

MultiCloud K8s-Plugin-service API

- 1 Definitions API
 - 1.1 Managing Definitions
 - 1.1.1 Command to create (POST) Definition:
 - 1.1.2 Command to UPLOAD artifact for Definition Created:
 - 1.1.3 Command to GET Definitions:
 - 1.1.4 Command to DELETE Definitions:
- 2 Profiles API
 - 2.1 Command to create (POST) Profile
 - 2.2 Command to UPLOAD artifact for Profile
 - 2.3 Command to GET Profiles
 - 2.4 Command to DELETE Profile
- 3 Instantiation API
 - 3.1 Delete Instantiated Kubernetes resources
 - 3.2 GET Instantiated Kubernetes resources
 - 3.3 LIST all Instantiated Kubernetes resources
- 4 Status API
 - 4.1 GET Instantiated Kubernetes resource's status
- 5 Configuration API
 - 5.1 Configuration Template API
 - 5.1.1 Command to POST (create) Template:
 - 5.1.2 Command to UPLOAD Template:
 - 5.1.3 Command to GET Templates:
 - 5.1.4 Command to DELETE Templates:
 - 5.1.5 Example
 - 5.2 Configuration Values API
 - 5.2.1 Command to POST (create) Configuration Values
 - 5.2.2 Command to DELETE Configuration Values
 - 5.2.3 Command to PUT (Modify) Configuration Values
 - 5.2.4 Command to TAG configuration Values
 - 5.2.5 Command to ROLLBACK configuration Values
- 6 Reachability/Connectivity Info API
 - 6.1 Command to POST Connectivity Info
 - 6.2 Command to GET Connectivity Info
 - 6.3 Command to DELETE Connectivity Info
 - 6.4 Command to UPDATE/PUT Connectivity Info
- 7 Infrastructure workload LCM API

Definitions API

K8splugin artifacts start in the form of Definitions. These are nothing but Helm Charts wrapped with some metadata about the chart itself.

The package can be generated using the helm package command.

A Sample Helm Chart can be viewed in this patch: <https:// Gerrit.onap.org/r/#/c/76371/>

The contents for that chart are below.

```
> cd vagrant/tests/vnfs/testrb/helm
> find vault-consul-dev
vault-consul-dev
vault-consul-dev/Chart.yaml
vault-consul-dev/values.yaml
vault-consul-dev/templates
vault-consul-dev/templates/service.yaml
vault-consul-dev/templates/deployment.yaml
vault-consul-dev/charts
vault-consul-dev/charts/common/values.yaml
vault-consul-dev/charts/common/templates/_service.tpl
vault-consul-dev/charts/common/templates/_repository.tpl
vault-consul-dev/charts/common/templates/_name.tpl
vault-consul-dev/charts/common/templates/_namespace.tpl
vault-consul-dev/charts/common/templates
vault-consul-dev/charts/common/Chart.yaml
vault-consul-dev/charts/common

#Create a tar.gz for upload
> tar -cf vault-consul-dev.tar vault-consul-dev
> gzip vault-consul-dev.tar
```

Managing Definitions

The following curl commands can be used to upload a definition and the helm chart for that definition.

With the following JSON content

```
{
  "rb-name": "test-rbdef",
  "rb-version": "v1",
  "chart-name": "vault-consul-dev", //optional field. chart-name will be detected if this is not provided.
  "description": "testing resource bundle definition api",
  "labels": {
  }
}
```

Command to create (POST) Definition:

```
curl -i -d @create_rbdefinition.json -X POST http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition
```

Command to UPLOAD artifact for Definition Created:

```
curl -i --data-binary @vault-consul-dev.tar.gz -X POST http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/test-rbdef/v1/content
```

Command to GET Definitions:

```
# Get all Definitions
curl -i http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition

# Get one Definition
curl -i http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/test-rbdef/v1
```

Command to DELETE Definitions:

```
#Delete all versions of definition
curl -i -X DELETE http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/test-rbdef
#Delete a particular version of definition
curl -i -X DELETE http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/test-rbdef/v1
```

Profiles API

Once the Definitions are created, we are ready to create some profiles so that we can customize that definition and instantiate it in Kubernetes.

