Elastic Cloud Compute Cluster (EC3)

Amanda Calatrava, Miguel Caballer
Universitat Politècnica de València (UPV)
Contact: amcaar@i3m.upv.es
What is EC3?

- EC3 was created with the idea of providing virtual elastic computer clusters on Cloud platforms.

Facilitate access to computing platforms for non-experienced users

Automatic management of elasticity, reducing costs (public cloud) and energy expenditure (private cloud).

Compatible with a wide range of cloud providers (public, federated and on-premises).

Maintain the traditional work environment, with clusters configured with a well-known middleware.

Automatic configuration of the application execution environment.

Support for hybrid clusters.
• **EC3** deploys and configures virtual elastic clusters. It relies on **IM** to deploy the machines and on **CLUES** to automatically manage the elasticity.

• Offers a set of predefined templates to configure the resources through **Ansible**: - Kubernetes, Mesos, SLURM, Torque, SGE, HTCondor, Nomad.
Deployment models

- An **homogeneous cluster** is composed by working nodes that have the same characteristics.
- In an **heterogeneous cluster** the working nodes can have different characteristics (hardware and software). For example, nodes with GPUs.
- **Cloud Bursting** (Hybrid clusters): consists on launching nodes in two or more different Cloud providers. This is done to manage user quotas or saturated resources.
Automatic Elasticity

Elasticity Management: ability to adapt the size of the cluster to the workload dynamically and automatically, ensuring transparency.

- The **elasticity module** is responsible for dynamically adding and removing nodes from the cluster by monitoring the LRMS. **Self-management.**

- Deployment policies (**scale out**):
  - On demand: a node is deployed for each job that comes to the queue.
  - Bursts: deploys a group of VMs for each job in the queue, assuming that if a job arrives at the LRMS, there is an increased chance that new jobs will arrive soon. (i.e. HTC applications).

- Undeployment policies (**scale in**):
  - On demand: ends idle nodes when there are no pending jobs in the LRMS queue.
  - Delayed power off: inactive nodes turn off after a certain configurable period of time. (i.e. public clouds)
The EGI AoD allows small laboratories and individual researchers the access to a wide range of computational resources and on-line services to manage and analyse large amount of data.

Inside this service we find the EC3 portal:
- The EC3 AoD portal enables to launch virtual elastic clusters on top of EGI FedCloud resources using the EC3 tool.
- It only requires the EGI Check-in account (and vo.access.egi.eu VO) to access to the service.
- The user is guided step by step in the deployment process.
- Documentation and tutorials are available, i.e. configuring a Galaxy cluster for data intensive research.

EC3 in the EGI Applications on Demand
EC3 Portal (EGI AoD)

- EC3aaS facilitates the usage of EC3 to non-experienced users:
  - It presents an user-friendly web interface that allows to easily deploy and configure a virtual elastic cluster on EGI Cloud Compute resources.
  - Limited actions: create, list and destroy.

- Documentation: https://ec3.readthedocs.io/en/latest/ec3aas.html
Wizard with 6 simple steps

1. Cluster configuration
   - Choose the LIMES (Local Resource Management System) of your cluster

2. Operation System
   - Choose the operation system

3. Instance details
   - Choose the instance type

4. Cluster's size & Name
   - Choose the number of nodes

5. Endpoint
   - Enter the endpoint information

6. Resume and launch
   - Submit the configuration and launch the cluster
EC3 Client

• More powerful client interface than the Web interface:
  – More control over the cluster (reconfigure, clone, migrate, stop, restart, update).
  – Support for hybrid clusters
  – Support for golden images

• The user needs to define an authorization file
• EC3 Client Source Code in GitHub: https://github.com/grycap/ec3
• EC3 Client container image available:
  – in Docker Hub:
    • https://hub.docker.com/r/grycap/ec3/
  – In GitHub Container Registry:
    • https://github.com/grycap/im-dashboard/pkgs/container/im-dashboard

   {launch,list,show,templates,ssh,reconfigure,destroy,clone,migrate,stop,restart,transfer,update}

Operation:

  launch  launch a new cluster
  list    list launched clusters
  show    print RADL
  templates list available templates
  ssh     connect to cluster via SSH
  reconfigure reconfigure the cluster
  destroy destroy a launched cluster
  clone   clone a launched cluster in another Cloud provider
  migrate migrate a launched cluster to another Cloud provider
  stop    stop a launched cluster
  restart restart a previously stopped cluster
  transfer transfer a previously launched cluster
  update  update the RADL of the WNs
First create a file auth.txt with a single line like this:

```
id = egi; type = EGI; host = CESGA; vo = vo.access.egi.eu; token = <egi_aai_token_value>
```

Replace `<egi_aai_token_value>` with a valid EGI Checking access token (you can also use `oidc-agent`).

Create an `ubuntu-cesga.radl` setting the correct URI of the base image to use i.e.:

```
appdb://CESGA/egi.docker.ubuntu.16.04?fedcloud.egi.eu
```

The next command deploys a Kubernetes cluster based on an Ubuntu image:

```
$ ec3 launch mycluster kubernetes ubuntu-cesga -a auth.txt -y
```

Creating infrastructure

Infrastructure successfully created with ID: 719bd78e-25b0-11ec-8796-de075615d95b

Front-end state: running, IP: 193.144.46.211
Usage examples

- **EKaaS**: The EKaaS (Elastic Kubernetes as a Service) is an on-demand service to deploy Elastic Kubernetes clusters on the EGI Cloud Compute.
  - Demo: [https://youtu.be/WAG4VaBMIyI](https://youtu.be/WAG4VaBMIyI)

- **OSCAR**: is Functions as a Service (FaaS) computing model for file-processing applications. It enables the creation of highly-parallel event-driven file-processing serverless applications that execute on customized Docker containers than run on an elastic Kubernetes cluster.
  - Demo: [https://youtu.be/ZtAlVc1uLwc](https://youtu.be/ZtAlVc1uLwc)

- Launch a Galaxy Portal configured to use a Torque elastic cluster as computing back-end with EC3 client:

- **SAPS** (Surface Energy Balance Automated Processing Service) in EOSC Synergy:
  - Demo: [https://youtu.be/mM6xJJRS3Cs](https://youtu.be/mM6xJJRS3Cs)
Video tutorials and demos in YouTube:
https://www.youtube.com/playlist?list=PLgPH186Qwh_1I0esmaTLjd35O-QqdWf9k

EC3 AoD portal:
https://servproject.i3m.upv.es/ec3-ltos

EC3 in EOSC Marketplace:

EC3 official documentation:
https://ec3.readthedocs.io/

EC3 source code:
https://github.com/grycap/ec3

EC3 training course in EOSC Synergy Learning platform:
https://moodle.learn.eosc-synergy.eu/course/view.php?id=14&section=0#tabs-tree-start
Thank you for your attention!!

Amanda Calatrava
Universitat Politècnica de València (UPV)
Contact: amcaar@i3m.upv.es