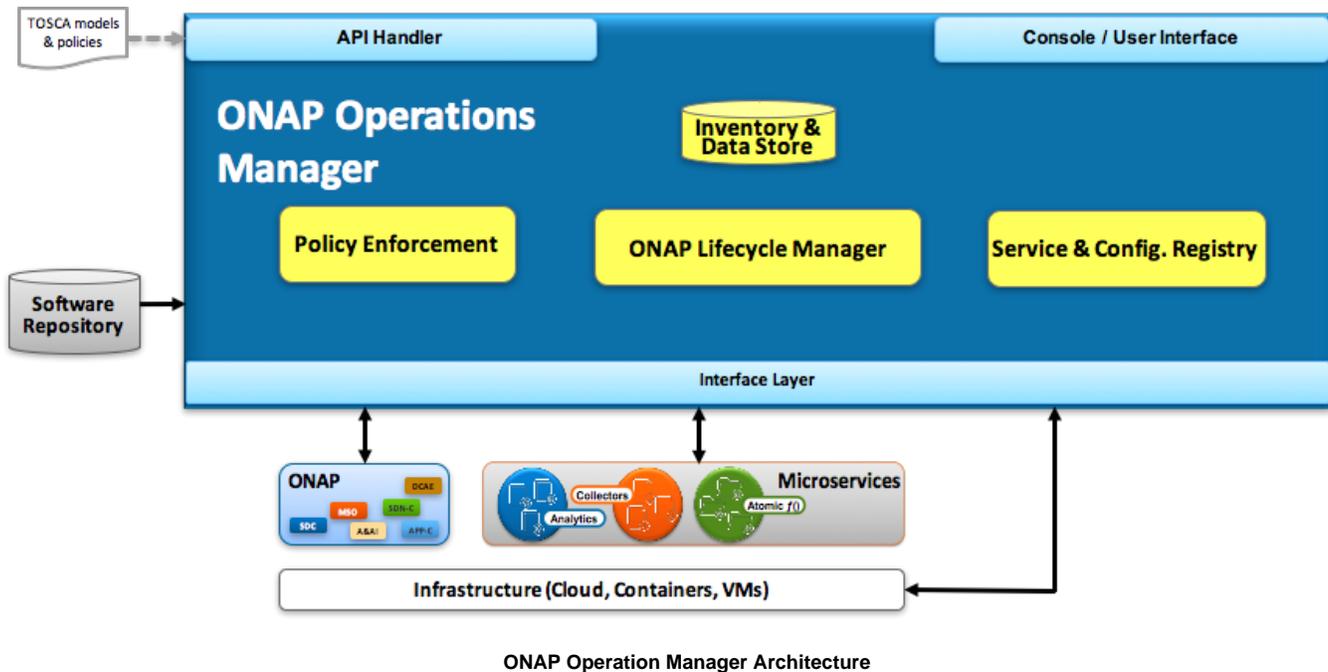


OOM with TOSCA and Cloudify

ONAP Operations Manager is an ONAP tool to efficiently Deploy, Manage, Operate the ONAP platform and its components (e.g. MSO, DCAE, SDC, etc.) and infrastructure (VMs, Containers).

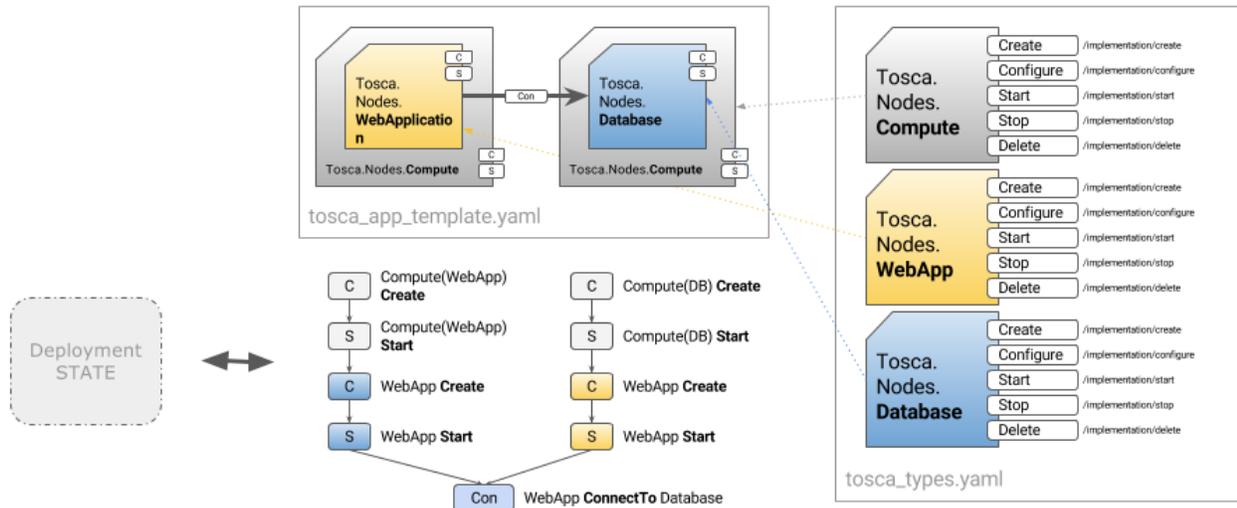
With OOM, service providers will have a single dashboard/UI to deploy & un-deploy the entire (or partial) ONAP platform, view the different instances being managed and the state of each component, monitor actions that have been taken as part of a control loop (e.g., scale in-out, self-heal), and trigger other control actions like capacity augments across data centers.



