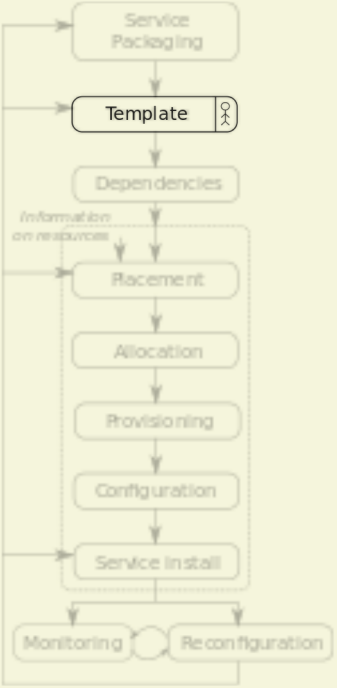
Toward a Domain Specific Language (DSL) to deploy a multi-region OpenStack

Hélène Coullon, Dimitri Pertin
Inria, LS2N, IMT Atlantique, Nantes, France
helene.coullon@inria.fr, dimitri.pertin@inria.fr

Christian Pérez
Inria, LIP, ENS Lyon, Lyon, France
christian.perez@inria.fr

Deployment model



2. Template

- written by the deployment designer
- describe the deployment plan
- contain:
  - services
  - configuration
- extra levels of expressiveness:
  - **constraints** (e.g. CPU, RAM, OS)
  - **dependencies**
  - **orchestration** (e.g. scalability)

Decentralized OpenStack (1/2)

- Emerging applications are facing latency and bandwidth limitations due to cloud-centric architecture.
- Running on complex infrastructures:
  - Internet of Things (IoT)
    - schedule resources close to Things
  - Radio Access Network (RAN)
    - schedule resources in base stations
- Toward a decentralized IaaS manager to manage such infrastructure.
- OpenStack is the de-facto IaaS manager.

Decentralized OpenStack (2/2)

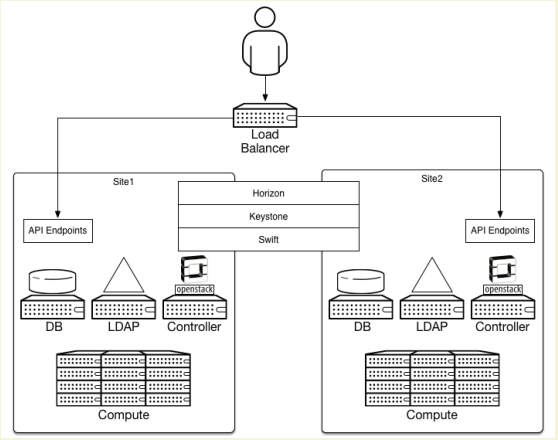
How to operate a decentralized OpenStack?

- Nova cells, Tricircle
- ...
- OpenStack regions
  - in this talk, we focus on OpenStack regions

OpenStack regions

An OpenStack region segregates different OpenStack instance.

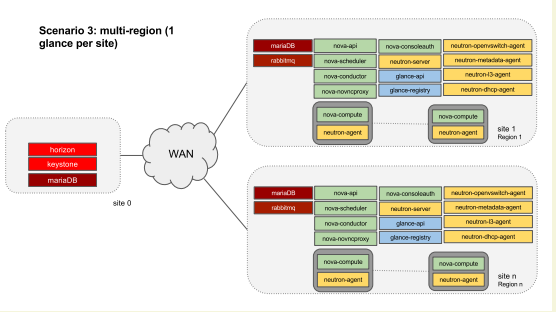
- Each region is an OpenStack deployment
- Including separate API endpoints for services
- Different regions share one Keystone (access control) and Horizon



Multi-region scenarios (1/3)

(shared Keystone/Horizon)

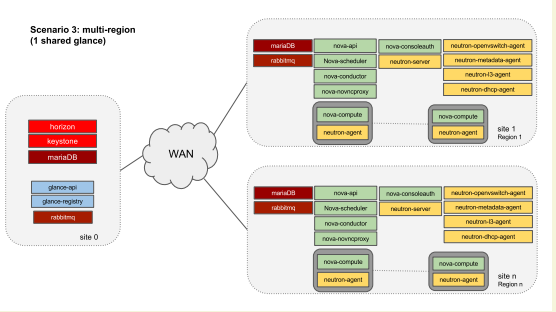
- 1 deployment per region
  - region_0: shared services
  - region_n: local services
- each service must be registered to shared Keystone



