OPENINFRA DAYS POLAND

Kraków, 12.06.2019

TRIPLEO

an OpenStack Community Project

config-download explained

Piotr Kopeć, Red Hat

Agenda

1. What is TripleO?
2. TripleO evolution
    a. Before Pike/Queens
    b. Pike/Queens and after
3. Config-download
4. References and further reading

What is TripleO?

PLANNING

Network topology

Service parameters

Resource capacity

DEPLOYMENT

Pre-flight checks/validations

Deployment orchestration

Service configuration

OPERATIONS

Major version upgrades

Applying minor updates

Scaling up and down

Source: [2]

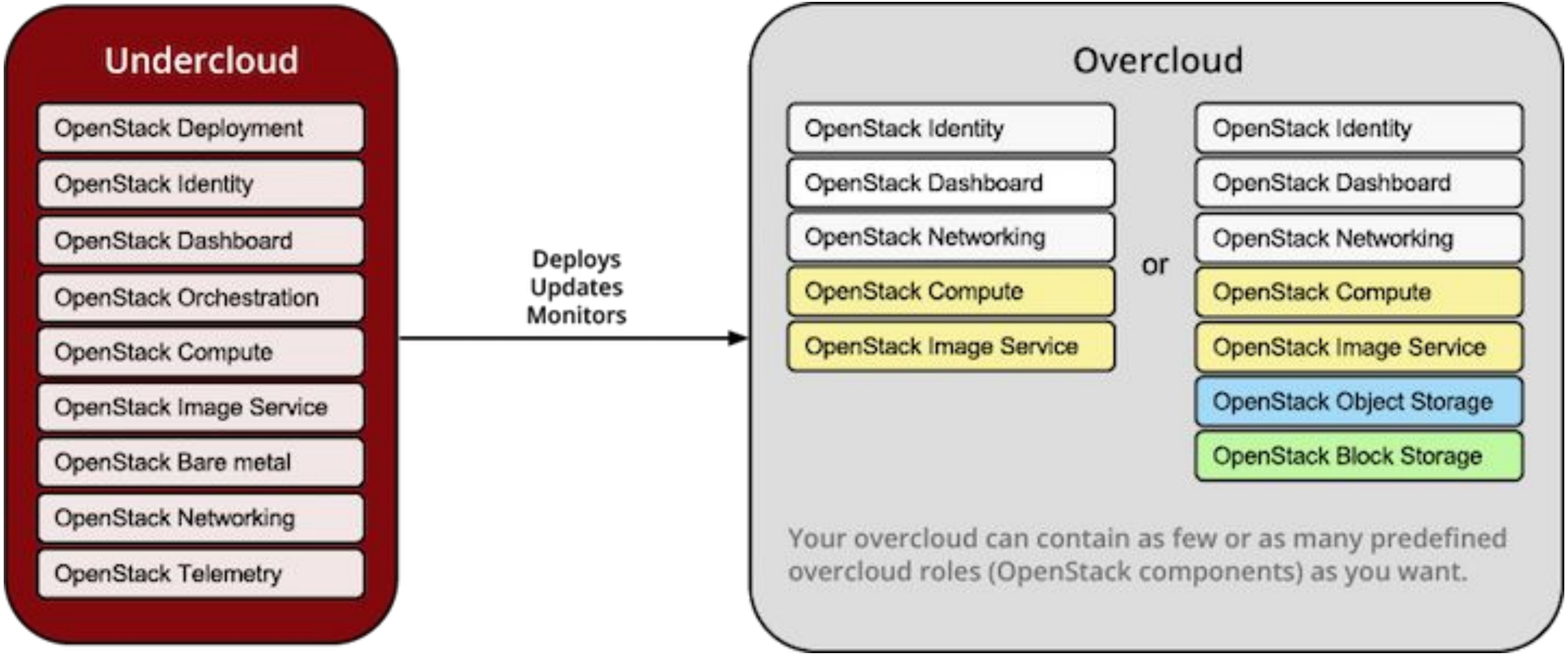
TripleO components

- Heat
- Nova
- Ironic
- Keystone
- Mistral
- Neutron
- Swift
- Zaqar

TRIPLEO

an OpenStack Community Project

Undercloud and Overcloud



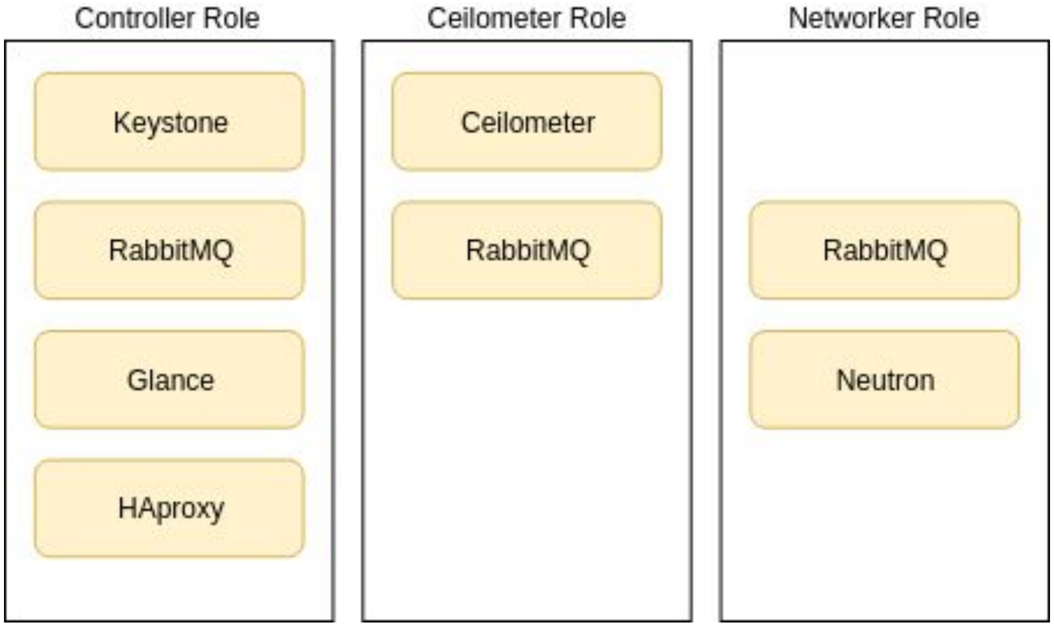
Your overcloud can contain as few or as many predefined overcloud roles (OpenStack components) as you want.

