster.conf' file which is parsed and processed during using ElasticHPC, so be careful when configuring this file, it is a one time configuration on the other hand there are two types to use ElasticHPC command mode and Config. file mode also you can use Agent Web interface from here

Configuration file is divided into 9 sections, where main cluster configuration and one section for each cloud service provider as shown below, fill only the labeled in red :

1. Google Compute Engine:

GCE has different authentication approach, you have to enable/activate the associated API for every resource you use, so to use ElasticHPC for GCE, and use Auth 2.0 to authorize requests, you have to follow these steps:

```
[GCE]
# GOOGLE COMPUTE ENGINE CONFIGURATION
PROJECT_ID = <PROJECT ID>      # Get Your Project Id from here
ZONE = us-central1-a           # More infor about Google Zones
CLIENT_SECRET = config/client_secret.json
COMPUTE_SCOPE = https://www.googleapis.com/auth/compute
OAUTH_STORAGE = config/oauth2.dat
IMAGE_PROJECT = <PROJECT ID>
SERVICE_EMAIL = default
NETWORK = default
SCOPES= https://www.googleapis.com/auth/devstorage.full_control
API_VERSION = v1
CLUSTER_CLIENT_KEY = keys/key
INSTANCE_TYPE= <INSTANCE TYPE>  # Machine types
MASTER_TYPE= <MAIN NODE TYPE>
ROOT_DISK=disks
```

2- Microsoft Windows Azure:

A. To Configure Windows Azure Certificates user has to do the following steps, user must create three certificates to manage the cluster using Azure API platform.

- public key .pem file 'mycert.pem' in client side
- server key .cer file 'mycert.cer' uploaded into Windows Azure Portal with active subscription.
- Service key .pfx file to authenticate requests.

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout \
    mycert.pem -out mycert.pem
$ openssl x509 -inform pem -in mycert.pem -outform der -out \
   mycert.cer
$ openssl pkcs12 -export -out mycert.pfx -in mycert.pem -name \
  "My Certificate"
```

B. Login to your Windows Azure portal from SETTINGS select MANAGEMENT CERTIFICATES then click on UPLOAD to upload 'mycert.cer' certificate to the associated valid subscription.

NOTE:

mycert.pfx file has a password this password will be added to windows azure config file later.

c. Move all certificate to the keys directories:

```
$ mv mycert.*  $EHPC_AZURE/keys/
```

```
[AZURE]
# MICROSOFT WINDOWS AZURE CONFIGURATIONS
SUBSCRIPTION_ID = <SUBSCRIPTION ID> # Subscription id
THUMBPRINT = <THUMBPRINT>  # Thumbprint or fingerprint
STORAGE_ACCOUNT=elastichpc # Storage and affinity group name
STORAGE_KEY     =          # optional
CERTIFICATE_PATH = /keys/mycert.pem #
PKFILE = /keys/mycert.cer
CERT_DATA_PATH = /keys/mycert.pfx
CERT_PASSWORD = <YOUR PASSWORD>
REGION = WUS                    # West US Region
INSTANCE_TYPE= Small            # Small Machine
MASTER_TYPE= Small              # Small Machine
CONTAINER=newcontainer          # Container Name
```

3. Amazon Web Services:

This section discusses how to get the AWS credentials required by HPC-cloud:

A. To get security credentials click on Account from Console bar,from the Account Page, choose 'Security Credentials'.

B. To get Private Key and X.509 certificate (required to start/terminate machines) are file found, From X509 Certificate tab, click on 'Create new Certificate', Download 'Private Key' and 'X. 509 Certificate'.

C. Access Key and Secret Key are required for the use of S3 and EMR.Go to 'Access Keys' tab, Copy your 'Access Key ID' or create a new one. Click the 'show' under 'Secret Access Key' to get the Access Key

```
[AWS]
# AMAZON WEB SERVICES CONFIGURATIONS
pkey= pk.pem
cert= cert.pem
accessKey= <ACCESS KEY>
secretKey= <SECRET KEY>
keyPair= <YOUR KEY PAIR>
securityGroup= <SECURITY GROUP>
INSTANCE_TYPE = <INSTANCE TYPE>
```

```
MASTER_TYPE = <MAIN NODE TYPE>
REGION = EU1
ZONE=eu-west-1b
```

4. **Main Cluster Configuration:**

```
[CLUSTER]
# CLUSTER CONFIGURATION
CLUSTER_PREFIX = <CLUSTER PREFIX>
CLUSTER_SIZE = <NUMBER OF COMPUTE NODE>
```

5. **Image Configuration:**

User can use a valid URL for Windows Azure to create new image from valid VHD file (optional).