Multi-region scenarios (2/3)

(shared Keystone/Horizon + Glance)

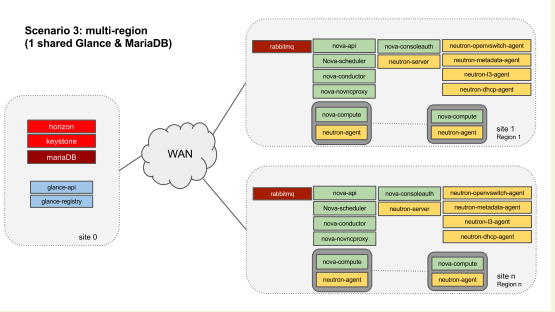
- 1 deployment per region
  - region_0: shared services
  - region_n: local services
- each service must be registered to shared Keystone
- nova must be configured to use the shared Glance



Multi-region scenarios (3/3)

(shared Keystone/Horizon + Glance + MariaDB)

- 1 deployment per region
  - region_0: shared services
  - region_n: local services
- each service must be registered to shared Keystone
- nova must be configured to use the shared Glance
- each service must be configured to use the shared MariaDB

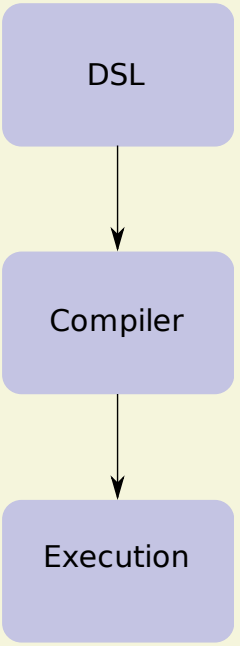


Deploying multi-region is hard

- We previously analyzed tools to deploy an OpenStack instance
  - Kolla, Enos, Juju, Kubernetes, TripleO
- But deploying multiple regions is hard because:
  - multi-region is not enabled in the deployment tools
  - when possible, many manual steps are required
- We need tools to automating multi-region deployment steps

DSL to deploy multi-region

1. DSL
    - express multi-region template
2. Compiler
    - compute required artifact files
3. Execution
    - call multiple times deployment tool



Agenda

1. Single to multi-region: a case-study with Enos
    - Enos single region
    - Enos multi-region
2. A closer look at OpenStack deployment workflow
    - Single region deployment workflow
    - Multi-region deployment workflows
3. Description of a DSL for multi-region deployments
    - DSL
    - Compiler
    - Execution
4. Conclusion and Future Works

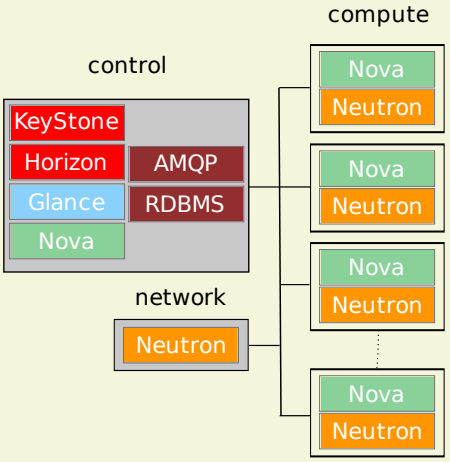
1. Single to multi-region

A Case-study with Enos

Enos single region

Ansible Inventory:

```
[mariadb:children]
control

[rabbitmq:children]
control

[keystone:children]
control

(...)

[neutron-controller:children]
network

(...)
```



