

CDS Integration with K8S workloads

Konrad Bańka (Samsung)
Łukasz Rajewski (Orange)

14.01.2020

Agenda

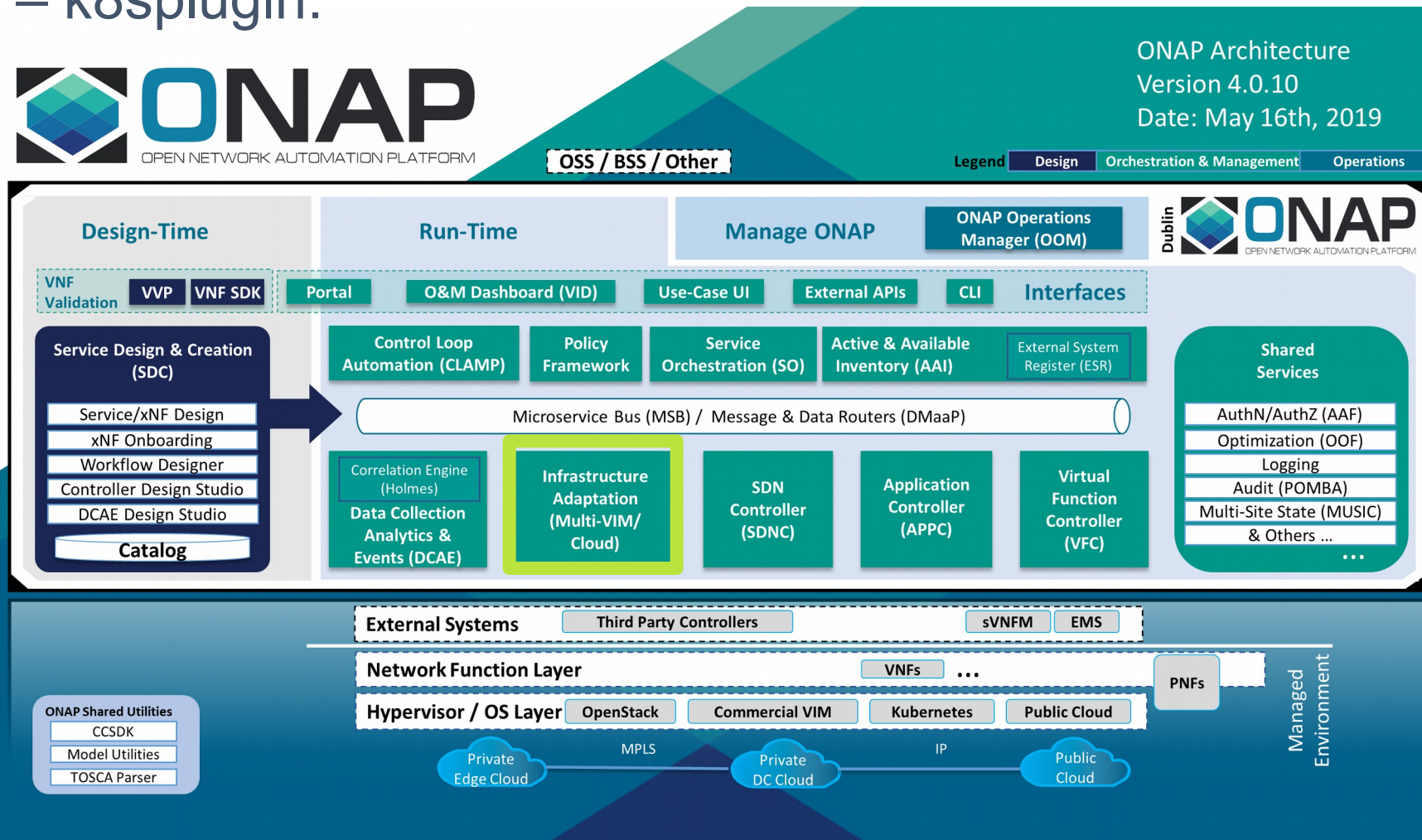
- Introduction
- CNF workflow in El Alto
- Changes for Frankfurt: CDS integration for CNFs
- Future Plans/Possibilities (for CNFs/vFW use-case)
- vFW use-case
- Discussion

Agenda

- Introduction
- CNF workflow in El Alto
- Changes for Frankfurt: CDS integration for CNFs
- Future Plans/Possibilities (for CNFs/vFW use-case)
- vFW use-case
- Discussion

Introduction

For CNFs lifecycle operations, we're using Multicloud/k8s project with its main part – k8splugin.