```
[IMAGE]
# IMAGE CONFIGURATION
GCE_IMAGE_ID = elastichpc37          # Google Image Id
AZURE_IMAGE_ID = ehpc-generic7       # Azure Image Id
AZURE_OS_URL= <OS VHD FILE URL>      # VHD URL For
AWS_IMAGE_ID = ami-aa07bbdd          # AWS Image Id
```

6. **Firewall and EndPoints Configurations:**

SSH is configured by default

```
[FIREWALL]
# FIREWALL AND ENDPOINTS CONFIGURATION
FW_NAME=ehpcd,ehpcweb,http      # port name
FW_PORT=5000,8080,80            # port number
FW_PROTOCOL=tcp,tcp,tcp         # port protocol
```

7. **Data disk and block storage configurations:**

This section is used only for attaching new data disks to main node of the running cluster.

```
[DISK]
# BLOCK STORAGE CONFIGURATION
GCE_SNAPSHOT = <GCE SNAPSHOT>   # create Disk from snapshot
GCE_SNAPSHOT_PROJECT = <PROJECT ID> # snapshot project id
AZURE_DISK_URL = <DATA DISK VALID VHD FILE>
DISK_NAME = <DATA DISK NAME>
```

```
DISK_ZONE = <DATA DISK ZONE>
DISK_INSTANCE = <ATTACHED TO INSTANCE ID '0,1,..'>
DISK_NUMBER = <NUMBER OF DISKS>
DISK_SIZE = <DATA DISK SIZE>
```

8. **Gluster File System Configurations:**

This section is used to configure when user need to install a shared file system using GlusterFS. ElasticHPC supports GlusterFS over Windows Azure and Google Compute Engine only. To use a shared file system over AWS use NFS section.

```
[GLUSTERFS]
# GLUSTER FILE SYSTEM CONFIGURATION
MOUNT_POINT=/gluster/WGA/  # Mounting Point
VOLUME_NAME =gv0           # Volume Name
STRIPE     = 1             # Stripped data '1' to disable
REPLICATE  = 1             # Replicate data '1' to disable
FORMAT_DISK= False         #Format Data Disk as XFS Filesystem
```

9. **NFS configuration:**

Fill these configuration to install NFS shared file system.

```
[NFS]
# NFS CONFIGURATION
MOUNTING_POINT=/home            # Mounting Point 'GCE, Azure'
DEVICE=/home                    # Device Name    'GCE, Azure'
FSID=0                          # File System ID 'GCE, Azure'
# AWS ONLY attach new volume
AWS_VOLUME_SIZE=10              # New EBS volume Size
AWS_MOUNT_POINT=/gluster        # Mounting point AWS only
# in case of attach an exist volume
AWS_VOLUME_DEVICE=              # EBS Device
AWS_VOLUME_NAME=
```

2.f ElasticHPC Pre-configured Image

This section shows how to get a pre-configured image into your local image repository, user can use public image which in case of AWS that supports direct public share to their AMI market place, unlike Google that does not support any direct public sharing for pre-configured

image, on the other hand Windows Azure provides the user public image share using VM Depot Open Technologies where user upload the image to specific region or all regions. In this section, this guide handle all use cases:

1. For **AWS** AMI, user can use This AMI: **ami-aa07bbdd**
2. According to **Google Compute Engine**, user has to use the shared object file tar.gz which contains the image, this file will converted to image using the following steps:
   $ gcloud compute images create elastic-image --source-uri <TAR.GZ FILE URL>
3. To use a pre-configured image for **Azure**, user has to add the following link to image section and assign the value for AZURE_OS_URL=
   https://elastichpc.blob.core.windows.net/elastic9580/Disk1.vhd

3. Compute Node Functionality

For each section in this chapter, user has two options to use ElasticHPC, the first is to use command mode and the second is to use the configuration mode.

3.a Create New Cluster

- **configuration mode**

Here, as shown you can create a cluster over multiple cloud service providers using the same command line the only thing that user has to specify is the service provider

provider name: aws, gce, or azure

```
$./ehpcutils --provider=<provider name> --create
```

- **command mode**

In command mode, user has to specify every single detail for specific service provider but on the other hand user does not even need to configure cluster.conf, to create new cluster follow the command lines as shown below:

AWS

