Ansible: use-cases

Marica Antonacci - INFN BARI
marica.antonacci@ba.infn.it

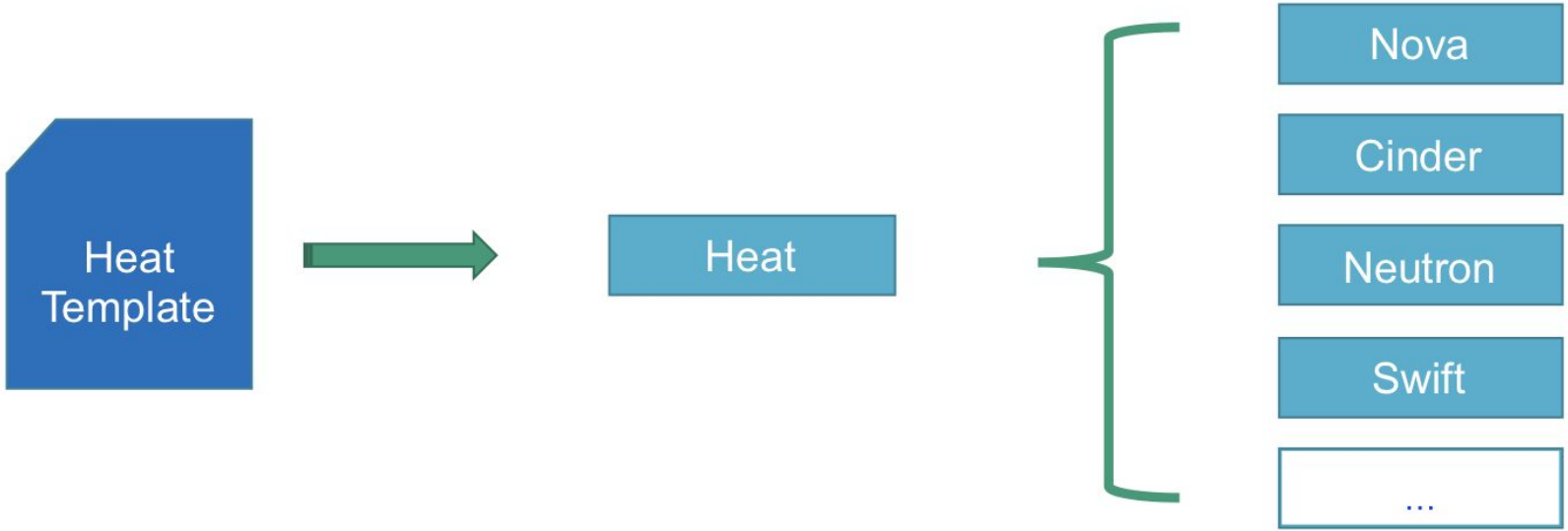
Corso Ansible/Foreman/Puppet
5-8 June 2018, INFN BARI

Outline

From INDIGO-DataCloud experience:

- Ansible and Heat
- Ansible and TOSCA
- Ansible and Docker

HEAT and HOT templates



Software Configuration

There are two main ways for running SW configuration scripts in VMs:

- User-data + cloudinit
  - Run once after instance first boot
- Software Deployment resources
  - Run on every stack create/update
  - Send a signal back to Heat when finished
  - You can define dependencies among different scripts
  - Requires special services (hooks) running in the VM

An example

```
nginx_config:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ansible
    config: |
      ---
      - name: Install and run Nginx
        connection: local
        hosts: localhost
        tasks:
         - name: Install Nginx
           apt: pkg=nginx state=installed update_cache=true
           notify:
            - Start Nginx
        handlers:
         - name: Start Nginx
           service: name=nginx state=started
```

