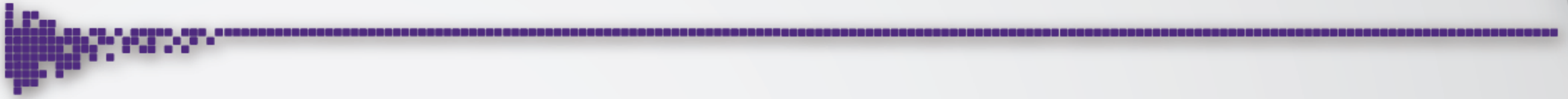
Indigo-dc TOSCA example

Alfonso Pérez – alpegon3@upv.es

Universitat Politècnica de Valènica

INDIGO-DataCloud is co-founded by the Horizon 2020Framework Programme

Use case



- We want to transform the example deployed the first day of the Hands-on
- We know what we want to deploy
  - A front-end node with apache + saxs installed
- We have the required extra files and packages
  - Everything needed is in the link:
    - https://github.com/ICS-MU/westlife-mustweek2017/tree/solution/solved

Use case

- First:
  - Look for already defined TOSCA types:
    - Check indigo tosca-types repository (custom_types.yaml):
      - https://github.com/indigo-dc/tosca-types/blob/master/custom_types.yaml
  - For this example we are using a fork of the repository (to be able to update the files):
    - https://github.com/alpegon/tosca-types/blob/master/custom_types.yaml

Creating the apache node

- Look for already defined TOSCA types:
  - Check indigo tosca-types repository (custom_types.yaml):
    - https://github.com/indigo-dc/tosca-types/blob/master/custom_types.yaml
- We find an Apache node:

```
180    tosca.nodes.WebServer.Apache:
181      derived_from: tosca.nodes.WebServer
182      interfaces:
183        Standard:
184          create:
185            implementation: https://raw.githubusercontent.com/indigo-dc/tosca-types/master/artifacts/apache/apache_install
186          start:
187            implementation: https://raw.githubusercontent.com/indigo-dc/tosca-types/master/artifacts/apache/apache_start.y
```

Creating the apache node

- Usage of the node found in our template:

```
node_templates:

  apache:
    type: tosca.nodes.WebServer.Apache
    requirements:
      - host: web_server
```

Creating the saxs node

- Look for the TOSCA type
- No type found
- We need to define a new type
  - The base node type to define a software component is:
    - tosca.nodes.SoftwareComponent
  - We are going to use ansible to configure the node

Creating the saxs node

- New type defined in the custom_types files
- Node implementation stored in the artifacts folder

```
189  # INDIGO non normative types
190    tosca.nodes.indigo.Saxs:
191      derived_from: tosca.nodes.SoftwareComponent
192      artifacts:
193        docker_agent_role:
194          file: alpegon.saxs
195          type: tosca.artifacts.AnsibleGalaxy.role
196      interfaces:
197        Standard:
198          configure:
199            implementation: https://raw.githubusercontent.com/alpegon/tosca-types/master/artifacts/saxs/sax-basic-install.yml
200      requirements:
201        - host:
202            node: tosca.nodes.indigo.Compute
203            relationship: tosca.relationships.HostedOn
```

Creating the saxs node

INDIGO - DataCloud

- Node implementation:
  - https://raw.githubusercontent.com/alpegon/tosca-types/master/artifacts/saxs/sax-basic-install.yml
  - Uses ansible galaxy capabilities

```
---
- hosts: localhost
  connection: local
  roles:
  - { role : alpegon.saxs }
```