```
$./ehpc-client --create --conf=default -r=EU1
 --main-node-type=m1.xlarge --extra-nodes-type=m1.large n=5
 extraMachinesPBSCores=2
```

GCE

```
$./ehpc-client --command --oauth=config/GCE.dat --count=2 \
 --type=f1-micro --region=us-central1-a --image=elastichpc37 \
 --project=direct-mason-545 --cluster=hpc --kpair=keys/key --create
```

Azure

```
$./ehpc-client --command --thumb=<THUMBPRINT> --subid=<SUBSCRIPTION
ID> --pem=<PEM CERT.> --pfx=<PFX CERT> --cer=<SERVER CERT.>--count=2\
 --type=ExtraLarge --region=WUS --image=azure-image \
 --cluster=hpc --create
```

3.b List Compute Nodes

- **configuration mode**

List nodes in the cluster

```
$./ehpcutils --provider=<provider name> --compute --list
```

- **command mode**

AWS

```
$./ehpc-client --nodes-lst -d=<DOMAIN OF MAIN NODE>
```

GCE

```
$./ehpc-client --nodes --list --cluster=<cluster name>
```

Azure

```
$./ehpc-client --nodes --list --cluster=<cluster name>
```

3.c Terminate Cluster

- **configuration mode**

To terminate cluster user has to add the cluster name to the following command line

```
$./ehpcutils --provider=<provider> --terminate
--cluster=<clustername>
```

- **command mode**

AWS

```
$./ehpc-client --nodes-terminate -d=<DOMAIN_OF_MAIN_NODE>
 -id=<AWS_INSTANCE_ID>
```

GCE

```
$./ehpc-client --terminate --cluster=<MAIN NODE> --command
 --oauth=config/GCE.dat --project=<PROJECT ID>
 --region=<REGION>
```

Azure

```
$./ehpc-client --terminate --cluster=<MAIN NODE> --command
 --cert=<SERVER CERT> --pem=<PEM CERT.> --pfx=<PFX>
--pfx-pass=<PFX PASSWORD> --thumb=<THUMBPRINT>
--subid=<SUBSCRIPTION ID>
```

3.d Spot Instances 'AWS' only

Only this option in command mode

- **command mode**

Warning AWS automatically terminates the spot instances when the price exceed the maxium you set so make sure to backup the output of the processing tools.

3.d.1 Spot Request

The program will check every minute to see if the requested machines are running and then it will make the required configurations and print the url of the main node.

```
$./ehpc-client --spot-add -r=EU1 -n=2 -t=m1.medium --price=0.06
```

3.d.2 Spot Resume

Resume an interrupted spot request monitor.

```
$./ehpc-client --spot-resume
```

3.d.3 Cancel Spot

This part shows how to cancel an offer for spot instances. Cancel offer sir-7147fc07 in eu-west-1

```
$./ehpc-client --spot-cancel -r=EU1 -id=sir-7147fc07
```

4. Block Storage Functionality

Block storage is used to store images and hard disks, i.e Windows Azure uses Page Blobs to store Virtual Hard Disks (VHDs), Google Compute uses persistent disk to store hard disks, and finally AWS uses EBS volumes to store images and Hard disks which can be in SSD or magnetic hard disks

This section shows how to use the block storage and manage them over AWS, GCE, and Azure. User can use one of two supported mode like compute node functionality "command mode or configuration mode"

4.a Create Empty Data Disk

- **configuration mode**

provider "aws,gce,azure"

```
$./ehpcutils --provider=<PROVIDER> --data --size=100 --disk=disk1
```

- **command mode**

AWS

Create a new volume 10GB volume in zone eu-west-1b in region EU1

```
$./volumes --create-volume -r=EU1 -z=eu-west-1b -s=10
```

GCE

Create empty hard disk with a size of 200 GB

```
$./ehpc-client --disk --size=200 --diskname=tempdisk
--project=direct-mason-545 --command --oauth=config/GCE.dat
```

Azure

Create empty hard disk in VHD format with a size of 200 GB

```
$./ehpc-client --disk --size=200 --region=WUS --diskname=newdisk
 --cert=<SERVER CERT> --pem=<PEM CERT.> --pfx=<PFX> --pfx-pass=<PFX
 PASSWORD> --thumb=<THUMBPRINT> --subid=<SUBSCRIPTION ID>
```

4.b Create Hard Disk From Snapshot

Snapshot is used to store hard disks at specific state specially images, this option is not supported for Windows Azure, that is due to all of images and hard disks stored in a page blob format, so the following commands show how create hard disk from snapshot on GCE and AWS.

- **configuration mode**

cloud providers “aws, gce”

create a data disk from snapshot

