

Production Deployment Tools for IaaSes: an Overall Model and Survey

Hélène Couillon, Dimitri Pertin

Inria, LS2N, IMT Atlantique, Nantes, France

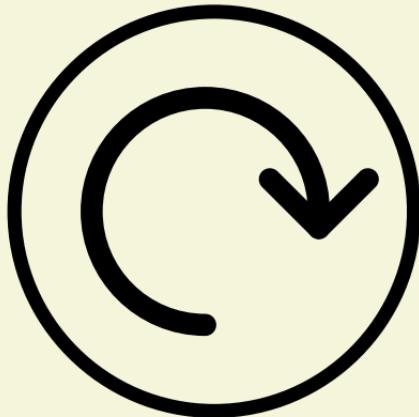
helene.couillon@inria.fr,
dimitri.pertin@inria.fr

Christian Pérez

Inria, LIP, ENS Lyon, Lyon, France

christian.perez@inria.fr

Software architecture



Monolithic

- single entity
- multi-tasks



Modular

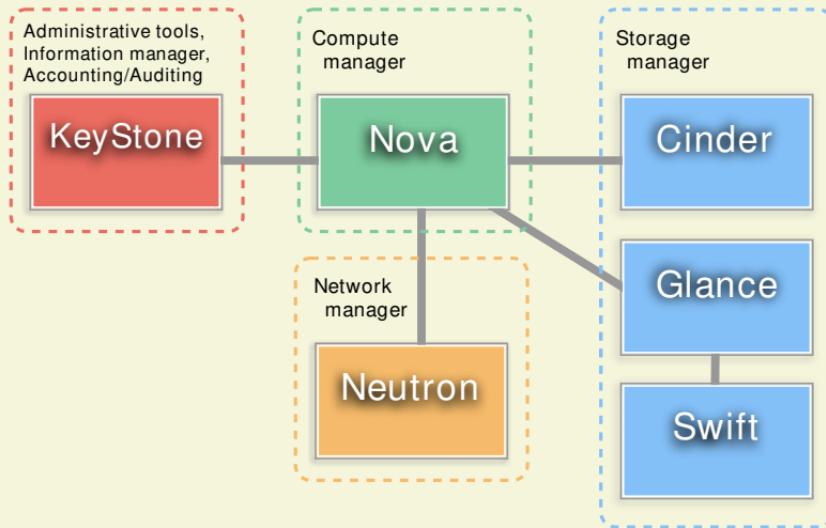
- separation of concerns
- multiple modules/services
- interconnected through interfaces*

Distributed software



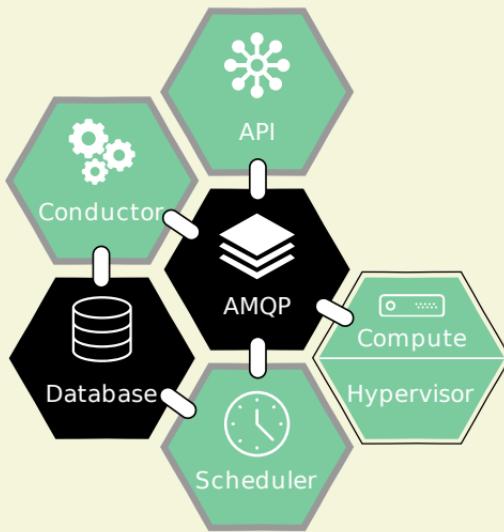
- multiple services
- distributed over multiple servers
- interconnected by a network

OpenStack



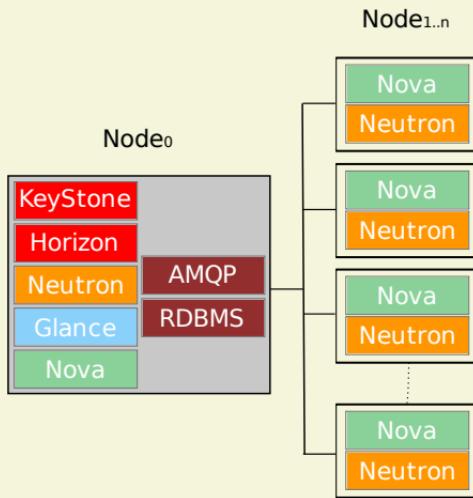
- IaaS manager: Compute, Storage, Network
- Modular architecture (composed of more than 30 projects)
- Communication by REST APIs, Message Queues (AMQP) and Databases

Example: Nova components



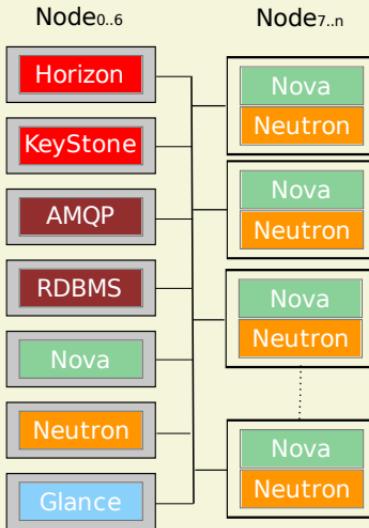
- 12 services in nova (only four represented here in green)
- **Control** (api, conductor, scheduler, highlighted in gray) and **compute** services
- More than 150 services in OpenStack (based on Kolla container images)

Deployment scenarios (1/3)