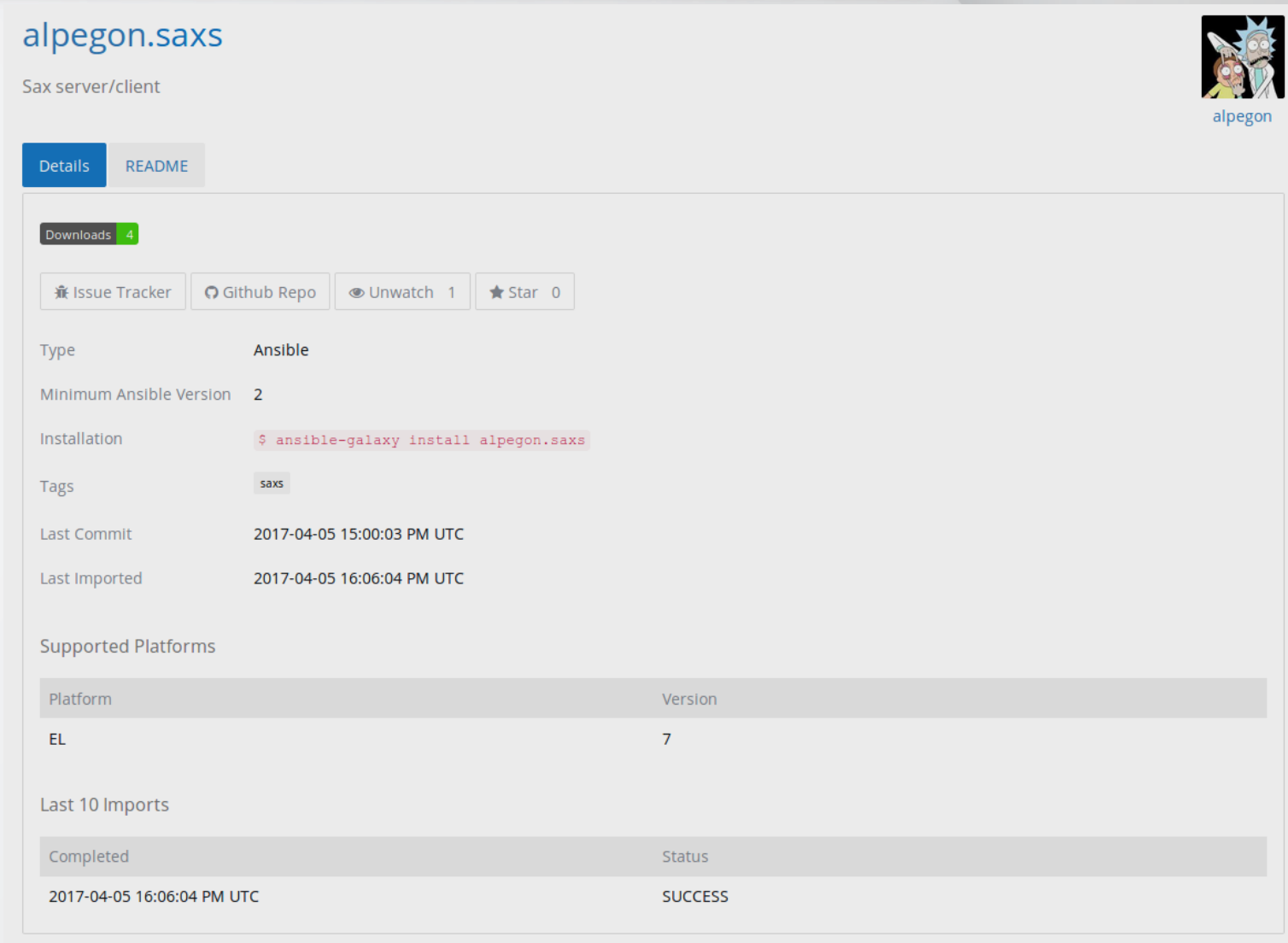
Creating the saxs node

- Using ansible for defining node configuration:
  - Create new ansible role with the configuration desired:
    - https://github.com/alpegon/ansible-role-saxs
  - Register the role in ansible galaxy:
    - https://galaxy.ansible.com/alpegon/saxs/

Creating the saxs node

INDIGO - DataCloud

- Ansible galaxy
  - Linked with github
  - Automatically updated



Creating the saxs node

- The role in ansible galaxy:
  - https://galaxy.ansible.com/alpegon/saxs/
- Uses ansible role structure
  - files: store scripts and packages
  - meta: store role related information
  - tasks: store tasks to be executed by ansible

Creating the TOSCA template

- Check the TOSCA template in:
  - https://github.com/alpegon/ansible-role-saxs/blob/master/README.md
- We want an apache node with saxs installed
- We want a public ip and the ports 80 and 443 open
- We want a host with 1GB of ram and 1 cpu
- We want the OS to be CentOS7
- As ouput we want to see the server IP

Deploying the TOSCA template

- You can use the IM:
  - http://www.grycap.upv.es/im/index.php
  - Can deploy in Opennebula, openstack, aws, windows azure, etc
- If you have access to the INDIGO infrastructure:
  - You can use the orchestrator
    - You will need to be registered and a valid token

Extending the TOSCA template

- You can use the elastic cluster node type to create an elastic cluster:
  - tosca.nodes.indigo.ElasticCluster
- You can use the LRMS node type to automatically use a LRMS (torque, slurm, etc)
  - tosca.nodes.indigo.LRMS.WorkerNode
    - tosca.nodes.indigo.LRMS.WorkerNode.Torque

More examples

INDIGO - DataCloud

- You can find more examples in:
  - https://github.com/indigo-dc/tosca-types/tree/master/examples
  - https://github.com/indigo-dc/tosca-templates
- All the implementations are open source
- Pull requests are welcome !