WHAT IS a TOSCA Model-driven Orchestration

Model driven orchestration allows modeling of services and applications using a declarative and descriptive language, then to operate the application using the model. This includes deployment of the application, healing failures, scaling and others. Instead of executing manually crafted workflows, the orchestrator creates dynamic execution workflow graph, based on the application model. Each node of the graph triggers a corresponding lifecycle operation execution, while the orchestrator keeps track and stores state in a context object allowing passing information from one operation to another.

Declarative Model-Driven Orchestration



- Define reusable TOSCA node types. Each type can declare the interfaces for its lifecycle operations.
- Create a TOSCA template YAML. The template describes the components, services and all of their dependencies and relationships.
- Cloudify Orchestrator creates a deployment graph, at run time. Executing the Install workflow traverses thru the graph, and executes relevant lifecycle operation interfaces.

Cloudify Support for OOM

ONAP uses containers and Kubernetes as the platform to run and manage an ONAP cluster.

Cloudify [support for Kubernetes](#) leverages cloudify TOSCA based multi-cloud infrastructure orchestration as well as for service orchestration to automate the deployment and management of ONAP on multi-cloud and real life heterogeneous environments that is comprised out of existing network services and other service elements that doesn't fit into a single Kubernetes platform.

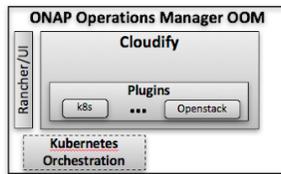
[Cloudify integration with ONAP](#) leverages Cloudify kubernetes blueprint to address the following aspects in OOM:

Bootstrapping and Managing Kubernetes Cluster on Multi Cloud

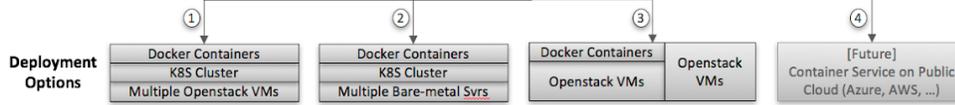
- Multi Cloud Provisioning - Cloudify sets the network and compute resources needed to run Kubernetes on bare metal, OpenStack, VMware, AWS, Google Cloud and Azure out of the box.
- Monitoring - Cloudify monitors the compute resources for kubernetes and allow built-in self-healing and scaling of the kubernetes instances.
- Self Healing and Scaling - Cloudify provides built-in support to allow self-healing and scaling of Kubernetes compute resources in case of a failure.

Deployment Options:

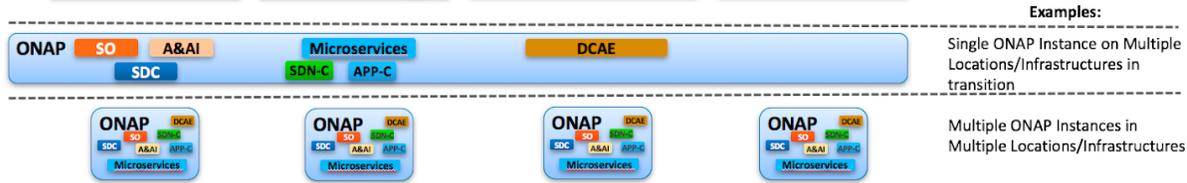
1. K8S on Openstack VMs
2. K8S on bare-metal servers
3. Openstack VMs or Docker on Openstack
4. Public Cloud Services
5. Future container orchestration??