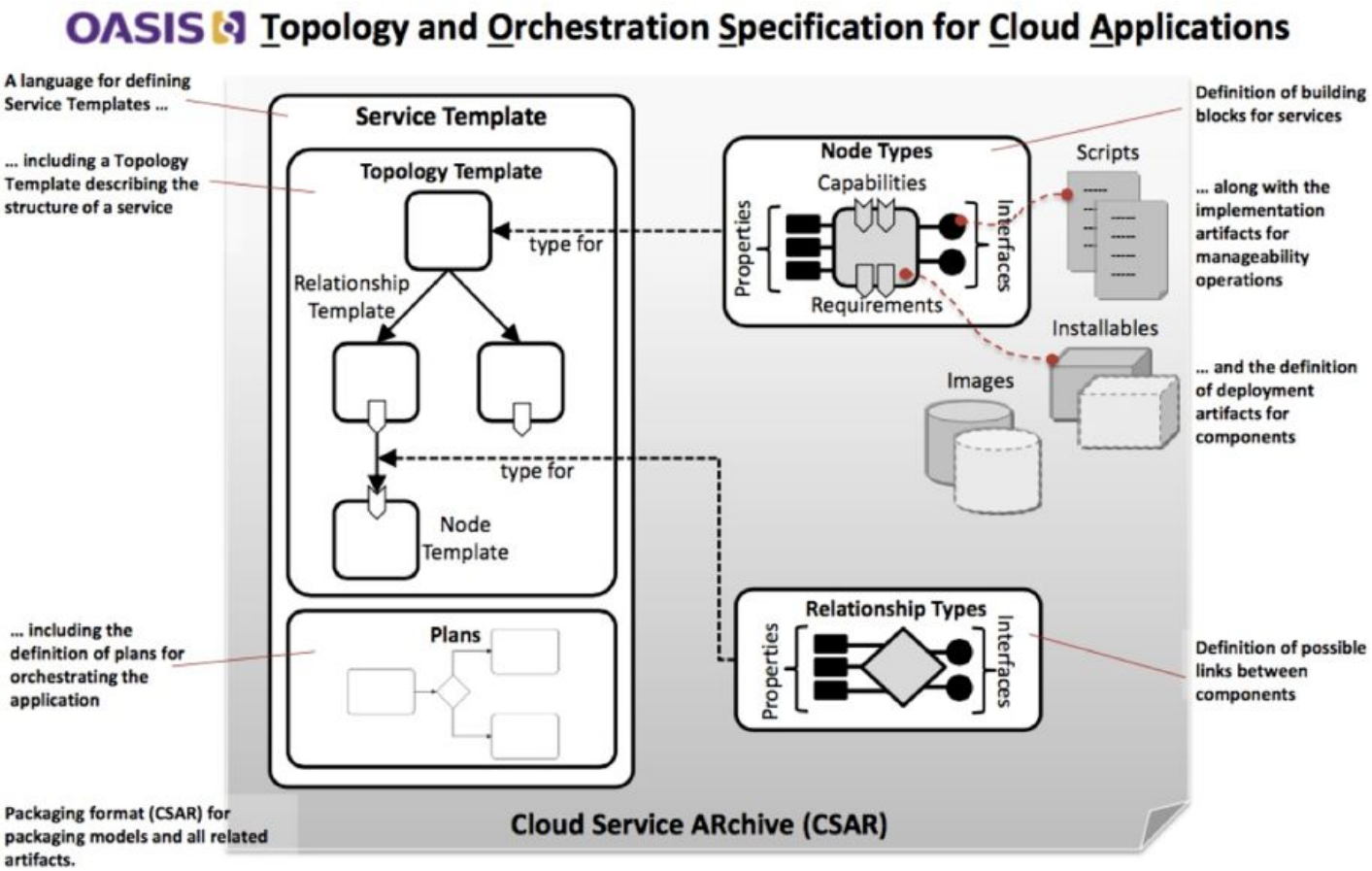
```
deploy_nginx:
  type: OS::Heat::SoftwareDeployment
  properties:
    signal_transport: TEMP_URL_SIGNAL
    config:
      get_resource: nginx_config
    server:
      get_resource: server
```

INDIGO Mesos Templates

Let's have a look at

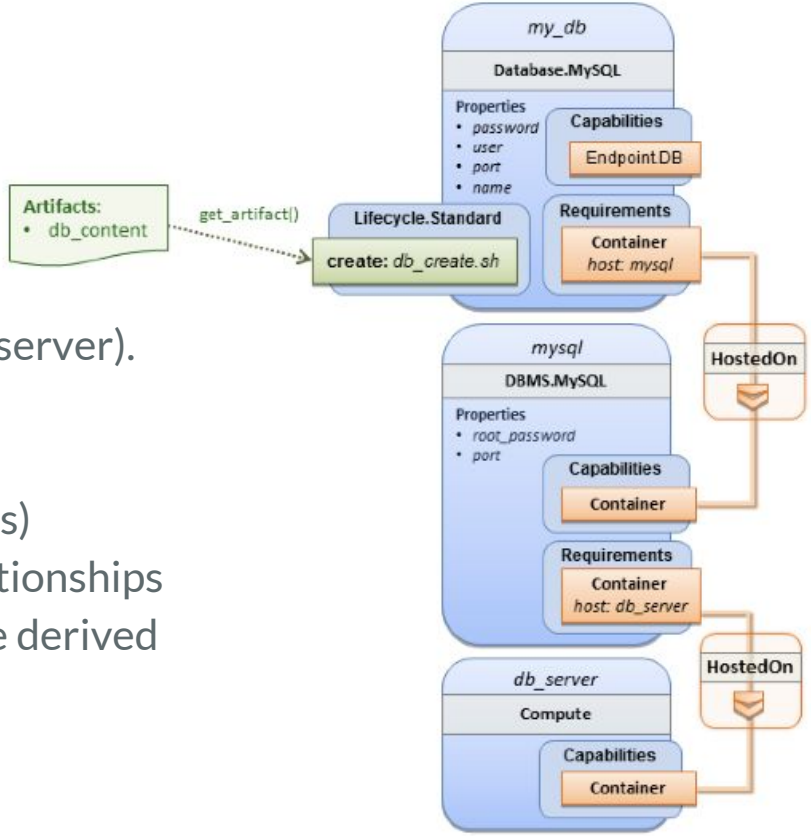
https://github.com/indigo-dc/mesos-cluster/tree/master/deploy/openstack-heat