```
$./ehpcutils --provider=<PROVIDER> --data --snapshot=markdups-data
```

- **command mode**

AWS

Create a new volume from snapshot snap-1626626 volume in zone eu-west-1b in region EU1

```
$./volumes --create-volume -r=EU1 -z=eu-west-1b -snap=snap-1626626
```

GCE

```
$./ehpc-client --disk --snapshot --diskname=markdups-copy
 --source=markdups-data --project=direct-mason-545 --command
 --oauth=config/GCE.dat
```

4.c Attach Hard Disk to Compute Node

- **configuration mode**

Create and attach hard disk to main node of running cluster, supported providers 'aws,gce,azure'

```
$./ehpcutils --provider=<PROVIDER> --data --size=100 --disk=disk1
--attach
```

- **command mode**

AWS

```
$./ehpc-client --volume-attach-mount --mountPoint=/vol6
--instance-id=i-121c2e5b --volume-id=vol-e27bb58a --format=Y
--device=/dev/sdm -d=ec2........com
```

GCE

User can create and attach hard disk by adding --attach option in create command line:

```
$./ehpc-client --disk --attach --instance=hpcnode0
 --diskname=markdups-copy --zone=us-central1-a --command
 --project=direct-mason-545 --oauth=config/GCE.dat
```

Azure

Azure's need to add the source VHD url.

```
$./ehpc-client --attachDataDisk --instance=elastic5000
--disk=new
--url=https://elastichpc.blob.core.windows.net/test/genomeKeyDisk.vhd
--cert=<SERVER CERT> --pem=<PEM CERT.> --pfx=<PFX> --pfx-pass=<PFX
PASSWORD> --thumb=<THUMBPRINT> --subid=<SUBSCRIPTION ID>
```

4.d Install and Configure GlusterFS

Gluster file system is a shared file system uses FUSE to share data and make a redundant data over storage systems. GlusterFS is supported only on GCE and Azure.

- **configuration mode**

Attach hard disk created from snapshot and install Gluster file system. supported providers are “gce and azure”.

```
$./ehpcutils --provider=<PROVIDER> --data --snapshot=markdups-data
--attach --gluster
```

- **command mode**

GCE

```
$./ehpc-client --disk --gluster --cluster=hpcnode --volume=gv0
--mountpoint=/gluster/gv0 --stripe=1 --replicate=1 --command
--project=direct-mason-545 --oauth=config/GCE.dat
```

Azure

For Windows Azure, user has to add --gluster option when attaching hard disk to main node

```
$./ehpc-client --attachDataDisk --instance=elastic5000 --disk=new
--url=https://elastichpc.blob.core.windows.net/test/genomeKeyDisk.vhd
--gluster --cert=<SERVER CERT> --pem=<PEM CERT.> --pfx=<PFX>
--pfx-pass=<PFX PASSWORD> --thumb=<THUMBPRINT> --subid=<SUBSCRIPTION
 ID>
```

4.e Install and Configure NFS

This section shows how to install and configure NFS over Cloud providers, NFS is supported only in command mode.

- **command mode**

AWS

AWS supports creating NFS when user create cluster, the following command line shows how to create an empty volume attached to main node as a shared NFS file system

A cluster can be created with NFS as shared file system. This can be achieved by using the create cluster with appropriate options

```
$./ehpc-client  --create --conf=cluster -r=EU1 -t=m3.medium
-n=5  -ami=ami-1c41f16b -sg=sg-fcda5c99
--nfs-mountPoint=/nfs --nfs-newVolume=200 -t=r3.2xlarge
```

GCE

```
$./ehpc-client --nfs --device=/home --mountpoint=/home --fsid=1
--cluster=micro  --command --project=direct-mason-545
--oauth=config/GCE.dat
```

Azure

```
$./ehpc-client --nfs --cluster=elastic5000  --cert=<SERVER CERT>
--pem=<PEM CERT.> --pfx=<PFX> --pfx-pass=<PFX PASSWORD>
 --thumb=<THUMBPRINT> --subid=<SUBSCRIPTION ID>
```

4.f List Hard Disks

- **configuration mode**

list all hard disks for specific service cloud provider