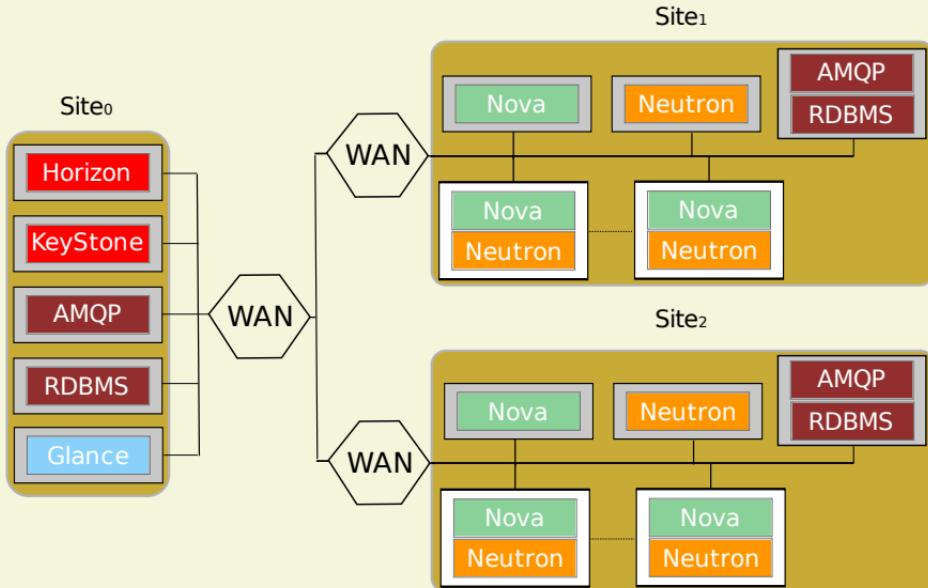
Single controller, multiple compute nodes

Deployment scenarios (2/3)



Multiple controllers, multiple compute nodes

Deployment scenarios (3/3)



Multiple controllers, multiple compute nodes, over multiple sites

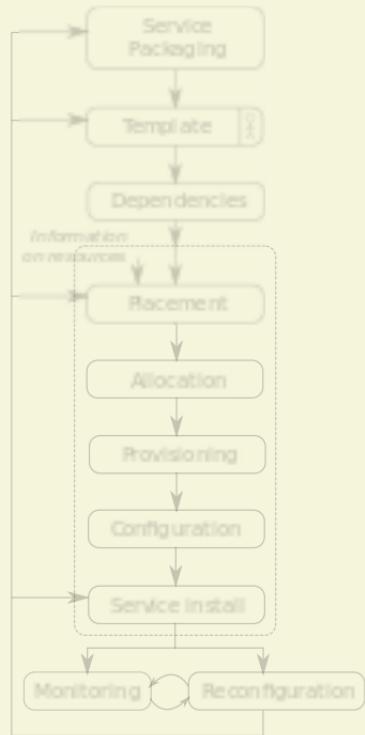
How are deployed such components?

Agenda

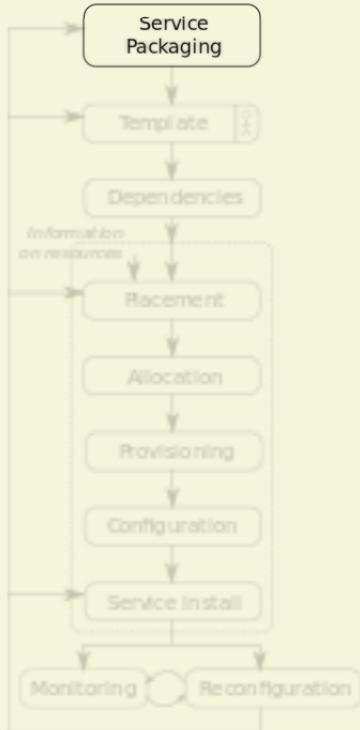
1. Application deployment model
 - Based on 10 steps
2. OpenStack deployment tools
 - Kolla, Enos, Juju, Kubernetes, TripleO
3. Discussions
 - Leveraging the model and the studied deployment tools
4. Conclusion

1. Application deployment model

Deployment model



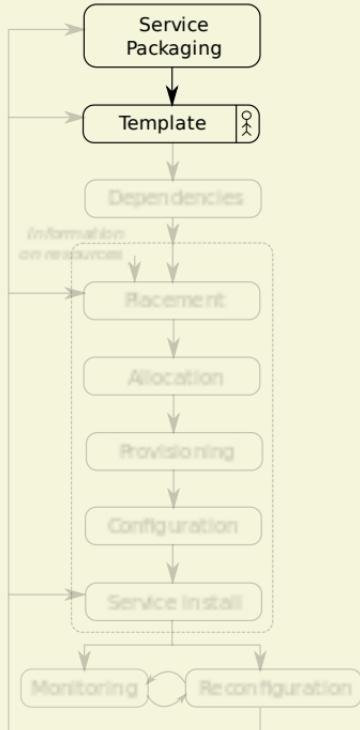
Deployment model



1. Service packaging

- prepare the software to be deployed
- split into multiple modules
- e.g. Docker containers, VM images

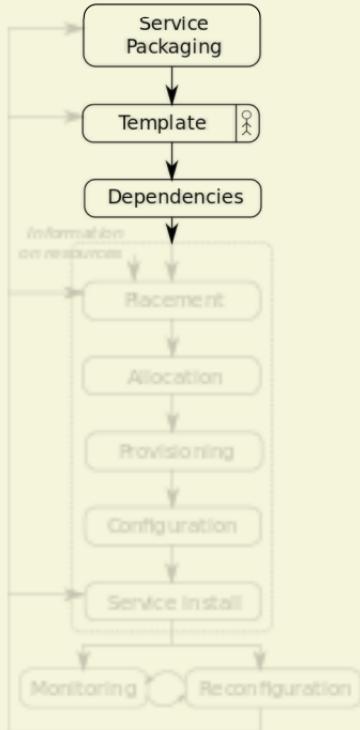
Deployment model



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)