Source: [1]

Repositories

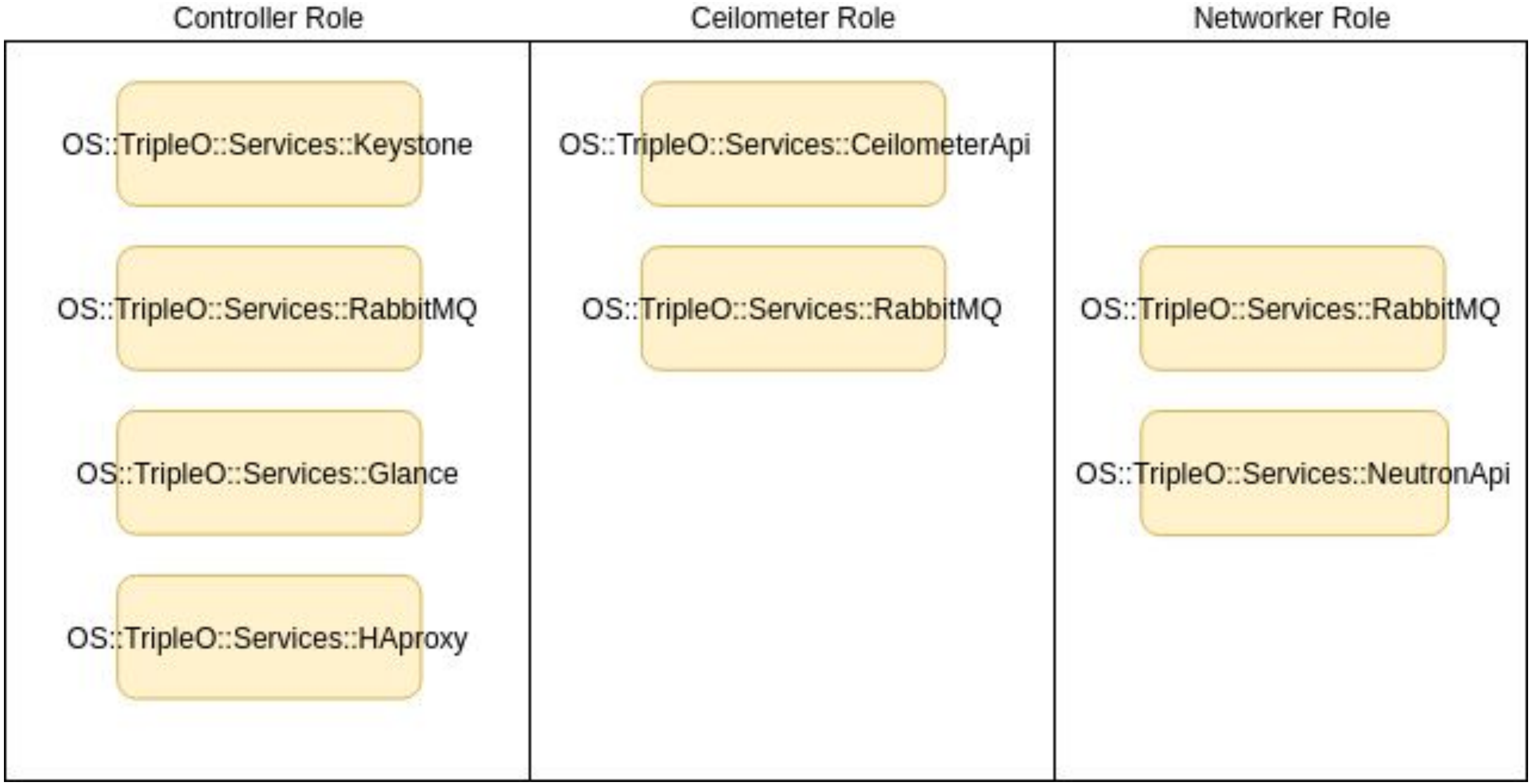
- Tripleo-common - libraries of common code including Mistral workflows and actions
- Tripleo-heat-templates - Heat templates necessary to deploy the overcloud
- Puppet-tripleo - contains various puppet modules for configuring services
  - Puppet-{service_name} - service specific puppet module

Roles



```
$ openstack overcloud profiles list
+--------------+-----------------+-----------------+
| Node Name    | Provision State | Current Profile |
+--------------+-----------------+-----------------+
| compute-0    | active          | compute         |
| controller-0 | active          | controller      |
+--------------+-----------------+-----------------+
```