K8s Plugin Role

- Upload of CNF definition – Helm Chart
- Customization of deployment
- Instantiation of CNF
- Configuration of CNF (Day 2 Configuration)
- Configuration of connectivity ONAP - K8S

Introduction

Resource Bundle (RB)

- *Resource Bundle* is k8splugin's object corresponding to vf-module resource. *Resource Bundle* contains Helm Chart as a definition of CNF.
- *Resource Bundle* is provided as "CLOUD_TECHNOLOGY_SPECIFIC_ARTIFACT" part of CSAR and is distributed automatically from SDC. It comes with dummy heat file that creates vf-module

RB Profile

- *Profile* is a Day 0 configuration object containing override data for Helm Chart. It allows customization of *Resource Bundle* with *Profile* content at the time of vf-module instantiation.

RB Instantiation

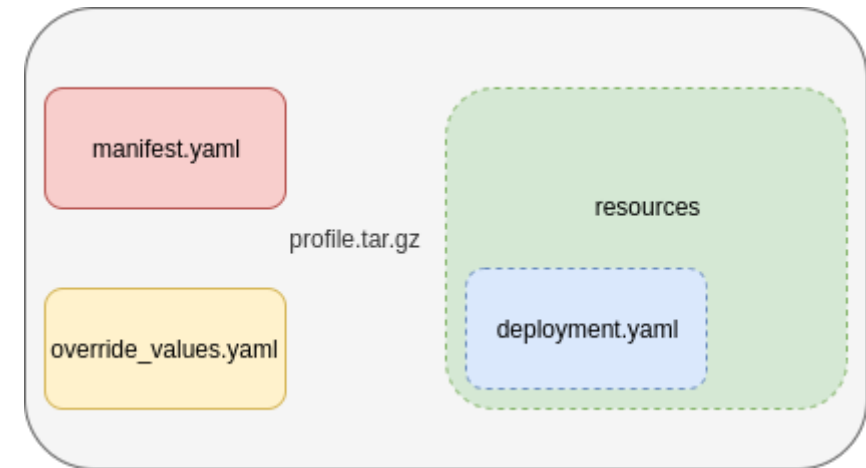
- In order to launch CNF instance, user needs to provide in VID both *Resource Bundle* and selected *Profile* to be used for customization.

Introduction

Example of RB Profile's manifest

```
---  
version: v1  
type:  
  values: "override_values.yaml"  
  configsource:  
    - filepath: resources/deployment.yaml  
      chartpath: templates/deployment.yaml
```