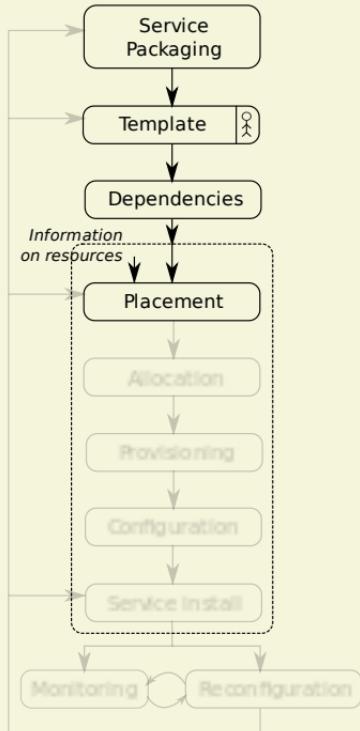
Deployment model



3. Dependencies

- read relations described in template
- can:
 - generate a dependency graph
 - rely on sequences
- can impact the 5 following steps

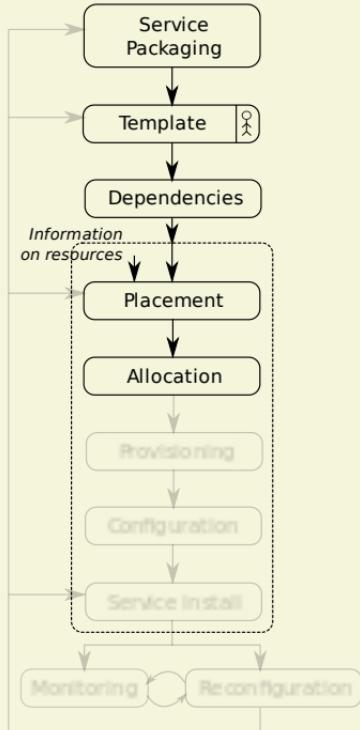
Deployment model



4. Placement

- map services to resources
- depending on declared constraints
- require info on resources
- can be:
 - **manual** or **automatic**
 - handled **internally** or **externally**

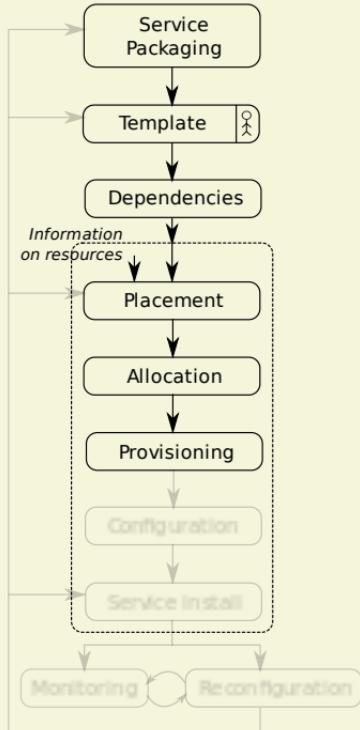
Deployment model



5. Allocation

- book resources
- can be:
 - **manual** or **automatic**
 - handled **internally** or **externally**

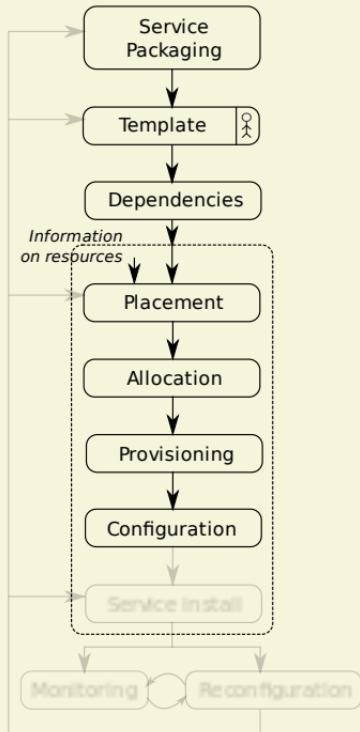
Deployment model



6. Provisioning

- set an environment on resources
- e.g. install OS, configure @IP, set ssh
- can be:
 - **manual** or **automatic**
 - handled **internally** or **externally**