Roles resources - roles_data.yaml



Controller Role

OS::TripleO::Services::Keystone

OS::TripleO::Services::RabbitMQ

OS::TripleO::Services::Glance

OS::TripleO::Services::HAproxy

Ceilometer Role

OS::TripleO::Services::CeilometerApi

OS::TripleO::Services::RabbitMQ

Networker Role

OS::TripleO::Services::RabbitMQ

OS::TripleO::Services::NeutronApi

Controller - roles_data.yaml

```
# roles_data.yaml
- name: Controller
  [...]
  networks:
    External:
      subnet: external_subnet
    InternalApi:
      subnet: internal_api_subnet
    Storage:
      subnet: storage_subnet
[...]
  ServicesDefault:
    - OS::TripleO::Services::Keystone
    - OS::TripleO::Services::GlanceApi
    - OS::TripleO::Services::HAproxy
  [...]
```

Evolution OS::TripleO::Services*

```
#overcloud-resource-registry-puppet.j2.yaml
resource_registry:
 OS::TripleO::Services::NovaApi: puppet/services/nova-api.yaml
```

Ocata

```
#overcloud-resource-registry-puppet.j2.yaml
resource_registry:
 OS::TripleO::Services::NovaApi: docker/services/nova-api.yaml
```

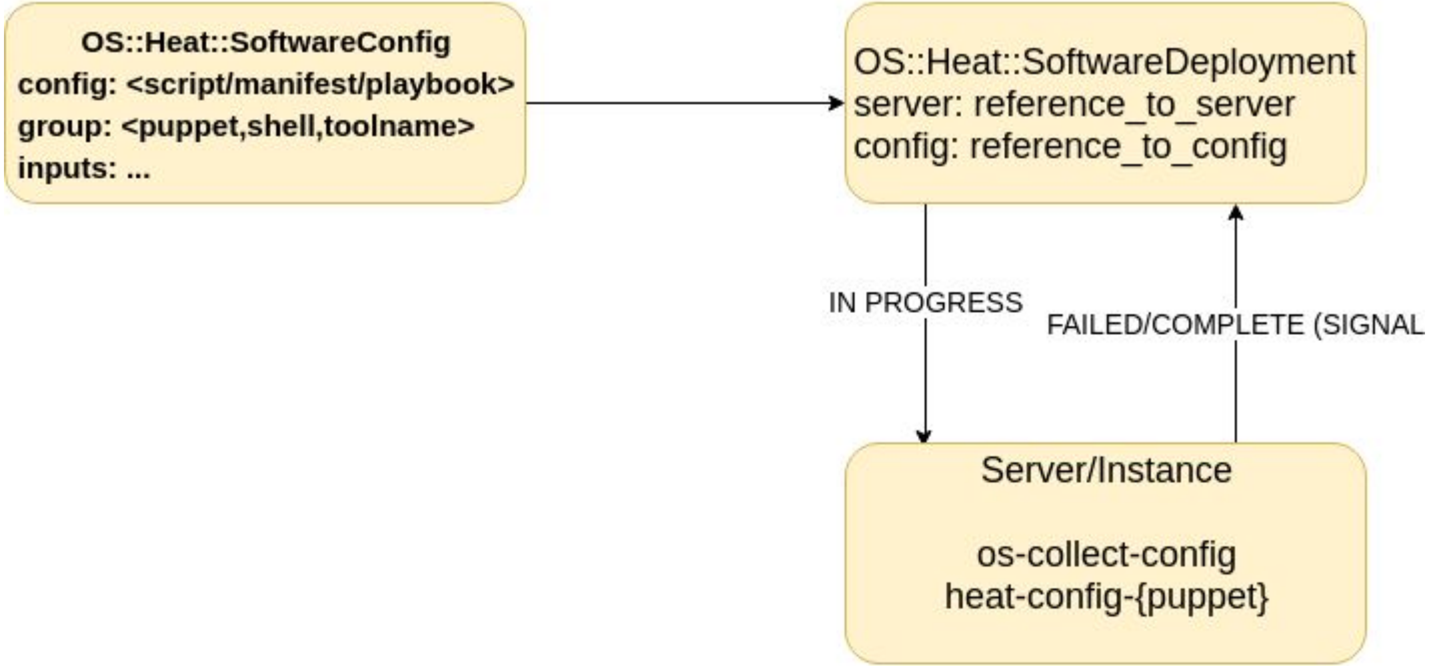
Queens

```
#overcloud-resource-registry-puppet.j2.yaml
resource_registry:
 OS::TripleO::Services::NovaApi: deployment/nova/nova-api-container-puppet.yaml
```

Stein

Evolution - Software Configuration

Before Queens:



OS::Heat::SoftwareConfig
config: <script/manifest/playbook>
group: <puppet,shell,toolname>
inputs: ...