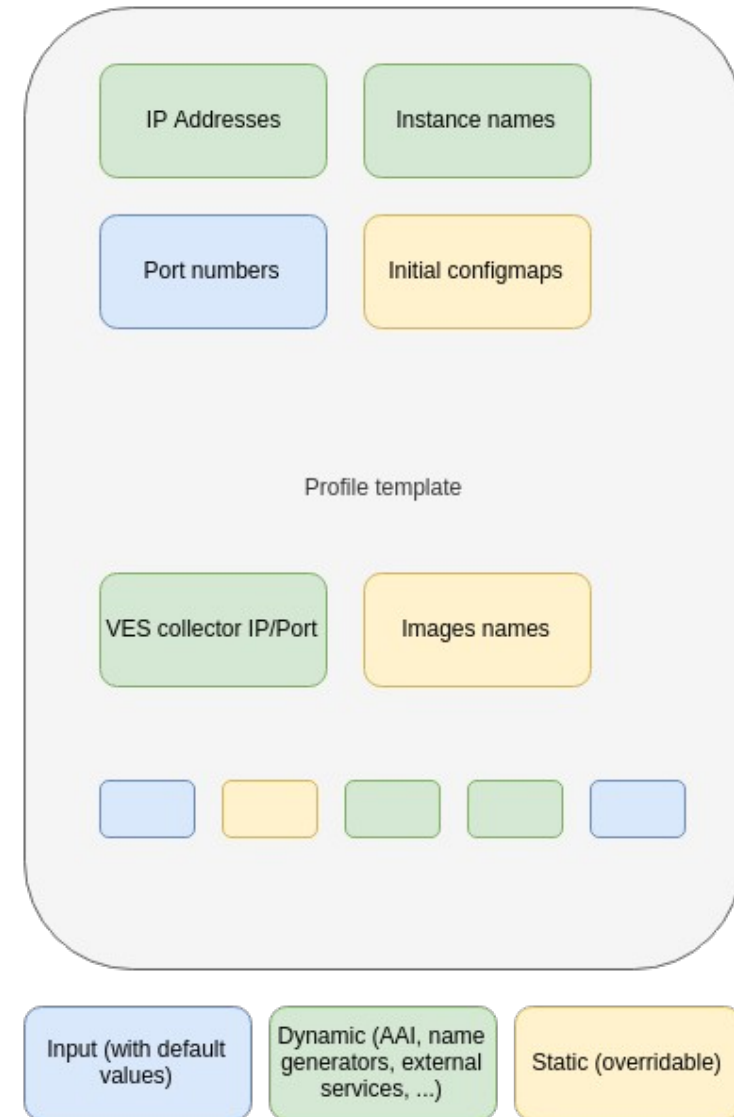
Example of profile's structure



Introduction

Example overrides

```
replicaCount: 1
image:
  repository: virtlet.cloud/ubuntu/16.04
  tag: latest
pullPolicy: IfNotPresent
resources:
  limits:
    memory: 4Gi
service:
  type: NodePort
  ports:
    port: 2831
    nodePort: 30831
vpg_name_0: test-vnf-vpg
vnf_name: test-vnf
dcae_collector_ip: 10.0.4.1
dcae_collector_port: 8080
demo_artifacts_version: 1.5.0
int_private1_net_id: unprotected-net
onap_private_net_id: onap-private-net
vpg_int_private1_ip_0: 192.168.10.2
vpg_onap_private_ip_0: 10.10.100.2
vfw_int_private1_ip_0: 192.168.10.3
vs_n_int_private2_ip_0: 192.168.20.3
int_private2_net_cidr: 192.168.20.0/24
```