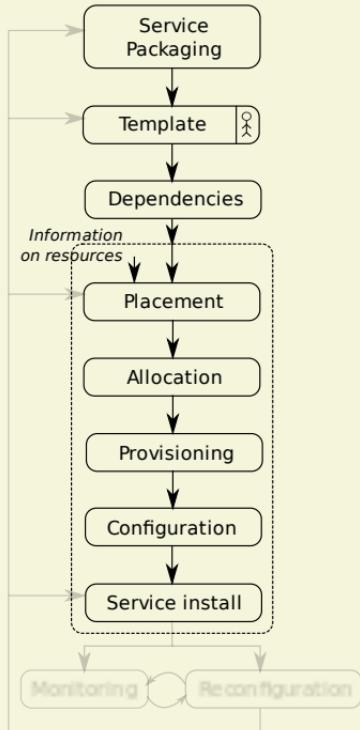
Deployment model



7. Configuration

- install environment dependencies
- e.g. python, Docker

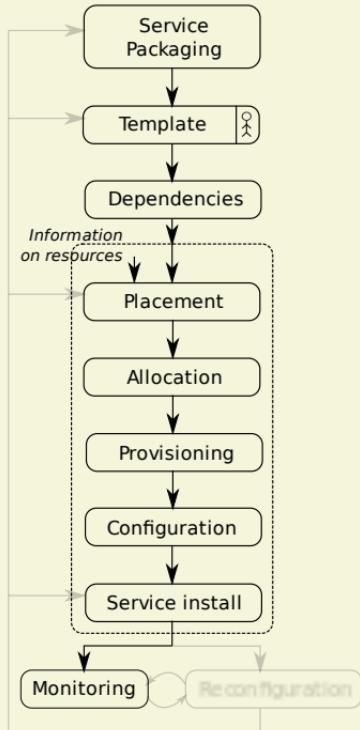
Deployment model



8. Service install

- build configuration files
- install services
- run services

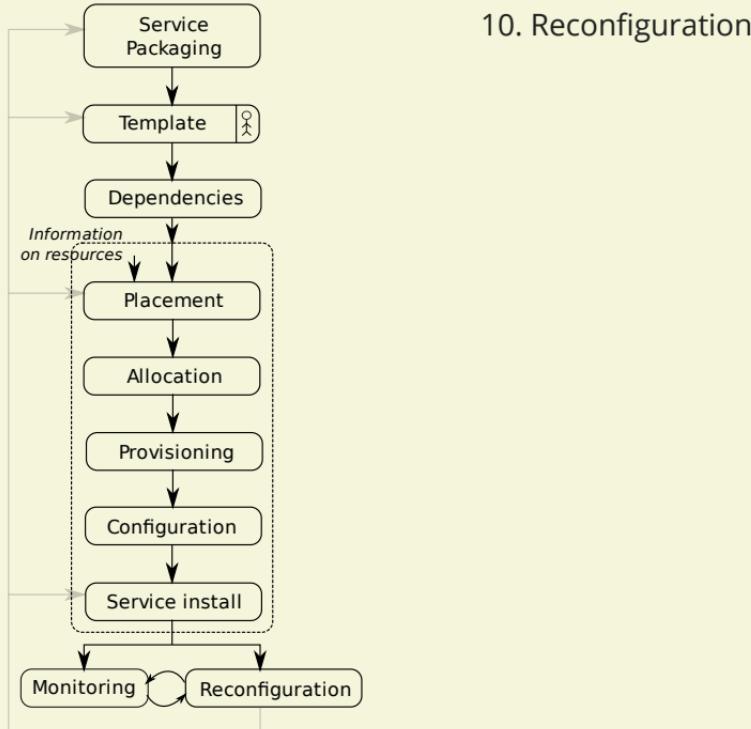
Deployment model



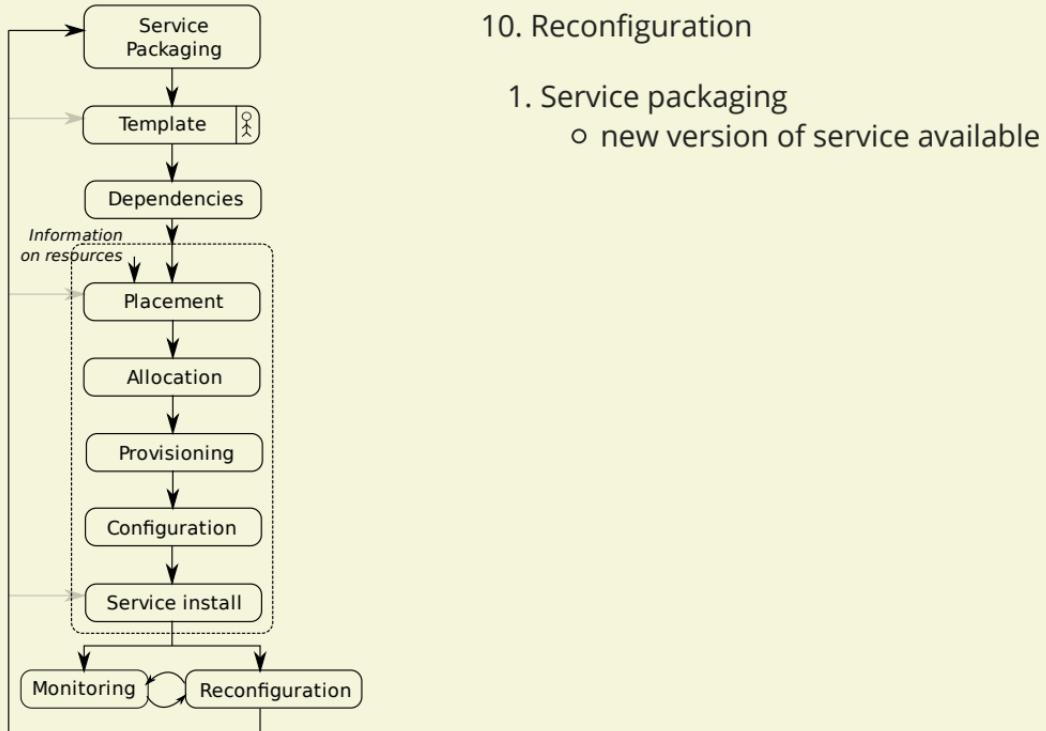
9. Monitoring

- keep track of resources
- check if dep and template diverge
- detect soft/hardware failures