A profile contains the following:

1. **manifest.yaml**
 - a. Contains the details for the profile and everything contained within
2. A **HELM** values override yaml file.
 - a. It can have any name as long as it matches the corresponding entry in the **manifest.yaml**
3. Any number of files organized in a folder structure
 - a. All these files should have a corresponding entry in **manifest.yaml** file

Sample Profile is described in the above patch

Create the profile artifact

Creating a Profile Artifact

```
> cd vagrant/tests/vnfs/testrb/helm/profile
> find .
manifest.yaml
override_values.yaml
testfol
testfol/subdir
testfol/subdir/deployment.yaml

#Create profile tar.gz
> cd profile
> tar -cf profile.tar *
> gzip profile.tar
> mv profile.tar.gz ../
```

Manifest file contains following

manifest.yaml

```
---
version: v1
type:
  values: "values_override.yaml"
configresource:
  - filepath: testfol/subdir/deployment.yaml
    chartpath: vault-consul-dev/templates/deployment.yaml
```

With this information, we are ready to upload the profile with the following JSON data

Profile JSON Body

```
{
  "rb-name": "test-rbdef",
  "rb-version": "v1",
  "profile-name": "p1",
  "release-name": "r1", //If release-name is not provided, profile-name will be used
  "namespace": "testnamespace1",
  "kubernetes-version": "1.12.3"
}
```

Command to create (POST) Profile

```
#Replace NODE_IP:30280/api/multicloud-k8s/v1 with NODE_PORT:30280/api/multicloud-k8s/v1 if you are connec
curl -i -d @create_rbprofile.json -X POST http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/test-rbdef/v1/profile
```

Command to UPLOAD artifact for Profile

```
curl -i --data-binary @profile.tar.gz -X POST http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/test-rbdef/v1/profile/p1/content
```

Command to GET Profiles

```
# Get all Profiles
curl -i http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/test-rbdef/v1/profile

# Get one Profile
curl -i http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/test-rbdef/v1/profile/p1
```

Command to **DELETE** Profile

```
curl -i -X DELETE http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/test-rbdef/v1/profile/p1
```

Instantiation API

Instantiate the created Profile via the following REST api

Using the following JSON:

```
{
  "cloud-region": "krd",
  "profile-name": "p1",
  "release-name": "release-x",
  "rb-name": "test-rbdef",
  "rb-version": "v1",
  "override-values": {
    "optionalDictOfParameters": "andTheirValues, like",
    "global.name": "dummy-name"
  },
  "labels": {
    "optionalLabelForInternalK8spluginInstancesMetadata": "dummy-value"
  }
}
```

Please note that both **labels**, **override-parameters** and **release-name** fields are optional.

When not provided, "**release-name**" will be reused from profile definition, however user should intend to provide own release name during instantiation if he wants to reuse same profile for multiple instances.

Instantiate the profile with the ID provided above

Command to Instantiate a Profile

```
curl -d @create_rbinstance.json http://NODE_IP:30280/api/multicloud-k8s/v1/v1/instance
```

This command returns the following JSON:

```
{
  "id": "ID_GENERATED_BY_K8SPLUGIN",
  "namespace": "NAMESPACE_WHERE_INSTANCE_HAS_BEEN_DEPLOYED_AS_DERIVED_FROM_PROFILE",
  "release-name": "RELEASE_NAME_AS_COMPUTED_BASED_ON_INSTANTIATION_REQUEST_AND_PROFILE_DEFAULT",
  "request": {
    "rb-name": "test-rbdef",
    "rb-version": "v1",
    "profile-name": "p1",
    "release-name": "release-x",
    "cloud-region": "krd",
    "override-values": {
      "optionalDictOfParameters": "andTheirValues, like",
      "global.name": "dummy-name"
    },
  },
  "labels": {
    "optionalLabelForInternalK8spluginInstancesMetadata": "dummy-value"
  },
}
```

```

    }
    "resources": [
{
      "GVK": {
{
          "Group":
"",
          "Kind":
"ConfigMap",
          "Version":
"v1"
        },
{
          "Name": "test-
cm"
        },
      },
{
      "GVK": {
{
          "Group":
"",
          "Kind":
"Service",
          "Version":
"v1"
        },
{
          "Name": "test-
svc"
        },
      },
{
      "GVK": {
        "Group": "apps",
        "Kind": "Deployment",
        "Version": "v1"
      },
      "Name": "test-dep"
    }
  ]
}

```

Delete Instantiated Kubernetes resources

The **id** field from the returned JSON can be used to **DELETE** the resources created in the previous step

This executes a Delete operation using the Kubernetes API.

Command to DELETE a resource

```
curl -X DELETE http://NODE_IP:30280/api/multicloud-k8s/v1/v1/instance/ZKMTSaxv
```

GET Instantiated Kubernetes resources

The **id** field from the returned JSON can be used to **GET** the resources created in the previous step

Command to GET a resource

```
curl -X GET http://NODE_IP:30280/api/multicloud-k8s/v1/v1/instance/ZKMTSaxv
```

Output of this call conforms to the same schema and contains (currently) same information as instantiation response (described above)

LIST all Instantiated Kubernetes resources

The following API can be used to **GET** all the instances we have created.