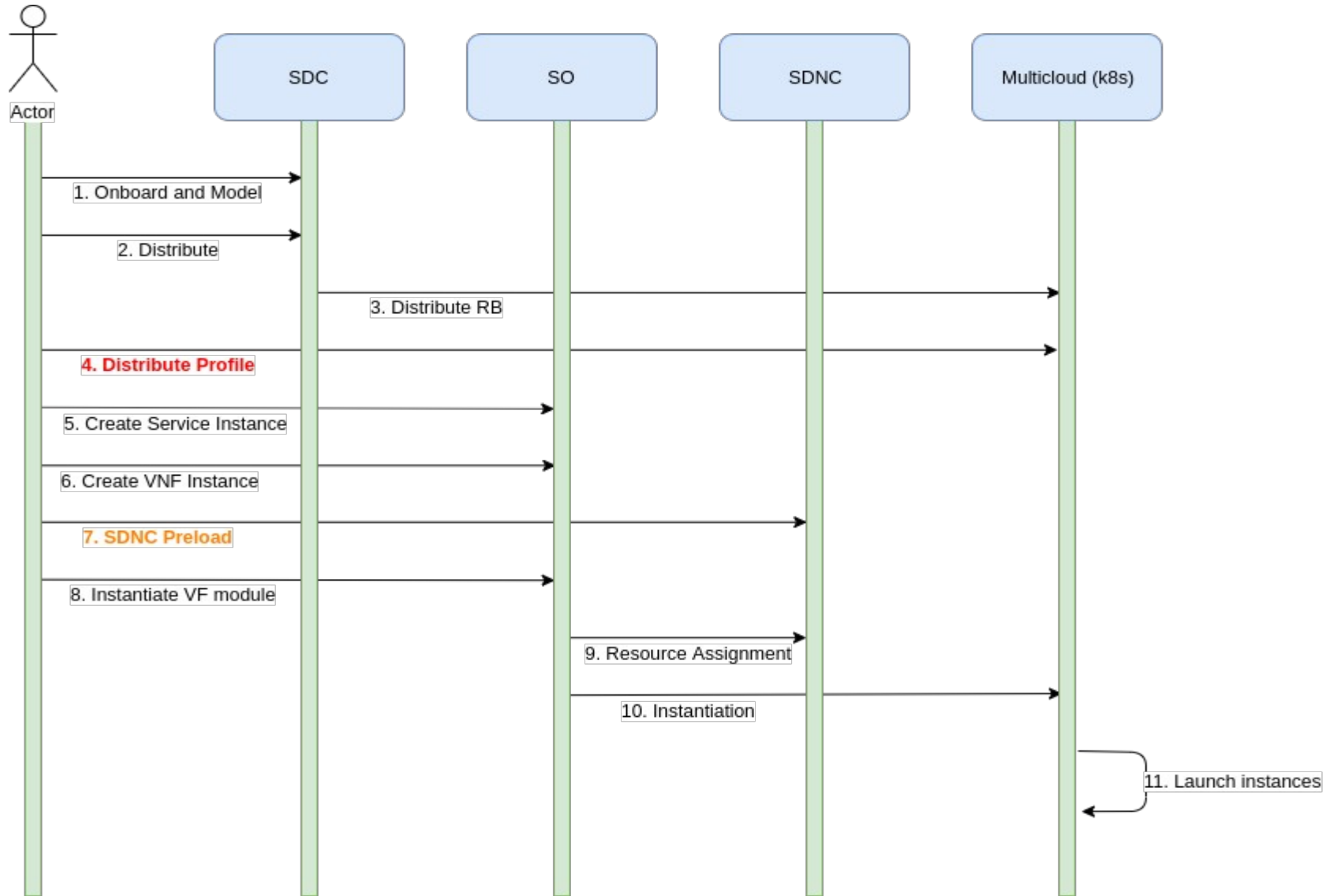
Agenda

- Introduction
- **CNF workflow in El Alto**
- Changes for Frankfurt: CDS integration for CNFs
- Future Plans/Possibilities (for CNFs/vFW use-case)
- vFW use-case
- Discussion

CNF workflow in E1 Alto

- Service instantiated with a'la carte SO workflow and VNF API
- **Manual k8s Resource Bundle's Profile upload**
- **SDNC preload required**
 - Lack of SDNC Directives as dummy heat has no input parameters
 - User Directives from additional file in instantiation from VID.
- User Directives needed: RB Name, RB Version, K8S Namespace and RB Profile Name
- Resource Bundle's name and version require their manual resolution

CNF workflow in E1 Alto



Agenda

- Introduction
- CNF workflow in El Alto
- **Changes for Frankfurt: CDS integration for CNFs**
- Future Plans/Possibilities (for CNFs/vFW use-case)
- vFW use-case
- Discussion

Changes for Frankfurt: CDS integration for CNFs

REQ-182, INT-1184 - Wiki: Integration with CDS

- Utilization of SO Macro workflow
 - Benefits from CDS and opens new opportunities for CNFs in ONAP in the future
 - Reduces instantiation steps from network operator's perspective
- Resource Bundle names / version based on *vf-module-model-invariant-id / vf-module-version-id*
 - Identifiers from VF module model, used to create RB Definition when model is distributed
 - RB Name and Version are not required in the SO instantiation flow, as they are being always sent by SO to multcloud in time of vf-module instantiation
- Parameters from User Directives moved to SDNC Directives
 - When VNF API is used only one parameter is required: k8s-rb-profile-name
 - In Macro flow with CDS additionally k8s-rb-profile-namespace parameter can be used