TOSCA


OASIS Topology and Orchestration Specification for Cloud Applications
A language for defining Service Templates ...
... including a Topology Template describing the structure of a service
... including the definition of plans for orchestrating the application
Packaging format (CSAR) for packaging models and all related artifacts.
Service Template
Topology Template
type for
Relationship Template
type for
Node Template
Plans
Node Types
Capabilities
Properties
Interfaces
Requirements
Scripts
Installables
Images
Relationship Types
Properties
Interfaces
Definition of building blocks for services
... along with the implementation artifacts for manageability operations
... and the definition of deployment artifacts for components
Definition of possible links between components
Cloud Service ARchive (CSAR)

TOSCA Topology

- Components in the topology are called Nodes
  - Each Node has a Type (e.g. Host, BD, Web server).
  - The Type is abstract and hence portable
- The Type defines Properties and Interfaces
- An Interface is a set of hooks (named Operations)
- Nodes are connected to one another using Relationships
- Both Node Types and Relationship Types can be derived



TOSCA types

- Normative types
- Custom types

- INDIGO custom types: https://github.com/indigo-dc/tosca-types/blob/master/custom_types.yaml: new types have been defined for elastic clusters, Marathon applications, Chronos jobs, etc.

The artifacts are ansible playbooks that use indigo-dc ansible roles

INDIGO ansible roles

```
ansible-galaxy install
indigo-dc.<role-name>
```

GALAXY

ABOUT | EXPLORE | BROWSE ROLES | BROWSE AUTHORS | SIGN IN

BROWSE AUTHORS

indigo-dc

INDIGO - DataCloud

Followers: 0

Search roles

Role	Description
ambertools	Ambertools applications
calico	Configure and run calico
chronos	Install Chronos
clues	Install Clues
consul	Install consul with dnsmasq
disvis-powerfit	disvis and powerfit applications
docker	Install docker engine
etcd	Configure and run dockerized etcd
galaxy	Install Galaxy portal
hadoop	Install a Hadoop Cluster
haproxy-consul	Install haproxy-consul
keepalived	Install keepalived
marathon	Install Marathon
mesos	Install Mesos components
nfs	NFS server/client
slurm	Install SLURM cluster
zabbix-agent	Install Zabbix agent in active mode
zookeeper	Install zookeeper

INDIGO TOSCA Templates

Let's have a look at

https://github.com/indigo-dc/tosca-templates

Ansible and docker

Running ansible playbooks in the Dockerfile:

```
RUN apt-get update -y
RUN apt-get install software-properties-common -y
RUN apt-add-repository ppa:ansible/ansible
RUN apt-get update && \
    apt-get install -y ansible && \
    rm -rf /var/lib/apt/lists/*
RUN ansible-galaxy install indigo-dc.oneclient && \
    ansible-playbook
/etc/ansible/roles/indigo-dc.oneclient/tests/test.yml
```

The same ansible recipes can be used for configuring bare-metal, cloud servers and containers..

Managing docker containers with ansible

Ansible provides some modules to manage containers:

docker_service: Consumes docker compose to start, shutdown and scale services

docker_container: Manage the life cycle of docker containers

docker_image: Build, load or pull an image, making the image available for creating containers. Also supports tagging an image into a repository and archiving an image to a .tar file.

docker_image_facts: inspect images, returning an array of inspection result

docker_login: Authenticate with a docker registry and add the credentials to your local Docker config file

docker_volume: Create/remove Docker volumes

ansible-container (NEW): a tool for building Docker images and orchestrating containers using Ansible playbooks