OS::Heat::SoftwareDeployment
server: reference_to_server
config: reference_to_config

IN PROGRESS

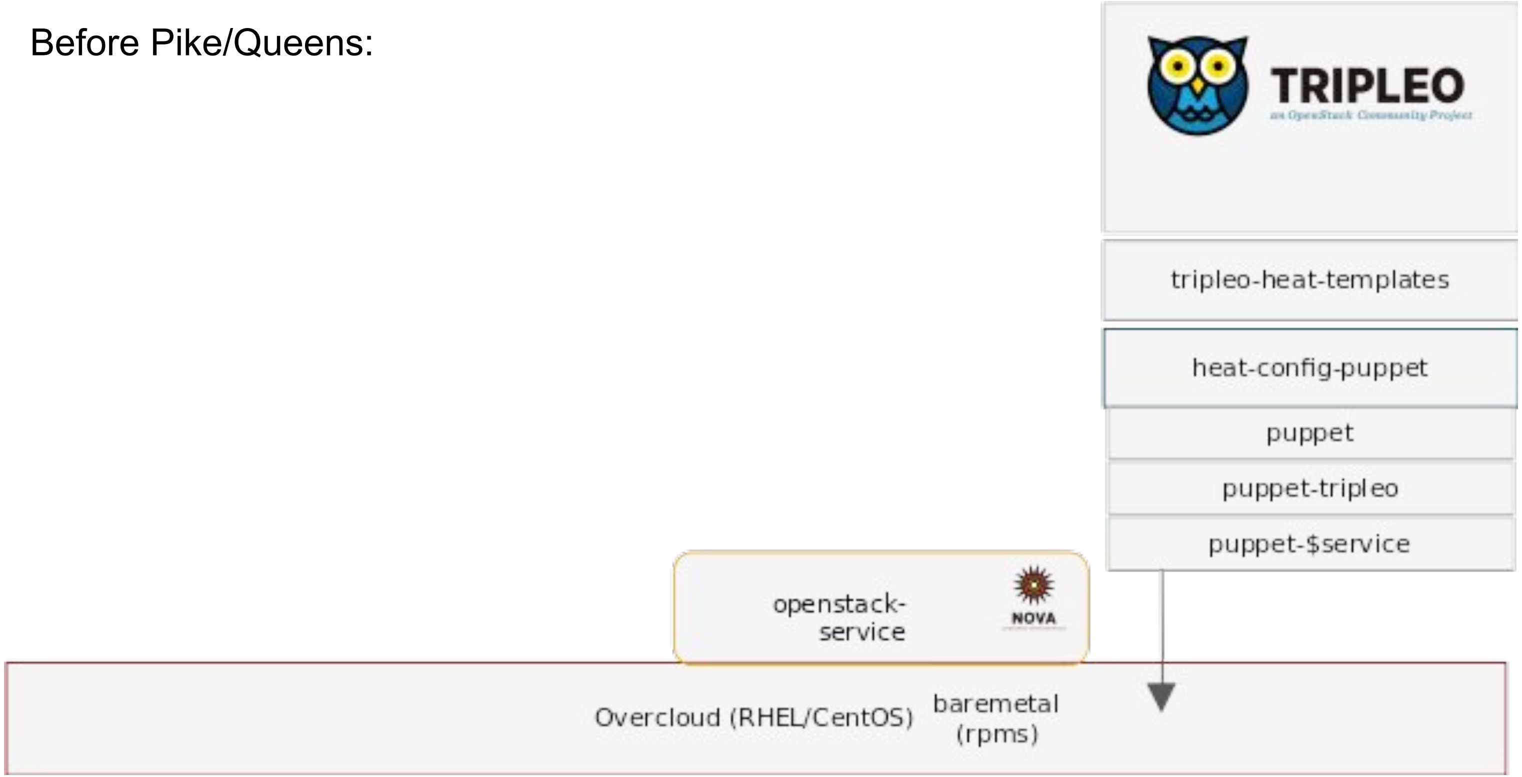
FAILED/COMPLETE (SIGNAL

Server/Instance

os-collect-config
heat-config-{puppet}

Evolution - Software Configuration

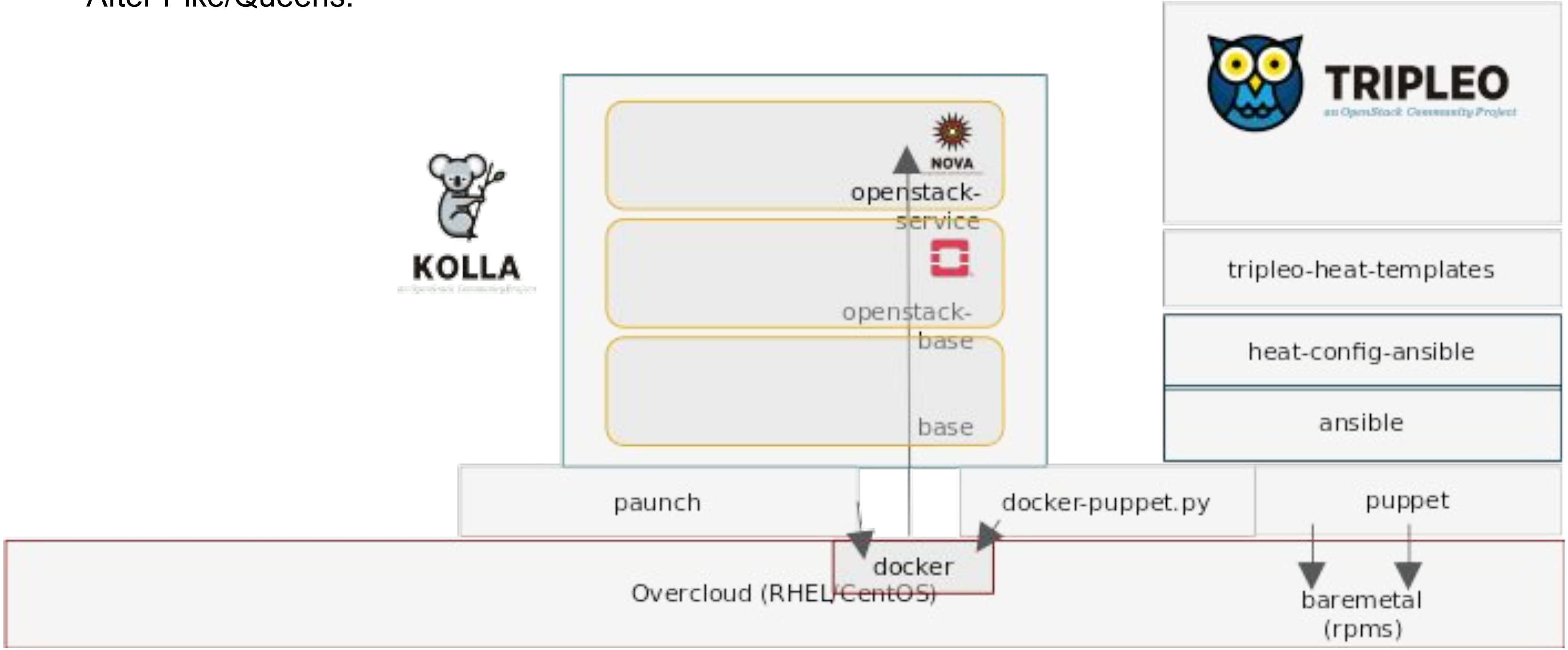
Before Pike/Queens:



Source: [2]

Evolution - Software Configuration

After Pike/Queens:



Source: [2]

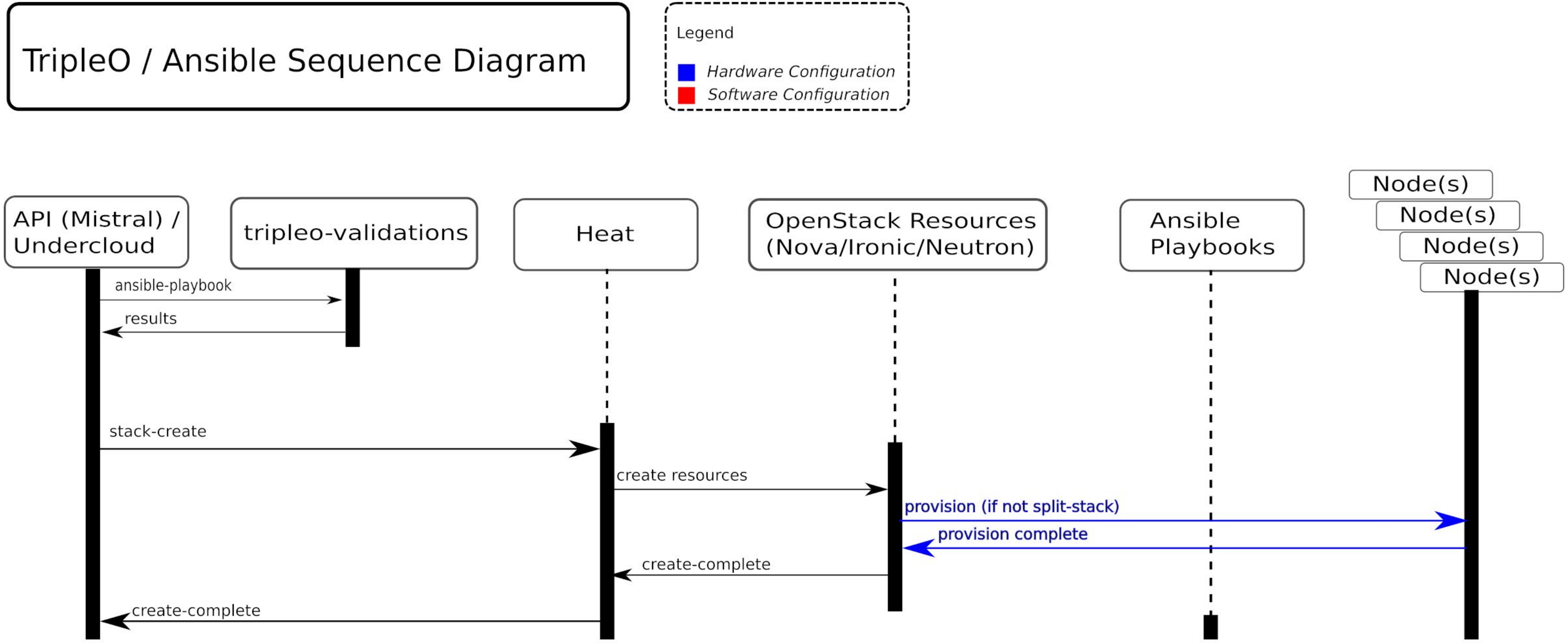
Containers

Paunch https://opendev.org/openstack/paunch/:

- Utility to launch and manage containers using YAML based configuration data
- Basically docker run wrapper

```
nova_api_db_sync:
   start_order: 0 # Runs before nova-conductor dbsync
   image: nova_api_image
   net: host
   detach: false
   user: root
   volumes: nova_api_bootstrap_volumes
    list_concat:
     - {get_attr: [ContainersCommon, volumes]}
     - {get_attr: [NovaApiLogging, volumes]}
     -
      - /var/lib/config-data/nova/etc/my.cnf.d/tripleo.cnf:/etc/my.cnf.d/tripleo.cnf:ro
      - /var/lib/config-data/nova/etc/nova/:/etc/nova/:ro
   command: "/usr/bin/bootstrap_host_exec nova_api su nova -s /bin/bash -c '/usr/bin/nova-manage api_db sync'"
```