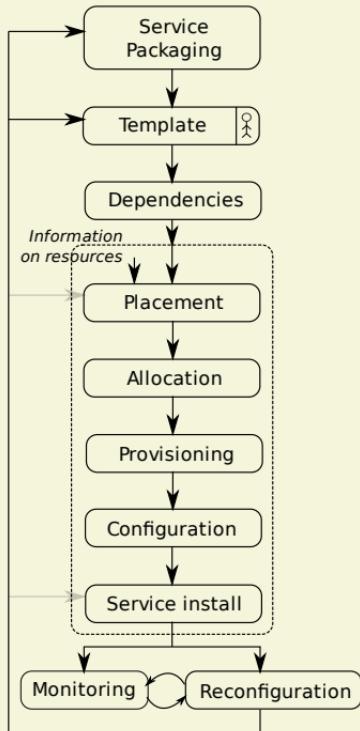
Deployment model



Deployment model



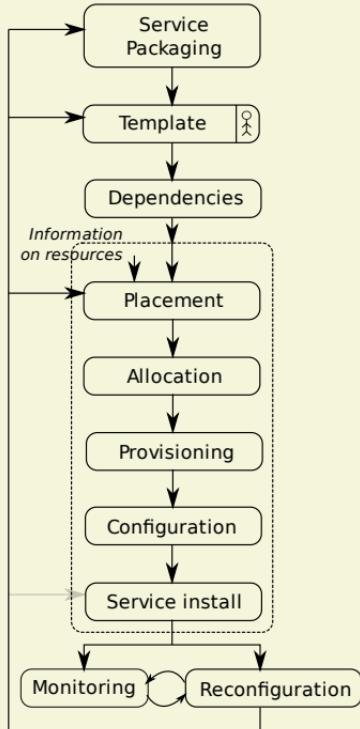
Deployment model



10. Reconfiguration

1. Service packaging
 - o new version of service available
2. Template
 - o modify the deployment topology

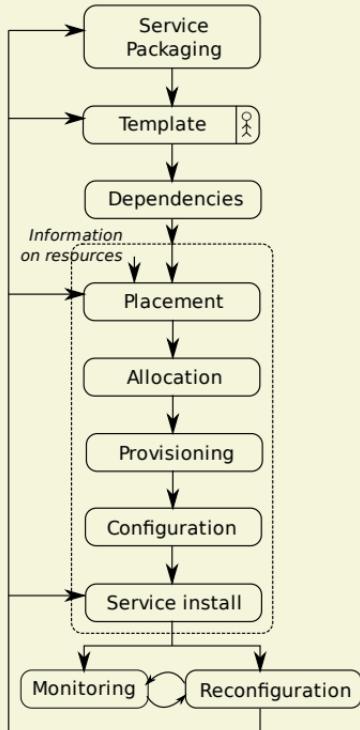
Deployment model



10. Reconfiguration

1. Service packaging
 - o new version of service available
2. Template
 - o modify the deployment topology
3. Placement
 - o rescheduling after hw failure

Deployment model

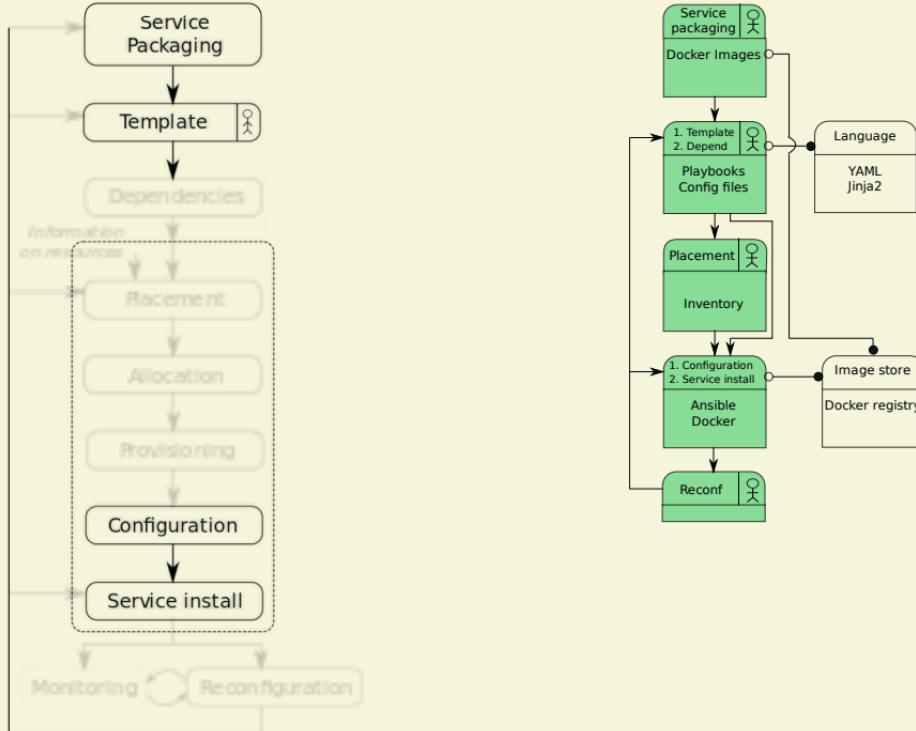


10. Reconfiguration

1. Service packaging
 - o new version of service available
 2. Template
 - o modify the deployment topology
 3. Placement
 - o rescheduling after hw failure
 4. Service install
 - o restart a failed service
- can be:
 - o **manual or automatic**

2. OpenStack deployment tools

1. Kolla



1. Kolla's template

Ansible Inventory:

```
[mariadb:children]
control
```

```
[rabbitmq:children]
control
```

```
[keystone:children]
control
```

```
[neutron:children]
network
```