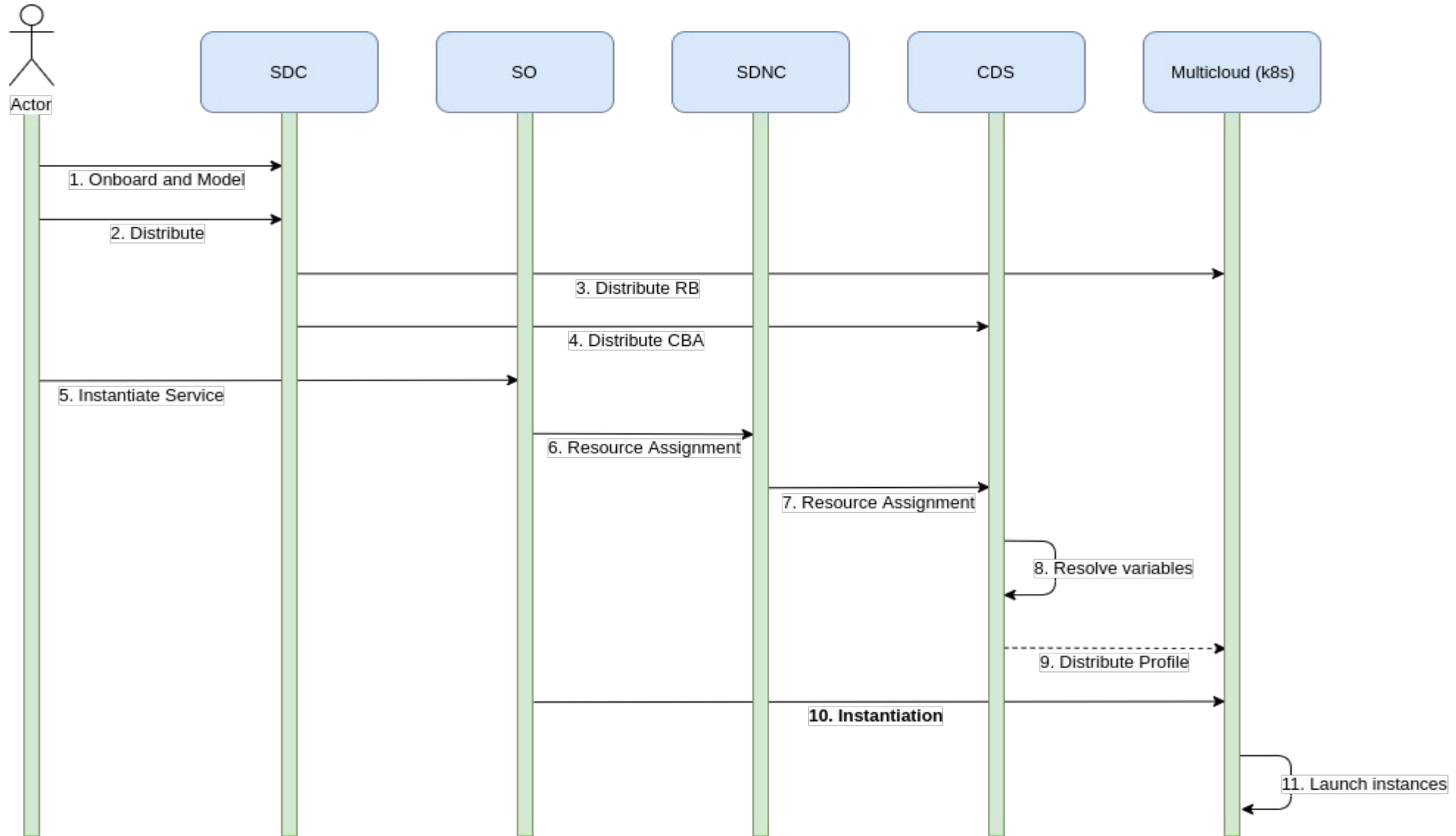
Changes for Frankfurt: CDS integration for CNFs

- CDS used for provisioning of variables
 - Use of CDS allows user to simplify instantiation process by discarding the need of SDNC preload, but still giving possibility to provide per-instance overrides
 - Dynamic variables provisioning may simplify future LCM operations implemented for CNFs
- CDS uploads RB profile as a part of resource assignment workflow
 - Profile can be uploaded for each vf-module
 - Profiles may be embedded in the CBA archive
 - Depending on need, profile can be uploaded by CDS if specifically designed in CBA model
 - Profile in CDS can be templated thus allowing for dynamic content defined there
 - Exemplary implementation for vFW CNF Use Case as additional workflow step in Resource Assignment phase.

Changes for Frankfurt: CDS integration for CNFs

- Many Helm Charts in one CSAR and VNF
 - It gives more design options. We can decouple Helm Charts into many vf-modules in order to manage them independently in ONAP
 - Utilized in vFW CNF Use Case to make vFW CNF CDS CBA similar to standard vFW CBA
- Implemented default profile so it is no longer mandatory resource for RB instantiation
 - If application doesn't demand it, it is possible to totally omit profile to instantiate CNF thus simplify process for simple CNFs
 - Combined with previous feature, it enables providing more static parameters (like DCAE IP/port) within profile with dynamic ones transferred with instantiation REST call
- K8s Plugin accepts input time parameters in time of instantiation
 - Instantiation time parameters allow for providing more “dynamic” parameters aside from profile thus reducing total amount of created profiles and their footprint
 - It eliminates the need of creation of custom profiles for many cases