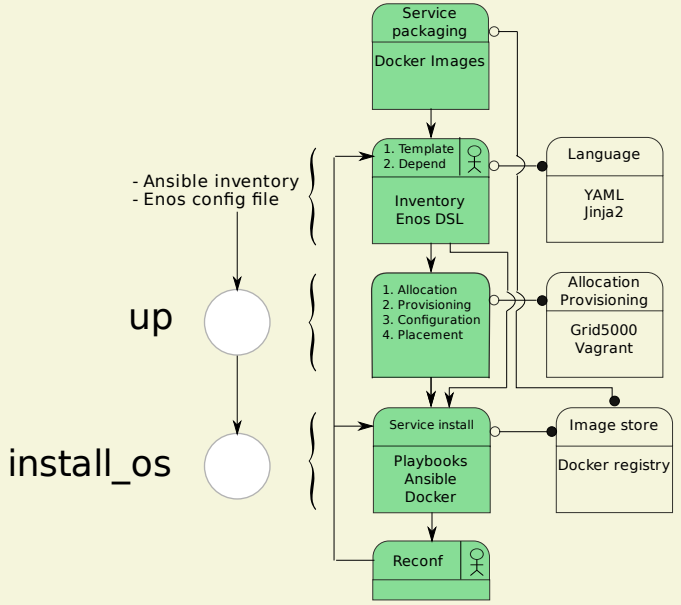
Enos DSL:

```
resources:
    paravance:
        control: 1
        network: 1
```

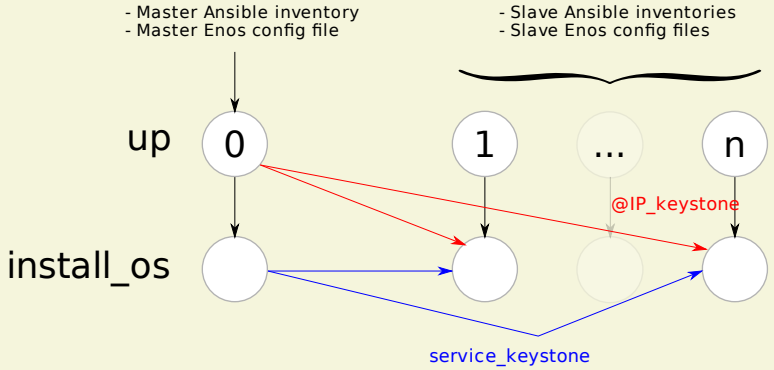
Enos single region

Enos worklow to deploy a region:

1. up: map resources to reachable @ip
2. install_os:
    1. generate service configuration files
    2. install and run services



Enos multi-region workflow



Example with deployment of multi-region scenario

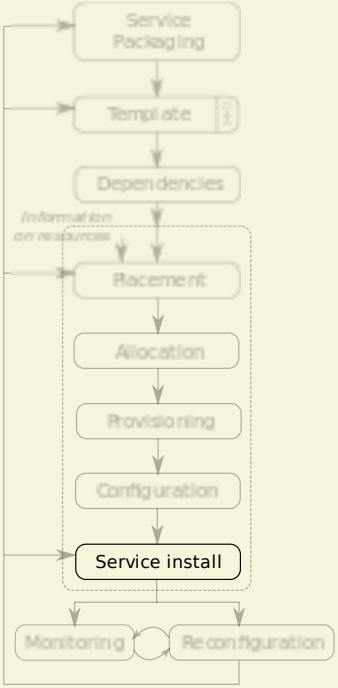
- Shared Keystone in region 0

Enos multi-region deployment is manually handled

1. Keystone @IP for OS service configuration
2. Keystone service for service registering

2. A Closer look at OpenStack deployment workflow

Deployment model



8. Service install

- build configuration files
- install services
- run services

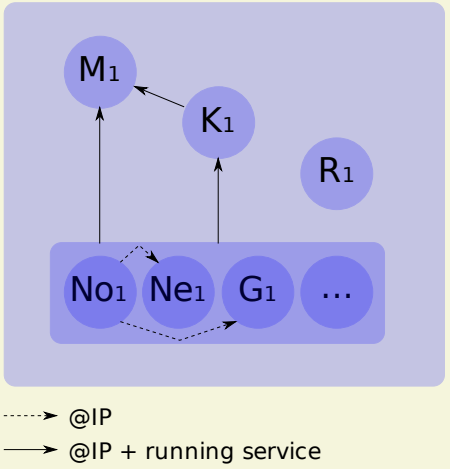
Single region workflow

Deployment workflow:

1. MariaDB
2. Keystone
3. OpenStack services
    - neutron (Ne), nova (No), glance (G)
    - any other optional projects*

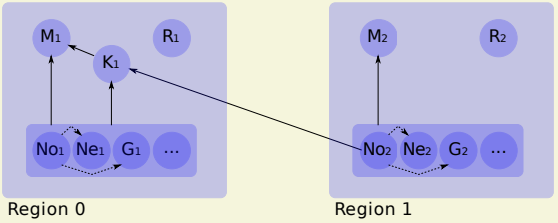
Each OpenStack service needs to:

- create a database in mariadb (M)
- register its endpoint in keystone (K)
- discover other services with keystone
    - except Nova (!) which
        - requires Neutron @IP
        - requires Glance @IP



Multi-region workflow (1/2)

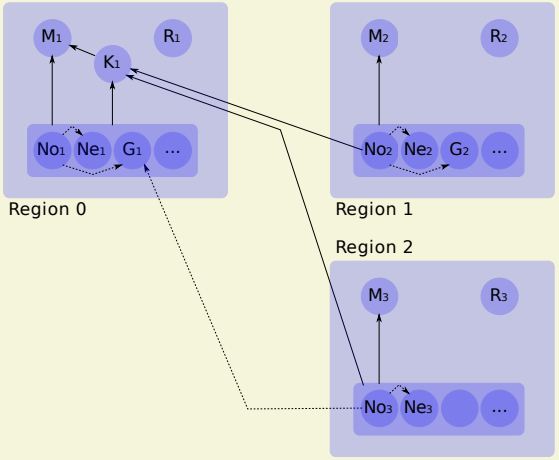
(shared Keystone)



- region_2 OS services require:
  - **K1 @IP** (obtained from up1) and **M2 @IP** (obtained from up2)
  - running **K1 and M2**
  - No(n) needs Ne(n) and G(n) @IP (obtained from up(n))

Multi-region workflow (2/2)

(shared Keystone + Glance)



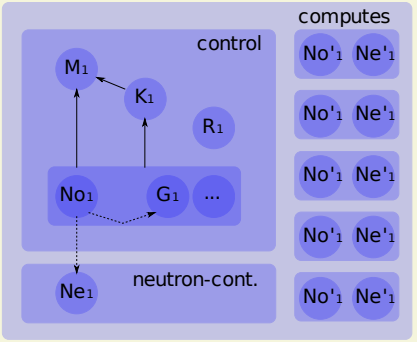
- region_3 OS services require:
  - **K1 @IP** (obtained from up1) and **M3 @IP** (obtained from up3)
  - running **K1 and M3**
  - No(3) needs Ne(3) and G(1) @IP (obtained from up1)

3. Description of a DSL for multi-region deployments

DSL proposition

Region 0 description (written in yml)

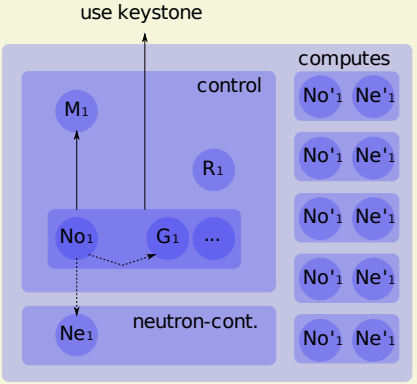
```
region_keystone:
  provide:
    control:
      services:
        - mariadb
        - keystone
        - rabbitmq
        - nova-controller
        - glance
    neutron-controller
    compute:
      services:
        - nova-agent
        - neutron-agent
      number: 5
```



DSL proposition

Region 1 description (written in yml)

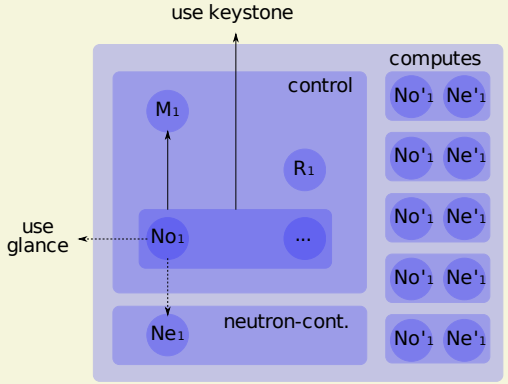
```
region_glance:
  provide:
    control:
      services:
        - mariadb
        - rabbitmq
        - nova-controller
        - glance
    neutron-controller
    compute:
      services:
        - nova-agent
        - neutron-agent
      number: 5
  use:
    keystone
```