- Supports a hybrid deployment thru plugin* technology (openstack, k8s, docker, azure, ...)
- Model-Driven, TOSCA blueprints**
- Interfaces with multiple container orchestration technologies (initial: kubernetes; future: cloud native)



* Might need synergy with multi-VIM Project
 ** SDC will design blueprints (post-R1)



Examples:

Single ONAP Instance on Multiple Locations/Infrastructures in transition

Multiple ONAP Instances in Multiple Locations/Infrastructures

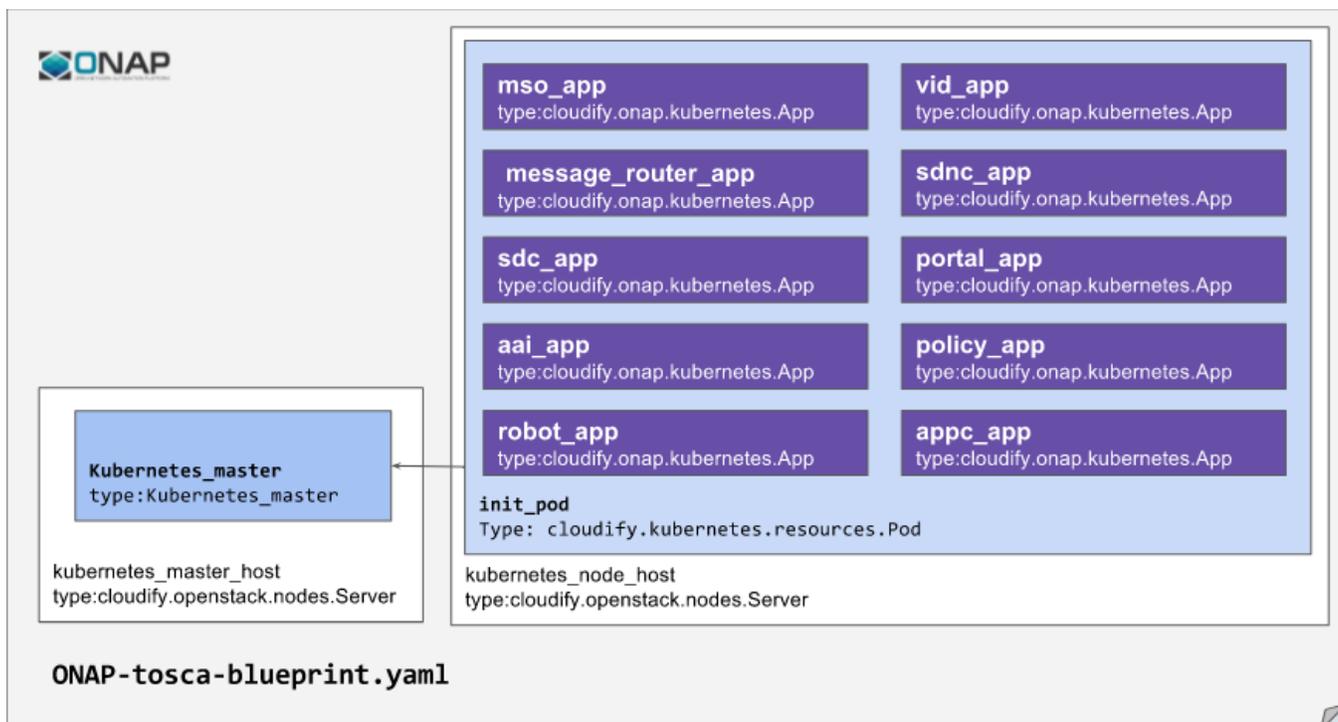
Cloudify integration with ONAP Operation Manager (OOM)

Service Orchestration of Kubernetes Micro Services across Hybrid Stack

Cloudify [Kubernetes Plugin](#) provides a mean to model the relationship of existing Kubernetes micro-services and connect them with other services such as existing network services or other services that runs outside of ONAP.

Cloudify's Kubernetes Roadmap includes a Cloudify Provider for Kubernetes that will take this role of infrastructure provisioner. This enables users to create custom implementations for underlying routing, storage, and compute services. For example, we can define a pod with a load balancer in Openstack that does not rely on Openstack floating IP, but instead on a virtual IP based on another vendor VNF..

The example below shows a TOSCA blueprint which sets the relationship between all the relevant ONAP micro-services.