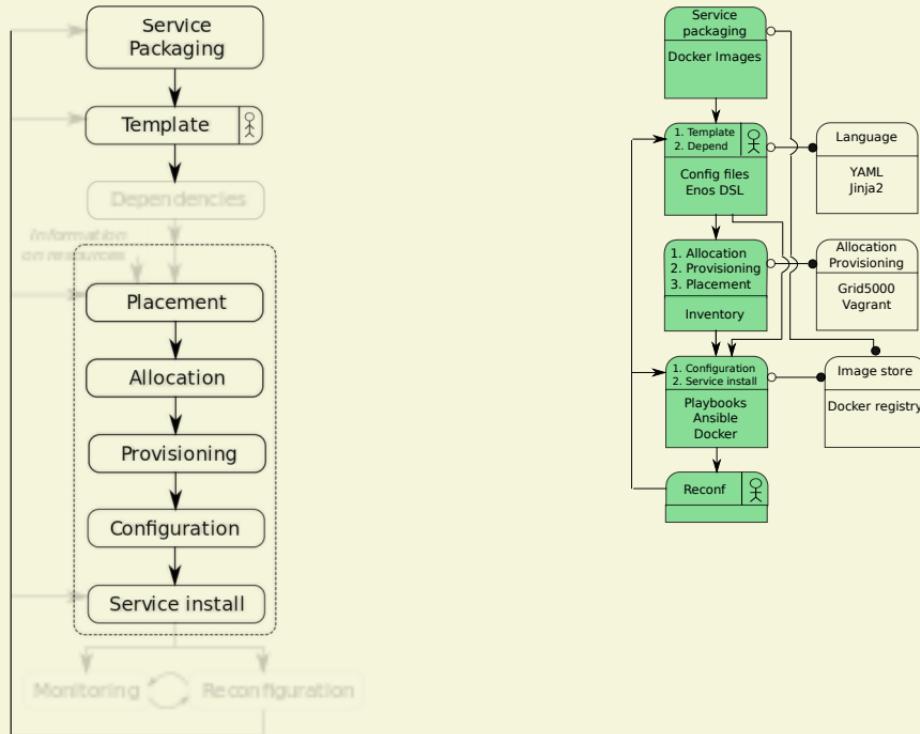
```
(...)
```

```
[control]
192.168.0.1
```

```
[network]
192.168.0.2
```

```
[compute]
192.168.1.1
192.168.1.2
192.168.1.3
```

2. Enos



2. Enos' template

Ansible Inventory:

```
[mariadb:children]
control
```

```
[rabbitmq:children]
control
```

```
[keystone:children]
control
```

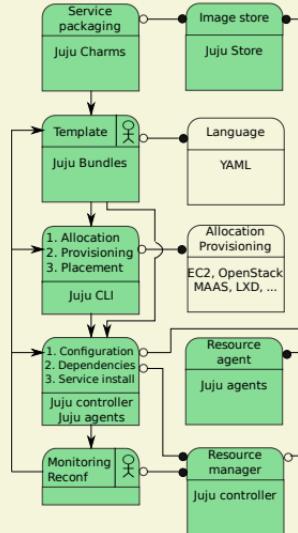
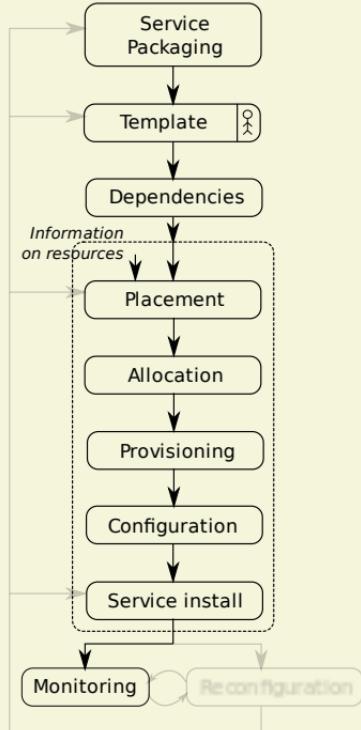
```
[neutron:children]
network
```

```
(...)
```

Enos configuration file:

```
resources:
  paravance:
    control: 1
    network: 1
    scheduler: 1
    rabbitmq: 3
```

3. Juju



3. Juju's template

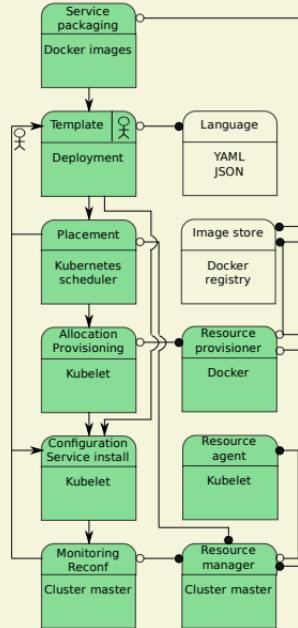
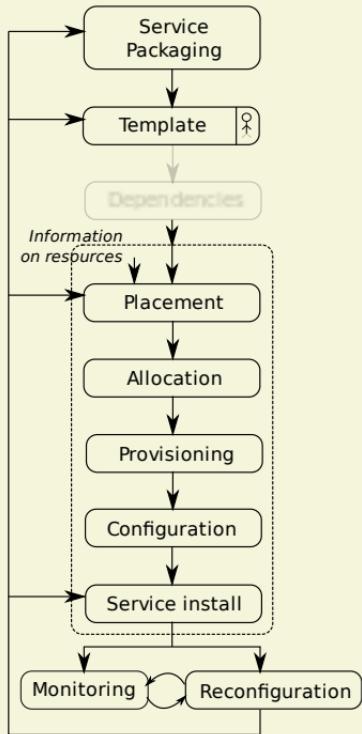
```
machines:
  '0':
    series: xenial
  '1':
    series: xenial
  '2':
    series: xenial
  '3':
    series: xenial

relations:
  - - nova-compute:amqp
    - rabbitmq-server:amqp
  - - nova-compute:image-service
    - glance:image-service
  - - nova-cloud-controller:cloud-compute
    - nova-compute:cloud-compute
  - - nova-cloud-controller:amqp
    - rabbitmq-server:amqp
  - - nova-compute:neutron-plugin
    - neutron-openvswitch:neutron-plugin

(...)

services:
  nova-compute:
    charm: cs:nova-compute-264
    num_units: 3
    options:
      enable-live-migration: true
      enable-resize: true
      migration-auth-type: ssh
    to:
      - '1'
      - '2'
      - '3'
```