Command to GET all resources

```
curl -X GET http://NODE_IP:30280/api/multicloud-k8s/v1/v1/instance
```

Status API

Status API allows for monitoring/querying for instantiated CNFs status. It allows checking both design-time (eg. image name, node port) or runtime (eg. container statuses, deployment readiness) capabilities of resources managed within k8splugin instances.

GET Instantiated Kubernetes resource's status

Command to GET instance status

```
curl -X GET http://NODE_IP:30280/api/multicloud-k8s/v1/v1/instance/{instanceID}/status
```

Example response:

Example response

```
{
  "request": {
    "rb-name": "vfw",
    "rb-version": "plugin_test",
    "profile-name": "test_profile",
    "release-name": "",
    "cloud-region": "kud",
    "labels": {
      "testCaseName": "plugin_fw.sh"
    },
    "override-values": {
      "global.onapPrivateNetworkName": "onap-private-net-test"
    }
  },
  "ready": false,
  "resourceCount": 12,
  "resourcesStatus": [
    {
      "name": "sink-configmap",
      "GVK": {
        "Group": "",
        "Version": "v1",
        "Kind": "ConfigMap"
      }
    }
  ]
}
```

```

    },
    "status": {
      "apiVersion": "v1",
      "data": {
        "protected_net_gw": "192.168.20.100",
        "protected_private_net_cidr": "192.168.10.0/24"
      },
      "kind": "ConfigMap",
      "metadata": {
        "creationTimestamp": "2020-09-29T13:26:51Z",
        "labels": {
          "k8splugin.io/rb-instance-id": "stoic_wright"
        },
        "name": "sink-configmap",
        "namespace": "plugin-tests-namespace",
        "resourceVersion": "10719454",
        "selfLink": "/api/v1/namespaces/plugin-tests-namespace/configmaps/sink-configmap",
        "uid": "f3199d15-2bd4-4e04-96a0-9e390757b43b"
      }
    }
  },
  {
    "name": "packetgen-service",
    "GVK": {
      "Group": "",
      "Version": "v1",
      "Kind": "Service"
    },
    "status": {
      "apiVersion": "v1",
      "kind": "Service",
      "metadata": {
        "creationTimestamp": "2020-09-29T13:26:51Z",
        "labels": {
          "app": "packetgen",
          "chart": "packetgen",
          "k8splugin.io/rb-instance-id": "stoic_wright",
          "release": "test-release"
        },
        "name": "packetgen-service",
        "namespace": "plugin-tests-namespace",
        "resourceVersion": "10719457",
        "selfLink": "/api/v1/namespaces/plugin-tests-namespace/services/packetgen-service",
        "uid": "d1d1e571-8ec0-4046-ba89-9c974c99f7cb"
      },
      "spec": {
        "clusterIP": "10.244.36.238",
        "externalTrafficPolicy": "Cluster",
        "ports": [
          {
            "nodePort": 30831,
            "port": 2831,
            "protocol": "TCP",
            "targetPort": 2831
          }
        ],
        "selector": {
          "app": "packetgen",
          "release": "test-release"
        },
        "sessionAffinity": "None",
        "type": "NodePort"
      },
      "status": {
        "loadBalancer": {}
      }
    }
  },
  {
    "name": "sink-service",
    "GVK": {

```

```

    "Group": "",
    "Version": "v1",
    "Kind": "Service"
  },
  "status": {
    "apiVersion": "v1",
    "kind": "Service",
    "metadata": {
      "creationTimestamp": "2020-09-29T13:26:51Z",
      "labels": {
        "app": "sink",
        "chart": "sink",
        "k8splugin.io/rb-instance-id": "stoic_wright",
        "release": "test-release"
      },
      "name": "sink-service",
      "namespace": "plugin-tests-namespace",
      "resourceVersion": "10719461",
      "selfLink": "/api/v1/namespaces/plugin-tests-namespace/services/sink-service",
      "uid": "d9e81332-dea6-419b-bd63-f4bec4514d33"
    },
    "spec": {
      "clusterIP": "10.244.30.37",
      "externalTrafficPolicy": "Cluster",
      "ports": [
        {
          "nodePort": 30667,
          "port": 667,
          "protocol": "TCP",
          "targetPort": 667
        }
      ],
      "selector": {
        "app": "sink",
        "release": "test-release"
      },
      "sessionAffinity": "None",
      "type": "NodePort"
    },
    "status": {
      "loadBalancer": {}
    }
  }
},
{
  "name": "test-release-packetgen",
  "GVK": {
    "Group": "apps",
    "Version": "v1",
    "Kind": "Deployment"
  },
  "status": {
    "apiVersion": "apps/v1",
    "kind": "Deployment",
    "metadata": {
      "annotations": {
        "deployment.kubernetes.io/revision": "1"
      },
      "creationTimestamp": "2020-09-29T13:26:51Z",
      "generation": 1,
      "labels": {
        "app": "packetgen",
        "chart": "packetgen",
        "k8splugin.io/rb-instance-id": "stoic_wright",
        "release": "test-release"
      },
      "name": "test-release-packetgen",
      "namespace": "plugin-tests-namespace",
      "resourceVersion": "10719551",
      "selfLink": "/apis/apps/v1/namespaces/plugin-tests-namespace/deployments/test-release-
packetgen",