DSL proposition

Region 2 description (written in yml)

```
region_simple:
  provide:
    control:
      services:
        - mariadb
        - rabbitmq
        - nova-controller
    neutron-controller
    compute:
      services:
        - nova-agent
        - neutron-agent
      number: 5
  use:
    keystone
    glance
```



DSL proposition

Region 2 description (written in yml)

```
region_simple:
  provide:
    control:
      services:
        - mariadb
        - rabbitmq
        - nova-controller
    neutron-controller
    compute:
      services:
        - nova-agent
        - neutron-agent
      number: 5
  use:
    keystone
    glance
```

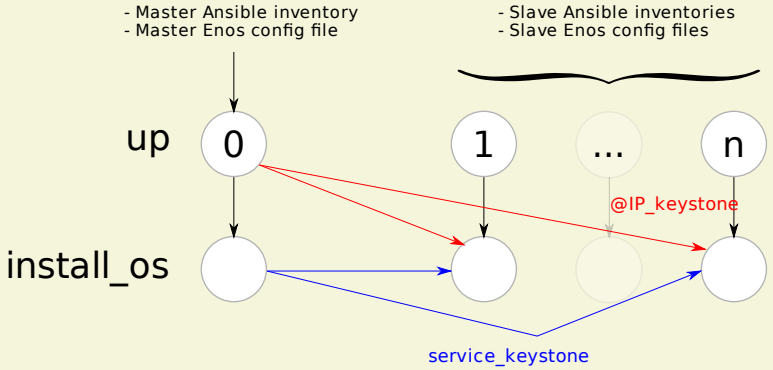
Assembly:

```
region_keystone:
  region0

region_glance:
  region1
      use:
        region0.keystone

region_simple:
  region2
      use:
        region0.keystone
        region1.glance
```

Compiler

- Based on a multi-region description written with our DSL
- Compiler generate for each region
  - an Ansible inventory
  - an Enos configuration file

Execution



1. Run in parallel `'enos up'` for each region
2. Wait until region[0] `up` is finished
3. Run `'enos install_os'` for the master region
4. Set Keystone @IP in config files for each region
5. Wait until region[0].keystone is running
6. Run `'enos intall_os'` for each slave region

3. Conclusion

Conclusion

Thanks to DSL + compiler + execution script, we are able to easily deploy a multi-region OpenStack

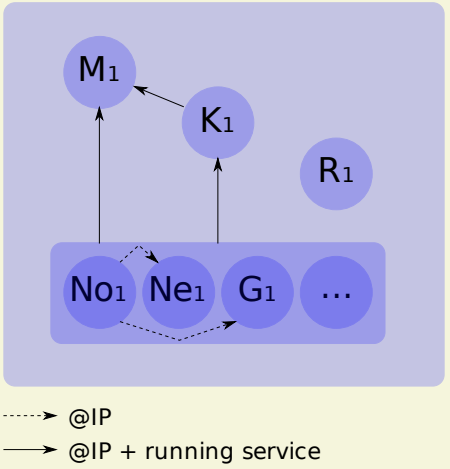
- We **analyzed** the deployment process of Enos
  - showed why it is hard to deploy multi-region
- We **analyzed** the deployment process of OpenStack services
  - for mono-region
    - understand intra-region dependencies
  - for multi-region
    - understand inter-region dependencies

Conclusion

- We proposed tools to ease multi-region deployments with Enos
  - a **DSL** to express multi-region deployment
  - a **compiler** to generate the required template files
  - an **execution script** to automatically manage:
    - the workflow of Enos tasks;
    - editing configuration files.

Future Works

- Implement this DSL/compiler on top of/inside Enos
- Because we use a DSL:
  - if an OpenStack service changes in the future, our process is broken
  - we do not want to rewrite the DSL
  - provide a mechanism to describe a service deployment workflow



Future Works

- Generalize our work
  - not specific to Enos
  - not specific to OpenStack