TOSCA Blueprint example

Multi Site Management for ONAP

Cloudify can run multiple instances of Kubernetes/ONAP from the same manager.

This is useful when setting Testing, QA, Production environment but its also useful when setting up Kubernetes cluster on multi-site deployment or even on edge devices.

Supporting other containers and non container based platforms

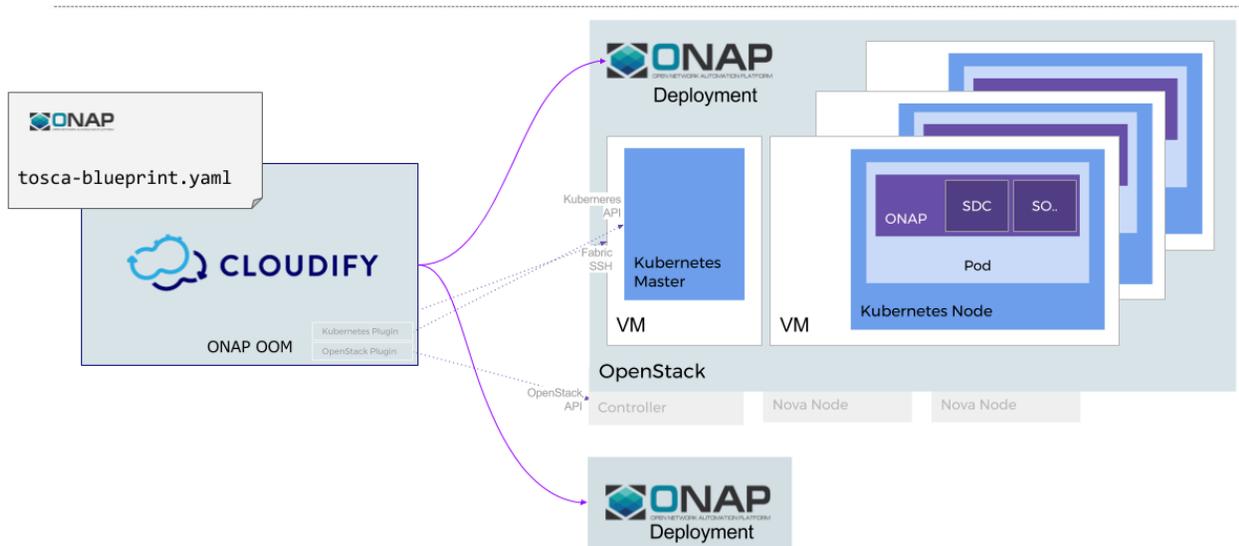
The container platform landscape is changing rapidly and include other alternatives to Kubernetes such as Docker Swarm as well as non container based platform.

Cloudify plugin approach provides an abstraction layer that decouples the actual service from the platform in which it will be running. This gives the flexibility for Operators to choose the platform of choice for their service as well as mix and match between services that runs on Kubernetes and Docker Swarm under the same automation flow.

ONAP Architecture with Kuberenetes and OpenStack

TOSCA Base ONAP OOM Release 1 is deployed in Kubrenetes PODs, that are run on OpenStack VM.

CLLOUDIFY OOM ARCHITECTURE



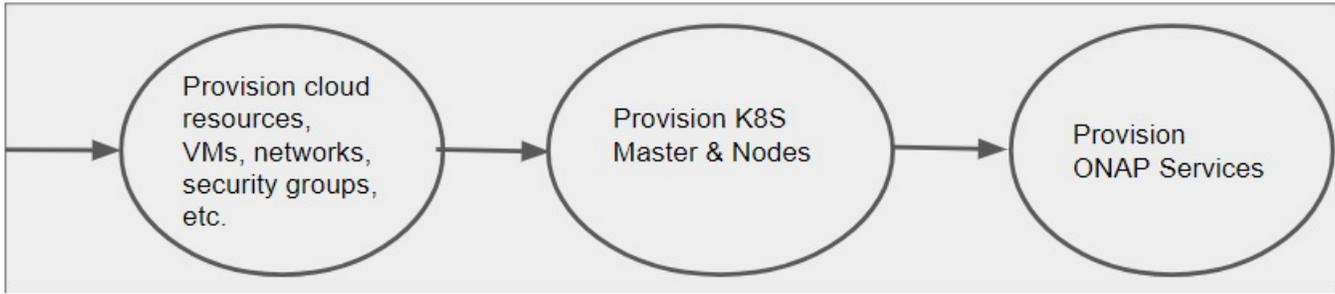
ONAP OOM TOSCA Models

TOSCA to model and orchestrate all ONAP services, in release 1 we lege, with all of their components, and then