4. Kubernetes

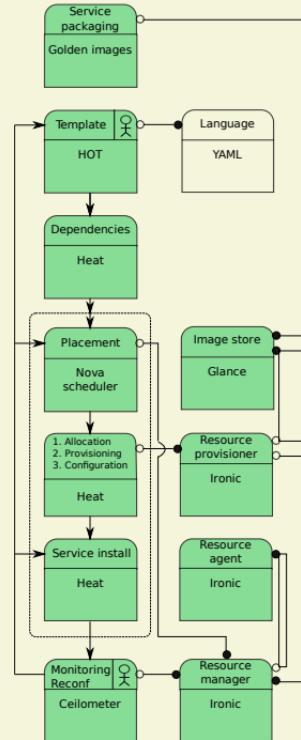
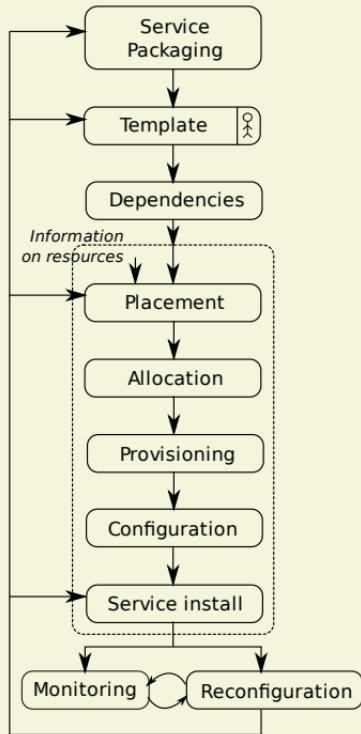


4.Kubernetes' template

```
kind: ReplicaSet
metadata:
  name: frontend
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
    matchExpressions:
      - {key: tier, operator: In, values: ["frontend"]}
  template:
    metadata:
      labels:
        app: guestbook
        tier: frontend
    spec:
      containers:
        - name: php-redis
          image: gcr.io/google_samples/gb-1
          resources:
            requests:
              cpu: 100m
              memory: 100Mi
      env:
```

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: frontend-scaler
spec:
  scaleTargetRef:
    kind: ReplicaSet
    name: frontend
  minReplicas: 3
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
```

5. TripleO



5. HOT

```
volume:  
  type: OS::Cinder::Volume  
  properties:  
    size: 1  
  
volume_attachment:  
  type: OS::Cinder::VolumeAttachment  
  properties:  
    volume_id: { get_resource: volume }  
    instance_uuid: { get_resource: server1 }  
  
server1:  
  type: OS::Nova::Server  
  properties:  
    image: myimage  
    flavor: m1.small  
  depends_on: [ server2, server3 ]  
  
server2:  
  type: OS::Nova::Server  
  (...)  
  
server3:  
  type: OS::Nova::Server  
  (...)
```

```
asg:  
  type: OS::Heat::AutoScalingGroup  
  properties:  
    max_size: 4  
    min_size: 1  
    resource: volume_attachment  
  
# Scaling Policies  
group_scaleup_policy:  
  type: OS::Heat::ScalingPolicy  
  properties:  
    adjustment_type: change_in_capacity  
    auto_scaling_group_id: {get_resource: asg }  
    scaling_adjustment: 1  
  