Config-download - Heat part



TripleO / Ansible Sequence Diagram

Legend

🟦 *Hardware Configuration*

🟥 *Software Configuration*

API (Mistral) / Undercloud

tripleo-validations

Heat

OpenStack Resources (Nova/Ironic/Neutron)

Ansible Playbooks

Node(s)

Node(s)

Node(s)

Node(s)

ansible-playbook

results

stack-create

create resources

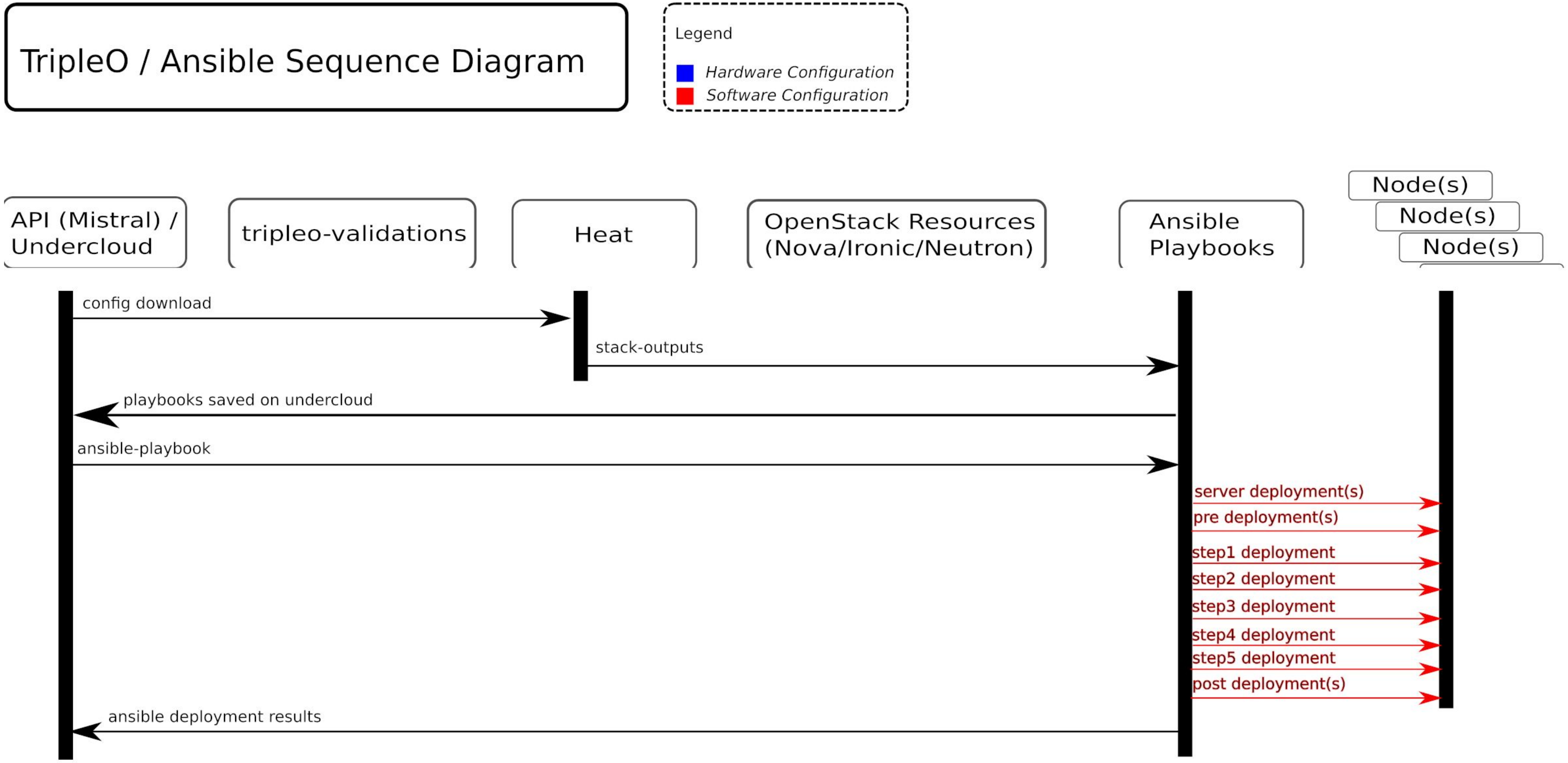
provision (if not split-stack)

provision complete

create-complete

create-complete

Config-download - Ansible part



TripleO / Ansible Sequence Diagram

Legend

🟦 *Hardware Configuration*

🟥 *Software Configuration*

API (Mistral) / Undercloud

tripleo-validations

Heat

OpenStack Resources (Nova/Ironic/Neutron)

Ansible Playbooks

Node(s)

Node(s)