Changes for Frankfurt: CDS integration for CNFs



Agenda

- Introduction
- CNF workflow in El Alto
- Changes for Frankfurt: CDS integration for CNFs
- **Future Plans/Possibilities (for CNFs/vFW use-case)**
- vFW use-case
- Discussion

Future Plans/Possibilities (for CNFs/vFW use-case)

- New Scaling Capability in K8s Plugin → Still to be designed
 - Will expose scaling functionality for selected resources created after instantiation of RB Definition: Deployment and Stateful Set
 - Dedicated API in Multicloud K8S
 - Integration with ScaleOut and ScaleIn generic Workflows
 - ScaleOut for VNFs creates always new heat stack - vf-module instance
 - ScaleOut of RB Instance (at least today) would result only in change of available vservers under the same vf-module → we should keep Scaling fasion the same for VNFs and CNFs
 - Requires proper alignment of vf-module's role for VNFs and CNFs
 - May result with specific requirements given on Helm Charts, i.e. only one Deployment or Stateful Set in RB Definition → Otherwise ONAP would have to select K8s resource for scaling
 - Requires proper design of helm chart in order to deliver input parameters for new pods in scaled Deployment
 - Update of information about generic-vnf, vf-module, vserver in AAI

Future Plans/Possibilities (for CNFs/vFW use-case)

- New Upgrade Capability in K8s Plugin → Still to be designed
 - Upgrade of existing RB Definition Instance base on new RB Profile or new RB Definition
 - Upgrade should works similarly to Helm Chart Upgrade
 - Existing RB Instance will be moved to new RB Definition
 - With RB Profile we can modify the RB Definition so with appropriate profile we can achieve almost full upgrade like with new RB Definition (right balance: RB Definition vs RB Profile)
- vFW CNF (Build and Replace) Upgrade with new definition or profile
 - (?) Upgrade of existing VSP with new CSAR with new Helm Packages through SDC and creation of new vf-model(s)
 - Upload of new vFW CNF CBA with new RB Profile(s) valid with new VSP
 - Modification of existing service model by replacing new vf-models with new versions
 - With distribution of service model new RB Definition(s) appear in K8s Plugin

Future Plans/Possibilities (for CNFs/vFW use-case)

- vFW CNF Upgrade with new definition (cd..)
 - Creation of new service instance but **with reference to existing one service**
 - CDS resolves parameters for instantiation i.e. with different profile name or different images
 - Inputs can be initially taken from those used for instantiation of referenced one
 - CDS Additionally resolves names of existing RB Instances – they are stored in AAI, in vf-module object
 - New Service instance is created
 - K8s Plugin Instantiation Broker receives name of existing RB Instances and calls Upgrade controller instead of creating new RB Instance – old RB Instance will disappear
 - As a result new generic VNF and new VF-modules are created
 - Old Service Instance is deleted what would not impact any resources on K8s
 - Old VF-Modules would refer to already non-existing RB Instances and can be easily removed

Agenda

- Introduction
- CNF workflow in El Alto
- Changes for Frankfurt: CDS integration for CNFs
- Future Plans/Possibilities (for CNFs/vFW use-case)
- **vFW use-case**
- Discussion

vFW use-case

Use-case located in
Demo → [heat/vFW_CNF_CDS](#)

[\[demo.git\]](#) / [heat](#) / [vFW_CNF_CDS](#) / [templates](#) /

```
drwxr-xr-x  .  ..
-rw-r--r--  24 .gitignore blob | history | raw
-rw-r--r-- 266 Makefile blob | history | raw
-rw-r--r--  67 README.txt blob | history | raw
drwxr-xr-x  -  base tree | history
-rw-r--r-- 15035 cba-dd.json blob | history | raw
drwxr-xr-x  -  cba tree | history
drwxr-xr-x  -  helm tree | history
```

Agenda

- Introduction
- CNF workflow in El Alto
- Changes for Frankfurt: CDS integration for CNFs
- Future Plans/Possibilities (for CNFs/vFW use-case)
- vFW use-case
- Discussion



ONAP

OPEN NETWORK AUTOMATION PLATFORM

Thank you