INFRASTRUCTURE MANAGER (IM)
Introduction

- IM is a service that **deploys virtual infrastructures** on top of Cloud resources.
- It uses **RADL** or **TOSCA** files to describe the infrastructure.
  - Infrastructure as code (IaC)
- The IM **automates** the deployment, configuration, software installation, monitoring and update of virtual infrastructures.
- It supports a wide variety of back-ends, thus making user applications **Cloud agnostic**.
IM Features

- It features **DevOps** capabilities.
  - Based on **Ansible**.
  - Provides recipes for common deployments.
  - Also supporting cloud-init scripts.

- IM works as a service that offers several interfaces:
  - XML-RPC and **REST** APIs.
  - Command-line application.
  - 2 Web-based GUI.

- It is distributed under the GNU **GPL v3.0** open source license and its source code is available on **GitHub**.
  - [https://github.com/grycap/im](https://github.com/grycap/im)
General View

- General platform to deploy on demand **customizable** virtual computing infrastructures.
  - With the **precise software configuration** required.
  - Allow to deploy any kind of complex infrastructure.
    - Hybrid ones.
  - Share Infrastructure descriptions.
  - **No need of pre-packaged VMIs.**
    - Enable re-using of VMIs.
  - The same complex infrastructure can be deployed both on-premise and in a public Cloud.
It supports a **wide range of cloud providers** and other computing back-ends:

- **Public**: Amazon Web Services, Google Cloud Platform, Microsoft Azure, T-Systems OTC, Exoscale, Linode and Orange.
- **On-premises**: OpenNebula, OpenStack, CloudStack, VMWare, libvirt.
- **Federated**: EGI FedCloud (OCCI and OpenStack), FogBow.
- **Containers**: Docker, Kubernetes
- The list above can be **easily extended** by plugins.
The user can provide a RADL or TOSCA documents as input to the IM, describing the infrastructure:

- **RADL**: Resource and Application Description Language.
  - High level Language to define virtual infrastructures and Specify VM requirements.

- **TOSCA (YAML spec 1.0)**: OASIS Standard
  - Open standard language to model application architectures to be deployed on a Cloud.
Let's access the IM Dashboard!!

https://www.grycap.upv.es/im
https://appsgrycap.i3m.upv.es:31443/im-dashboard/

See full demo video at:
- https://youtu.be/vmtzGOZxiUg
An RADL document has the following general structure:

The keywords `ansible`, `network`, `system` and `configure` assign some features or recipes to an identity `<id>`. The features are a list of constraints separated by `and`, and a constraint is formed by `<feature name> <operator> <value>`.

```plaintext
ansible <ansible_host_id> (<features>)
network <network_id> (<features>)
system <system_id> (<features>)
configure <configure_id> (<Ansible recipes>)
contextualize [max_time] ( system <system_id> configure <configure_id> [step <num>] ... )
deploy <system_id> <num> [<cloud_id>]
```
In this example
- A node type named “node” with 1 CPU and 512MB of RAM is defined.
- Connected to a public network
- In the configuration a user named “user1” is created.
- 1 node of type “node” is deployed
TOSCA

- Topology and Orchestration Specification for Cloud Applications
  - OASIS Standard
  - TOSCA Simple Profile in YAML Version 1.0
  - Standard to specify Cloud Topologies
  - Defines the interoperable description of services and applications hosted on the cloud
    - Including their components, relationships, dependencies, requirements, and capabilities
  - Enabling portability and automated management across cloud providers
tosca_definitions_version: tosca_simple_yaml_1_0

going imports:

- types:
  https://../custom_types.yaml
description: Deploy instance for Kepler
topology_template:

  inputs:

    memory_size:
      type: scalar-unit.size
      description: RAM memory
      default: 1 GB

node_templates:

  kepler:
    type: tosca.nodes.indigo.Kepler
    requirements:
      - host: kepler_server

kepler_server:
  type: tosca.nodes.indigo.Compute
capabilities:
  endpoint:
    properties:
      network_name: PUBLIC
      ports:
        vnc_port:
          protocol: tcp
          source: 5900

host:
  properties:
    num_cpus: 1
    mem_size:
      get_input: memory_size

os:
  properties:
    type: linux
distribution: ubuntu
version: 18.04

outputs:

  instance_ip:
    value:
      get_attribute:
        - kepler_server
        - public_address
        - 0

  instance_creds:
    value:
      get_attribute:
        - kepler_server
        - endpoint
        - credential
        - 0

Operation:
  list
  create <radl_file> [async_flag]
  destroy <inf_id>
  getinfo <inf_id> [radl_attribute]
  getradl <inf_id>
  getcontmsg <inf_id>
  getstate <inf_id>
  getvminfo <inf_id> <vm_id> [radl_attribute]
  getvmcontmsg <inf_id> <vm_id>
  addresource <inf_id> <radl_file> [ctxt flag]
  ...

Client-SIDE Tools: Web

- Two publicly-available web interface (also open-sourced).
  - [https://appsgrycap.i3m.upv.es:31443/im-dashboard/](https://appsgrycap.i3m.upv.es:31443/im-dashboard/)
    - For non advanced users
  - [https://appsgrycap.i3m.upv.es:31443/im-web/](https://appsgrycap.i3m.upv.es:31443/im-web/)
    - More functionality
  - Easily deploy infrastructures from a web browser
  - Also on GitHub:
    - [https://github.com/grycap/im-web](https://github.com/grycap/im-web)
    - [https://github.com/grycap/im-dashboard](https://github.com/grycap/im-dashboard)
Client-SIDE Tools: Web Portal

- Full featured Web interface
  - For advanced users
  - Enable to define and share your own topologies
    - TOSCA o RADL
Client-SIDE Tools: Web Dashboard

- Easy interface
  - For non advanced users
  - Easily deploy infrastructures from a web browser
    - Select it from a list of configurable list of templates.
APIs to be consumed by Clients

- **XML-RPC API**
  - API that follows the XML-RPC specification.

- **REST API**
  - IM Service can be accessed through a REST(ful) API
  - Follows the OpenAPI Specification

Where is the IM used?

- The IM is used in the VMOps Dashboard of EGI.
  - As the OpenStack communication layer to create VM topologies.
  - [https://dashboard.appdb.egi.eu/vmops](https://dashboard.appdb.egi.eu/vmops)
Where is the IM used?

- In the **INDIGO PaaS Orchestrator**:
  - [https://indigo-paas.cloud.ba.infn.it/](https://indigo-paas.cloud.ba.infn.it/)
  - Used at the PaaS Core to provide deployment of infrastructures to Cloud providers.

- In (Elastic Cloud Computing Cluster) **EC3**:
  - [https://servproject.i3m.upv.es/ec3-ltos/](https://servproject.i3m.upv.es/ec3-ltos/)
  - The IM is the Cloud orchestrator that deploys the virtual infrastructures.
Go back to the IM Dashboard!!

https://www.grycap.upv.es/im
https://appsgrycap.i3m.upv.es:31443/im-dashboard/

See full demo video at:
- https://youtu.be/vmtzGOZxiUg
More Information

Video demos in YouTube:
https://www.youtube.com/playlist?list=PLgPH186Qwh_37AMhEruhVKZSfoYpHkrUp

EOSC Marketplace Entry:

IM image in Docker Hub:
https://hub.docker.com/r/grycap/im/

Source Code in GitHub:
https://github.com/grycap/im

IM Info Web:
http://www.grycap.upv.es/im
Questions?