# Set Alarms  
cpu_alarm_high:  
  type: OS::Aodh::Alarm  
  properties:  
    description: Scale-up if volume > 5  
    meter_name: volume.size  
    statistic: avg  
    threshold: 5  
    alarm_actions:  
      - {get_attr: [group_scaleup_policy]}
```

3. Deployment Tool Comparison

Deployment tool comparison

Comparison table

	Kolla	Enos	Juju	Kubernetes	TripleO
Environment					
Packaging					
Template (all manual)					
Dependencies					
Placement					
Allocation					
Provisioning					
Monitoring					
Reconfiguration					

TABLE I
COMPARATIVE TABLE OF THE STUDIED DEPLOYMENT TOOLS BASED ON THE STEPS DEFINED IN SECTION II

Deployment tool comparison

1) Environment heterogeneity

	Kolla	Enos	Juju	Kubernetes	TripleO
Environment	containers	containers	any	containers	bare-metal
Packaging					
Template (all manual)					
Dependencies					
Placement					
Allocation					
Provisioning					
Monitoring					
Reconfiguration					

TABLE I

COMPARATIVE TABLE OF THE STUDIED DEPLOYMENT TOOLS BASED ON THE STEPS DEFINED IN SECTION II

Deployment tool comparison

2) Size of service packages

	Kolla	Enos	Juju	Kubernetes	TripleO
Environment	containers	containers	any	containers	bare-metal
Packaging	containers	containers	scripts	containers	disk images
Template (all manual)					
Dependencies					
Placement					
Allocation					
Provisioning					
Monitoring					
Reconfiguration					

TABLE I

COMPARATIVE TABLE OF THE STUDIED DEPLOYMENT TOOLS BASED ON THE STEPS DEFINED IN SECTION II

Deployment tool comparison

3) Expressiveness of the template

	Kolla	Enos	Juju	Kubernetes	TripleO
Environment	containers	containers	any	containers	bare-metal
Packaging	containers	containers	scripts	containers	disk images
Template (all manual)	no relations no constraints no orchestration	no relations no constraints no orchestration	relations constraints no orchestration	no relations constraints orchestration	relations constraints orchestration
Dependencies					
Placement					
Allocation					
Provisioning					
Monitoring					
Reconfiguration					

TABLE I

COMPARATIVE TABLE OF THE STUDIED DEPLOYMENT TOOLS BASED ON THE STEPS DEFINED IN SECTION II

Deployment tool comparison

4) Dependency management

	Kolla	Enos	Juju	Kubernetes	TripleO
Environment	containers	containers	any	containers	bare-metal
Packaging	containers	containers	scripts	containers	disk images
Template (all manual)	no relations no constraints no orchestration	no relations no constraints no orchestration	relations constraints no orchestration	no relations constraints orchestration	relations constraints orchestration
Dependencies	manual sequence install	manual sequence install	automatic graph install	manual sequence install	automatic graph from placement to install
Placement					
Allocation					
Provisioning					
Monitoring					
Reconfiguration					

TABLE I

COMPARATIVE TABLE OF THE STUDIED DEPLOYMENT TOOLS BASED ON THE STEPS DEFINED IN SECTION II

Deployment tool comparison

5) Automatic placement/allocation/provisioning

	Kolla	Enos	Juju	Kubernetes	TripleO
Environment	containers	containers	any	containers	bare-metal
Packaging	containers	containers	scripts	containers	disk images
Template (all manual)	no relations no constraints no orchestration	no relations no constraints no orchestration	relations constraints no orchestration	no relations constraints orchestration	relations constraints orchestration
Dependencies	manual sequence install	manual sequence install	automatic graph install	manual sequence install	automatic graph from placement to install
Placement	manual extern/intern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Allocation	NS extern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Provisioning	NS extern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Monitoring					
Reconfiguration					

TABLE I
COMPARATIVE TABLE OF THE STUDIED DEPLOYMENT TOOLS BASED ON THE STEPS DEFINED IN SECTION II

Deployment tool comparison

6) Monitoring

	Kolla	Enos	Juju	Kubernetes	TripleO
Environment	containers	containers	any	containers	bare-metal
Packaging	containers	containers	scripts	containers	disk images
Template (all manual)	no relations no constraints no orchestration	no relations no constraints no orchestration	relations constraints no orchestration	no relations constraints orchestration	relations constraints orchestration
Dependencies	manual sequence install	manual sequence install	automatic graph install	manual sequence install	automatic graph from placement to install
Placement	manual extern/intern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Allocation	NS extern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Provisioning	NS extern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Monitoring	NS	NS	manual	manual/auto	manual/auto
Reconfiguration					

TABLE I
COMPARATIVE TABLE OF THE STUDIED DEPLOYMENT TOOLS BASED ON THE STEPS DEFINED IN SECTION II

Deployment tool comparison

7) Automatic reconfiguration and prerequisites

	Kolla	Enos	Juju	Kubernetes	TripleO
Environment	containers	containers	any	containers	bare-metal
Packaging	containers	containers	scripts	containers	disk images
Template (all manual)	no relations no constraints no orchestration	no relations no constraints no orchestration	relations constraints no orchestration	no relations constraints orchestration	relations constraints orchestration
Dependencies	manual sequence install	manual sequence install	automatic graph install	manual sequence install	automatic graph from placement to install
Placement	manual extern/intern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Allocation	NS extern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Provisioning	NS extern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Monitoring	NS	NS	manual	manual/auto	manual/auto
Reconfiguration	manual	manual	manual resource software	manual/auto resource (auto) software (auto)	manual/auto resource (manual) software (manual)
	template	template	template scaling	template (manual) scaling (auto)	template (manual) scaling (auto)

TABLE I
COMPARATIVE TABLE OF THE STUDIED DEPLOYMENT TOOLS BASED ON THE STEPS DEFINED IN SECTION II

Deployment tool comparison

8) IaaS as a deployment tool

	Kolla	Enos	Juju	Kubernetes	TripleO
Environment	containers	containers	any	containers	bare-metal
Packaging	containers	containers	scripts	containers	disk images
Template (all manual)	no relations no constraints no orchestration	no relations no constraints no orchestration	relations constraints no orchestration	no relations constraints orchestration	relations constraints orchestration
Dependencies	manual sequence install	manual sequence install	automatic graph install	manual sequence install	automatic graph from placement to install
Placement	manual extern/intern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Allocation	NS extern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Provisioning	NS extern	automatic extern/intern	automatic extern/intern	automatic intern	automatic intern
Monitoring	NS	NS	manual	manual/auto	manual/auto
Reconfiguration	manual	manual	manual resource software	manual/auto resource (auto) software (auto)	manual/auto resource (manual) software (manual)
	template	template	template scaling	template (manual) scaling (auto)	template (manual) scaling (auto)

TABLE I
COMPARATIVE TABLE OF THE STUDIED DEPLOYMENT TOOLS BASED ON THE STEPS DEFINED IN SECTION II

4. Conclusion

Conclusion

1. We gave a **model** of the application deployment process.
2. We used this model to **analyze** production tools to deploy OpenStack.
3. We used the model and this analysis to **compare** these tools:
 - each tool has its pros and cons
 - TripleO has several interesting advantages
 - expressiveness
 - extensive scheduler
 - automatic reconfiguration
 - OpenStack as an IaaS and a deployment tool