Node(s)

config download

stack-outputs

playbooks saved on undercloud

ansible-playbook

server deployment(s)

pre deployment(s)

step1 deployment

step2 deployment

step3 deployment

step4 deployment

step5 deployment

post deployment(s)

ansible deployment results

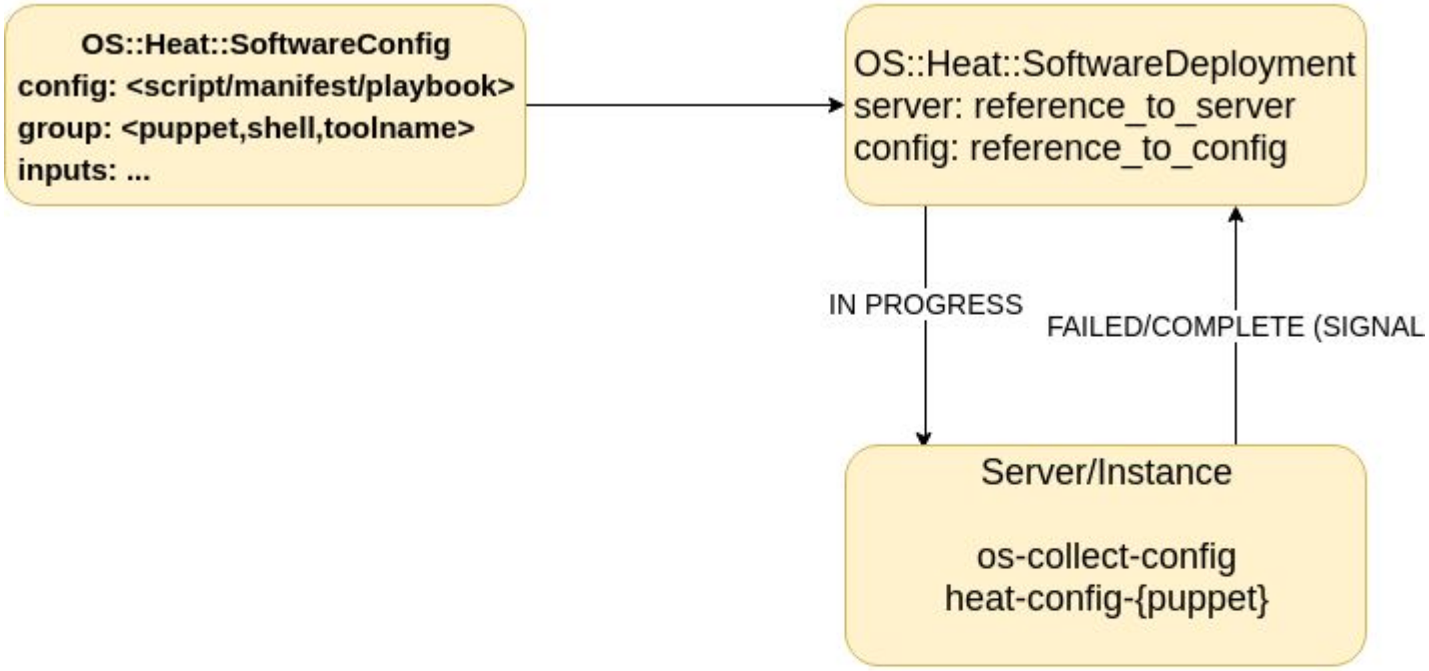
Deployment status

```
(undercloud) [stack@undercloud-0 ~]$ openstack stack list
+--------------------------------------+------------+-----------------+
| ID                                   | Stack Name | Stack Status    |
+--------------------------------------+------------+-----------------+
| 453b400a-0249-45ae-9fb1-a8724dbf3439 | overcloud  | UPDATE_COMPLETE |
+--------------------------------------+------------+-----------------+
```

```
(undercloud) [stack@undercloud-0 ]$ openstack overcloud status
+-----------+---------------------+---------------------+-------------------+
| Plan Name |       Created       |       Updated       | Deployment Status |
+-----------+---------------------+---------------------+-------------------+
| overcloud | 2019-06-11 21:19:01 | 2019-06-11 21:19:01 |   DEPLOY_SUCCESS  |
+-----------+---------------------+---------------------+-------------------+
```

Evolution - Software Configuration

Before Queens:



OS::Heat::SoftwareConfig
config: <script/manifest/playbook>
group: <puppet,shell,toolname>
inputs: ...

OS::Heat::SoftwareDeployment
server: reference_to_server
config: reference_to_config

IN PROGRESS

FAILED/COMPLETE (SIGNAL

Server/Instance

os-collect-config
heat-config-{puppet}

Config-download - how it is enabled

```
OS::TripleO::SoftwareDeployment: ../config-download-structured.yaml

OS::Heat::SoftwareDeployment: ../config-download-software.yaml
OS::Heat::StructuredDeployment: ../config-download-structured.yaml

OS::TripleO::DeploymentSteps: OS::Heat::None
OS::TripleO::WorkflowSteps: OS::Heat::None

OS::TripleO::Ssh::KnownHostsDeployment: OS::Heat::None
OS::TripleO::Ssh::HostPubKey: OS::Heat::None
```

Config-download - Mistral

```
# tripleo-common/workbooks/deployment.yaml
 config_download_deploy:

   description: >
     Configure the overcloud with config-download.

   input:
     - queue_name: tripleo
     - plan_name: overcloud
     - ssh_network: ctlplane
     - work_dir: /var/lib/mistral
     - verbosity: 1
     - blacklist: []
     - override_ansible_cfg: ""
     - config_download_timeout: 14400
```

Config-download - Mistral

```
# tripleo-common/workbooks/deployment.yaml
...
    download_config:
     action: tripleo.config.download_config
     input:
       work_dir: <% $.get('work_dir') %>/<% $.get('plan_name') %>
       container_config: <% $.get('plan_name') %>-config
...
```