```
$./ehpcutils --provider=<PROVIDER> --data --list
```

- **command mode**

AWS

```
$./volumes --lst-avaliable-volumes --region=EU1 --zone=eu-west-1a
--private-Key=<PRIVATE KEY> --certificate=<CERTIFICATE>
```

GCE

```
$./ehpc-client --list-disks  --command --project=direct-mason-545
--oauth=config/GCE.dat
```

Azure

```
$./ehpc-client --list-disks  --thumb=<THUMBPRINT>
--subid=<SUBSCRIPTION ID>
```

4.g Remove Hard Disks

This section shows how to delete hard disk from cloud provider

- **configuration mode**

```
$./ehpcutils --provider=aws --data --remove --disk=vol-9b7f409c

$./ehpcutils --provider=gce --data --remove --disk=demodisk1
```

- **command mode**

AWS

```
$./volumes --delete-volume -r=EU1 -id=vol-dc6ca2b4
--private-Key=<PRIVATE KEY> --certificate=<CERTIFICATE>
```

GCE

```
$./ehpc-client --remove-disk --disk=$disk_name  --command
--project=direct-mason-545 --oauth=config/GCE.dat
```

Azure

```
$./ehpc-client --remove-disk --disk=$disk_name --thumb=<THUMBPRINT>
--subid=<SUBSCRIPTION ID>
```

5. Job Submission Functionality

ElasticHPC uses Job submission functionality to manage the running jobs over cluster on different cloud service providers, ElasticHPC user PBS Torque to run jobs over Amazon Web Service, on the other hand uses Sun Grid Engine over Google Compute Engine and Microsoft Windows Azure. Job Scheduler is installed directly where cluster is initiated, user will not care about configuration or installation issues, once the cluster is created, user can submit jobs

Note:

Each submitted job has an owner to distinguish the type of job.

AWS case:

If owner is system, the command will execute on the command line as root. If owner is ubuntu, the command will execute on the command line as ubuntu. If owner is otherwise, this will be a PBS Torque"

GCE and Azure case:

If owner is 'ehpcuser', the command will executed on the main node, if owner is 'sge', this will be a sun grid engine job.

5.a Submit PBS Torque Job over 'AWS'

- **configuration mode**

```
$./ehpcutils --provider=aws --submit --command='sleep 10;ls -la
/home/ubuntu' --domain=ec2-....eu-west-1.compute.amazonaws.com
--owner=ubuntu
```

- **command mode**

Run Emboss seqret on /home/user/file1 and get the result on /result/seqret/output

```
$./ehpc-client --job-submit --domain='ec2.....com' --id=3
--owner=john --commmand='seqret -sequence /home/user/file1 -outseq
```

```
/result/seqret/output -feature Yes' --input-files=/home/user/file1
```

5.b Submit Sun Grid Engine Jobs

Job submission in case of sun grid engine done in configuration mode and does not need any credentials just the cluster owner.

- **configuration mode**

```
$./ehpcutils --provider=gce --submit --command="sleep 10"
--owner="sge"
$./ehpcutils --provider=azure --submit --command="sleep 10"
--owner="sge"
```

5.c Monitor Running Jobs

If user submits a job over Sun Grid Engine the user can monitor the status of job using two options first by adding --monitor to submission command line as shown above or using the web interface of cluster by using <MAIN NODE IP>:8080/sge then select monitor running jobs.

If user submits a job to PBS Torque over AWS, to monitor the job, user can use this command line

```
$./ehpc-client --job-status -d=ec2...com -id=1
```

6.Cross Cloud Service Provider

This section shows how to create a cluster over multiple cloud service provider, one of use cases where the user can start the main node as "on demand" node over AWS then can orchestrate the cluster creation over other cloud providers or use the spot instance, if the cluster of spot instances is terminated by AWS, main node can rebuild this cluster over GCE or Azure. Compute node termination will not affect the data where all data is shared using one of shared systems and attached to the main node that is has enabled termination protection option.

Configure Cross section:

User has to configure this section in cluster.conf file to enable cross platform option, and add all credentials and certificates that coincided to required cloud service providers, also determine where the master node will be and number of instances for each cloud provider.

```
[CROSS]
# CROSS CLUSTERS OVER DIFFERENT CLOUD PROVIDERS
AWS= 1
AZURE = 0
GCE = 4
MASTER = aws
```

6.a Create Cluster

```
$./ehpcutils --cross --create
```

6.b Terminate Cluster

```
$./ehpcutils --cross --terminate
```