[TOSCA Model to deploy Kuberenetes on OpenStack](#) (click on the link to view the blueprint)

[TOSCA Model to deploy ONAP on Kuberenetes](#) (click on the link to view the blueprint)

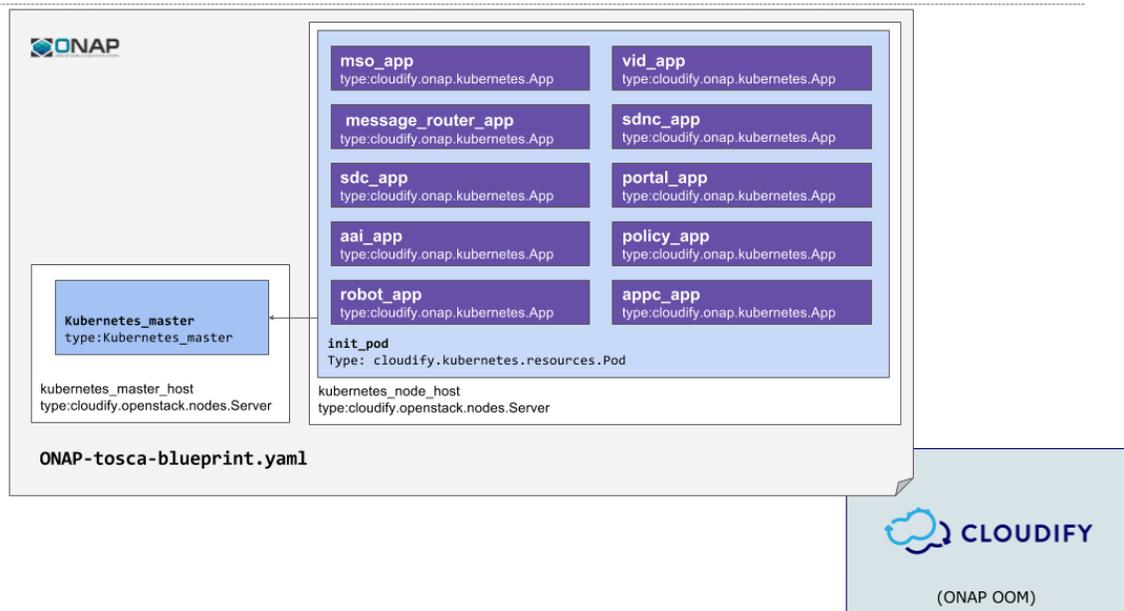
How To Deploy ONAP on Kubernetes using Cloudify



C

1. Create a deployment using the ONAP OOM TOSCA Blueprint
2. Provide user Inputs with OpenStack Credentials, Kubernetes credentials, etc'
3. Execute the Install workflow

CLOUDIFY OOM ARCHITECTURE



JIRA Track Progress

[OOM-46](#) - Platform infrastructure deployment with TOSCA IN PROGRESS

Wiki Links

[ONAP on Kubernetes on Cloudify](#) - ongoing experiments bringing up the system

OOM Cloudify Release 1 - Amsterdam

1. Cloudify Deploys Kubernetes on OpenStack VMS (TOSCA blueprint)
2. The second TOSCA blueprint deploys the ONAP services (see figure below) on Kubernetes
 - a. Cloudify creates a POD
 - b. Cloudify Deploys ONAP on POD (TOSCA blueprint)

1. Use TOSCA to provision OpenStack VMs
2. Use TOSCA to deploy Kubernetes
3. Use TOSCA to deploy ONAP services on Kubernetes
 - a. SO

- b. VID
- c. Message_Router
- d. SDNC
- e. SDC
- f. Portal
- g. AAI
- h. Policy
- i. Robot
- j. Appc

4. Support multiple deployments of ONAP(Test,QA, Prod) from same blueprint

OOM Cloudify Roadmap

Create reusable TOSCA node type for each ONAP service

ONAP dependencies expressed in TOSCA

Accept custom user inputs for ONAP services' configuration

Test ONAP on bare metal with TOSCA

Model ONAP inner service dependencies.

ONAP services auto heal, and auto scale

ONAP services placement

Configurable Hybrid deployment (some of ONAP services on cloud, and others on bare-metal/or other clouds)