

CNF failover by ONAP CDS and Policy

Linux Foundation Networking - 2022 Jan

Akos Leiter, Attila Hegyi, Peter Fazekas

11-01-2022

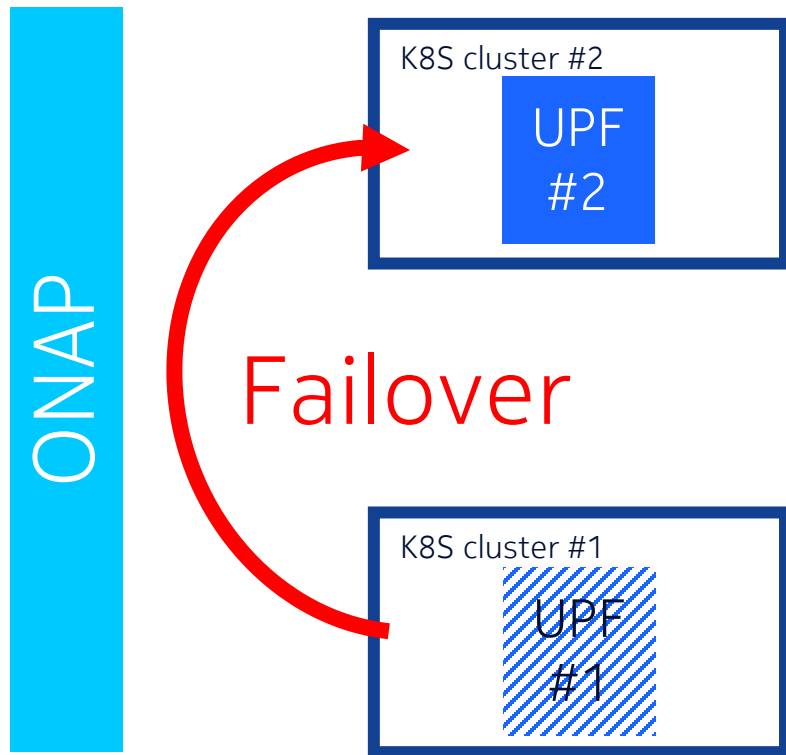
Agenda

Overview

Demo architecture

Demo

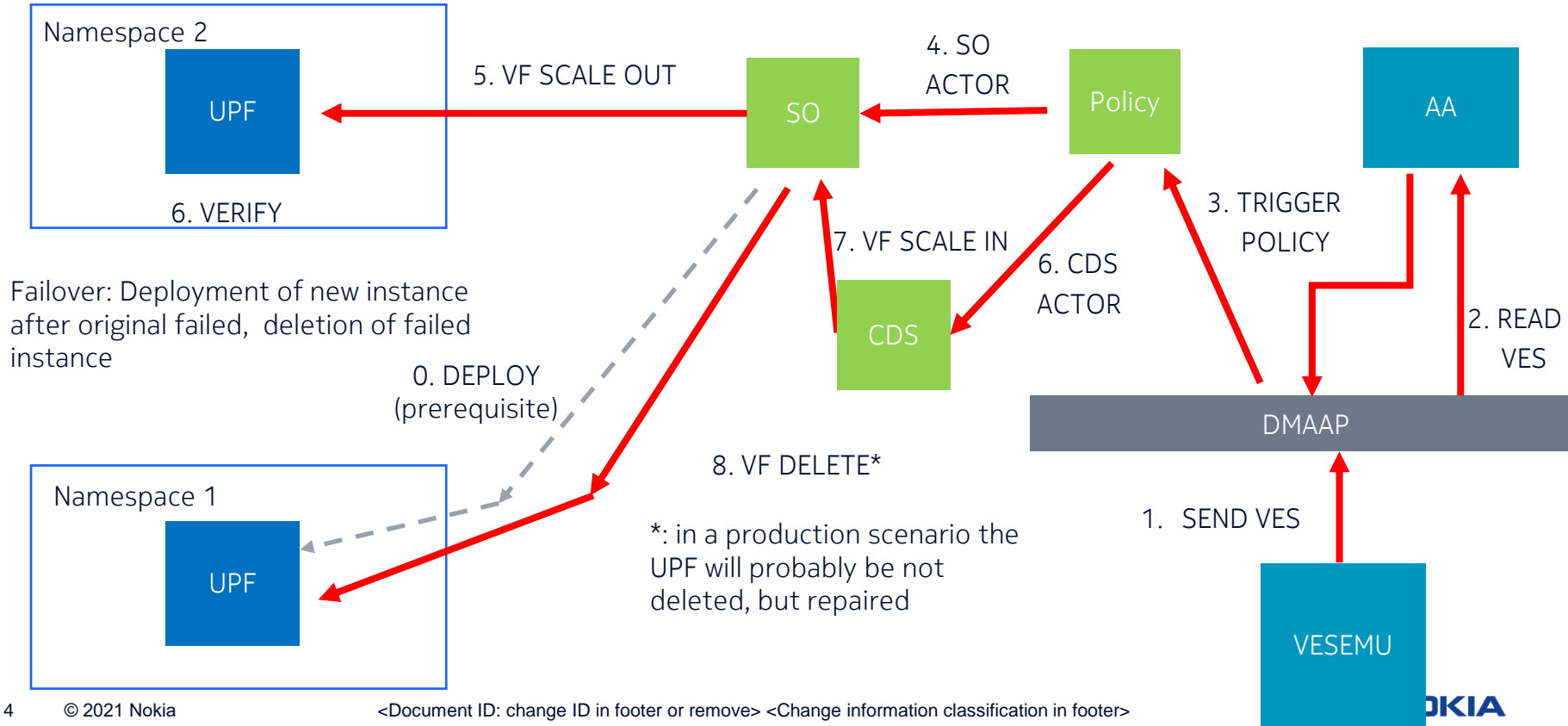
Overview



- UPF failure is detected by ONAP
- ONAP starts failover:
 - New UPF instance is deployed to a second K8S cluster
 - Faulty instance is deleted