Config-download - Mistral

```
# tripleo-common/workbooks/deployment.yaml
...
    generate_inventory:
      action: tripleo.ansible-generate-inventory
      input:
        ansible_ssh_user: tripleo-admin
        ansible_python_interpreter: <% $.get('deployment_options',
{}).get('ansible_python_interpreter', null) %>
        work_dir: <% $.get('work_dir') %>/<% $.get('plan_name') %>
        plan_name: <% $.get('plan_name') %>
        ssh_network: <% $.get('ssh_network') %>
        undercloud_key_file: <% $.get('work_dir') %>/.ssh/tripleo-admin-rsa
```

Config-download directory structure

```
(undercloud) [stack@undercloud-0 overcloud]$ cd
/var/lib/mistral/overcloud
(undercloud) [stack@undercloud-0 overcloud]$ tree
[...]
├── ansible.cfg
├── ansible.log
├── ansible-playbook-command.sh
├── BlockStorage
├── CephStorage
├── deploy_steps_playbook.yaml
├── fast_forward_upgrade_playbook.yaml
├── group_vars
├── overcloud-config.tar.gz
├── templates
├── tripleo-ansible-inventory.yaml
```

Config-download - ansible cmd

```
(undercloud) [stack@undercloud-0 overcloud]$ sudo cat ansible-playbook-command.sh
#!/bin/bash

PROFILE_TASKS_TASK_OUTPUT_LIMIT="0"
OS_USERNAME="admin"
ANSIBLE_CONFIG="/var/lib/mistral/overcloud/ansible.cfg"
OS_AUTH_URL="http://192.168.24.3:5000/v3"
ANSIBLE_LOCAL_TEMP="/var/lib/mistral/overcloud"
OS_PROJECT_NAME="admin"
HOME="/var/lib/mistral/overcloud"
ANSIBLE_CALLBACK_WHITELIST="profile_tasks"

timeout 4387 ansible-playbook -v
/var/lib/mistral/overcloud/deploy_steps_playbook.yaml --become --inventory-file
/var/lib/mistral/overcloud/tripleo-ansible-inventory.yaml "$@"
```

Config-download directory

```
(undercloud) [stack@undercloud-0 overcloud]$ git log
commit 7705583e881c458b3b77c2af89e3d05d096ad44e
Author: mistral <mistral@undercloud-0.redhat.local>
Date:   Tue Jun 11 17:20:23 2019 -0400

    Automatic commit by Mistral GetOvercloudConfig action.

    User: admin
    Project: admin

commit 145912471ed2f29b55eefe3f3e9a54d58f7bd1d1
Author: mistral <mistral@undercloud-0.redhat.local>
Date:   Sun Jun 9 19:46:28 2019 -0400

    Automatic commit by Mistral GetOvercloudConfig action.

    User: admin
    Project: admin
```

Config-download -- flags

```
--overcloud-ssh-user   # Initial ssh user used for creating tripleo-admin.
                       # Defaults to heat-admin

--overcloud-ssh-key    # Initial ssh private key (file path) to be used for
                       # creating tripleo-admin.
                       # Defaults to ~/.ssh/id_rsa

--override-ansible-cfg # path to an ansible config file, to inject any
                       # arbitrary ansible config to be used when running
                       # ansible-playbook

--stack-only           # Only update the plan and stack. Skips applying the
                       # software configuration with ansible-playbook.

openstack overcloud config download \
  --name overcloud \
  --config-dir config-download

--config-download-only # Only apply the software configuration with
                       # ansible-playbook. Skips the stack and plan update.
```

Config-download - benefits

- Clear separation of responsibilities (Heat - infrastructure, Puppet - configuration management, Ansible - drives software configuration)
- Faster Heat stack creation
- Easier troubleshooting
- More flexibility and fine-grained control
  - `./ansible-playbook-command.sh --syntax-check|--check`
  - `./ansible-playbook-command.sh --limit Controller|Compute`
  - `./ansible-playbook-command.sh --limit compute-0`
  - `./ansible-playbook-command.sh --list-tags`

```
 play #7 (Controller:overcloud): Host prep steps       TAGS: [host_prep_steps,overcloud]
      TASK TAGS: [host_prep_steps, overcloud]
  play #8 (Undercloud): External deployment step 1      TAGS:
[external_deploy_steps,external,step1]
      TASK TAGS: [container_image_prepare, external, external_deploy_steps, step1]
```

  - `./ansible-playbook-command.sh --skip-tags|--tags TAG --limit compute-0`

- Easier integration and customizations

Cool stuff and references

[1] TripleO documentation - https://docs.openstack.org/tripleo-docs/latest

[2] Steven Hardy blog - http://hardysteven.blogspot.com/

- TripleO Containerized deployments, debugging basics -
- Debugging TripleO revisited - Heat, Ansible & Puppet

[3] James Slage blog - https://blogslagle.wordpress.com

- Config-download with Ansible Tower

[4] http://jaormx.github.io

- TripleO service deployment steps
- Dissecting TripleO service templates series

[5] T-H-T Readme files

```
$ cd tripleo-heat-templates
$ find . -type f -name "README*"
./deployment/README.rst
./deployment/deprecated/README.rst
./deployment/experimental/README.rst
./roles/README.rst
./deployed-server/README.rst
./README.rst
./network/config/2-linux-bonds-vlans/README.md
./network/config/single-nic-vlans/README.md
```

Thank you!

IRC: pkopec on freenode.net
Slides: https://github.com/piotrkpc/OpenInfraDaysPL_2019