Demo scenario overview

Namespace-to-namespace failover



Components

UPF with minimal configuration

Workflows:

VNF remains but different VF modules

Scale out / scale in

- These are supported with SO actor in Policy
 - Add / remove VF modules
- Usecase: delete / reinstantiate complete VF
 - VF is single VF module
- Using scale out / in allows more general usecases

Custom microservices for the usecase

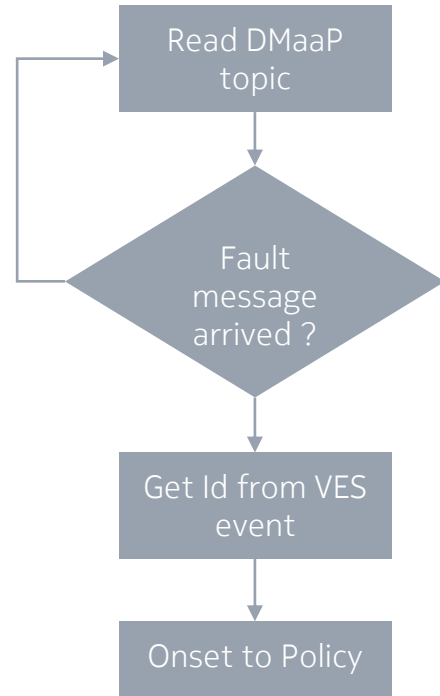
VESEMU

- Manually trigger a failure message to DMaaP
- Incorporates the necessary payload to further processing

AA

- Reads the relevant DMaaP topic
- Sends onset to Policy (via DMaaP)

```
{
  "event": {
    "commonEventHeader": {
      "domain": "fault",
      "eventId": "$EVENT_ID",
      "eventName": "vnfFaultEvent",
      "eventType": "faultEvent",
      "internalHeaderFields": {},
      "lastEpochMicrosec": 1,
      "nfNamingCode": "somestring",
      "nfcNamingCode": "somestring",
      "priority": "High",
      "reportingEntityId": "vnf_id",
      "reportingEntityName": "vnf_name",
      "sequence": 1,
      "sourceId": "5043408b-bd15-4595-aad0-1ac6c86f79c6",
      "sourceName": "vnf_name",
      "startEpochMicrosec": 1,
      "version": 1
    }
  }
}
```



Policy for failover

Drools operational policy with two operation

SO-Actor

- Built in vf-module scaleOut operation
- In Frankfurt it had its limitations
- Operation for scaleIn is not implemented correctly

CDS actor:

- Operation to call any CDS workflow
- We implemented the vf-module scaleIn in CDS

```
"operational.failover": {
  "type": "onap.policies.controlloop.operational.common.Drools",
  "name": "operational.failover",
  "version": "1.0.0",
  "type_version": "1.0.0",
  "metadata": {
    "policy-id": "operational.failover",
    "policy-version": "1.0.0"
  },
  "properties": {
    "id": "ControlLoop-cmgfailover-1",
    "trigger": "unique-policy-id-1-createvf",
    "controllerName": "usecases",
    "operations": [
      {
        "id": "unique-policy-id-1-createvf",
        "description": "Create VF",
        "operation": {
          "actor": "SO",
          "operation": "VF Module Create",
          "target": {
            "targetType": "VFMODULE",
            "entityIds": {
              "resourceID": ...
            }
          }
        },
        "payload": {
          "requestParameters": "{ 'usePreload': false, 'userParams': [] }",
          "configurationParameters": "[]"
        },
        "success": "unique-policy-id-1-deletevf-cds",
        "failure": "final_failure",
      },
      {
        "id": "unique-policy-id-1-deletevf-cds",
        "description": "Execute a scalein workflow",
        "operation": {
          "actor": "CDS",
          "operation": "scalein-workflow",
          "target": {
            "targetType": "VNF",
            "entityIds": {
              "resourceID": ...
            }
          }
        },
        "payload": {
          "artifact_name": "CMG UPF CNF_CDS",
          "artifact_version": "3.0.0",
          "data": "{}"
        },
        "success": "final_success",
        "failure": "final_failure",
      }
    ]
  }
}
```

Policy for failover

Drools operational policy – CDS actor

We had experimented with the CDS actor:

- Can trigger any custom CDS workflow
- Sending a gRPC execution request with:
 - **service-instance-id** and **vnf-id**
 - additional **payload** defined in the policy
- We implemented a **scaleIn** CDS workflow with Kotlin execution:
 - Collecting information from AAI
 - Determine vf-module to delete
 - Send a vf-module delete request to SO

```
"scalein-workflow": {
  "steps": {
    "scalein-step-1": {
      "description": "Resource Assign in workflow(scalein-workflow)",
      "target": "scalein-resolution",
      "activities": [
        {
          "call_operation": "ResourceResolutionComponent.process"
        }
      ],
      "on_success": [
        "scalein-step-2"
      ]
    },
    "scalein-step-2": {
      "description": "Scalein Workflow kotlin execution",
      "target": "scalein-target",
      "activities": [
        {
          "call_operation": ""
        }
      ]
    }
  ],
  "inputs": {
    "resolution-key": {
      "required": true,
      "type": "string"
    },
    "scalein-workflow-properties": {
      "description": "Dynamic PropertyDefinition for scalein-workflow.",
      "required": true,
      "type": "dt-scalein-workflow-properties"
    }
  }
}
```

Q&A so far?

Demo

Q&A

NOKIA