```



```

    "uid": "6538a28f-9b9c-42bf-a3bf-cf33525c6748"
  },
  "spec": {
    "progressDeadlineSeconds": 600,
    "replicas": 1,
    "revisionHistoryLimit": 10,
    "selector": {
      "matchLabels": {
        "app": "packetgen",
        "release": "test-release"
      }
    },
    "strategy": {
      "rollingUpdate": {
        "maxSurge": "25%",
        "maxUnavailable": "25%"
      },
      "type": "RollingUpdate"
    },
    "template": {
      "metadata": {
        "annotations": {
          "VirtletCloudInitUserData": "ssh_pwauth: True\nusers:\n- name: admin\n  gecos:
User\n  primary-group: admin\n  groups: users\n  sudo: ALL=(ALL) NOPASSWD:ALL\n  lock_passwd: false\n  passwd:
\"$6$rounds=4096$QA50CKHTE41$jRACivoPMJcOjLRgx13t.AMfU7LhCFwOWv2z66CQX.TSxBy50JoYtycJXSPR2JceG.8Tq
/82QN9QYt3euYEZW/\n\nrunCmd:\n  - export demo_artifacts_version=1.5.0\n  - export vfw_private_ip_0=192.168.10.3
\n  - export vsn_private_ip_0=192.168.20.3\n  - export protected_net_cidr=192.168.20.0/24\n  - export
dcae_collector_ip=10.0.4.1\n  - export dcae_collector_port=8081\n  - export protected_net_gw=192.168.20.100
/24\n  - export protected_private_net_cidr=192.168.10.0/24\n  - wget -O - https://git.onap.org/multicloud/k8s
/plain/kud/tests/vFW/packetgen | sudo -E bash\n",
          "VirtletLibvirtCPUSetting": "mode: host-model\n",
          "VirtletRootVolumeSize": "5Gi",
          "app": "packetgen",
          "k8s.plugin.opnfv.org/nfn-network": "{ \"type\": \"ovn4nfv\", \"interface\": { {
\"name\": \"unprotected-private-net\", \"ipAddress\": \"192.168.10.2\", \"interface\": \"eth1\", \"
defaultGateway\": \"false\"}, { \"name\": \"onap-private-net-test\", \"ipAddress\": \"10.0.100.2\", \"
interface\": \"eth2\", \"defaultGateway\": \"false\"} ] }",
          "k8s.v1.cni.cncf.io/networks": "[{ \"name\": \"ovn-networkobj\", \"namespace\":
\"default\"}]",
          "kubernetes.io/target-runtime": "virtlet.cloud",
          "release": "test-release"
        },
        "creationTimestamp": null,
        "labels": {
          "app": "packetgen",
          "k8splugin.io/rb-instance-id": "stoic_wright",
          "release": "test-release"
        }
      },
      "spec": {
        "affinity": {
          "nodeAffinity": {
            "requiredDuringSchedulingIgnoredDuringExecution": {
              "nodeSelectorTerms": [
                {
                  "matchExpressions": [
                    {
                      "key": "extraRuntime",
                      "operator": "In",
                      "values": [
                        "virtlet"
                      ]
                    }
                  ]
                }
              ]
            }
          }
        },
        "containers": [
          {

```

```

        "image": "virtlet.cloud/ubuntu/16.04:latest",
        "imagePullPolicy": "IfNotPresent",
        "name": "packetgen",
        "resources": {
            "limits": {
                "memory": "4Gi"
            }
        },
        "stdin": true,
        "terminationMessagePath": "/dev/termination-log",
        "terminationMessagePolicy": "File",
        "tty": true
    }
},
    "dnsPolicy": "ClusterFirst",
    "restartPolicy": "Always",
    "schedulerName": "default-scheduler",
    "securityContext": {},
    "terminationGracePeriodSeconds": 30
}
},
    "status": {
        "availableReplicas": 1,
        "conditions": [
            {
                "lastTransitionTime": "2020-09-29T13:27:02Z",
                "lastUpdateTime": "2020-09-29T13:27:02Z",
                "message": "Deployment has minimum availability.",
                "reason": "MinimumReplicasAvailable",
                "status": "True",
                "type": "Available"
            },
            {
                "lastTransitionTime": "2020-09-29T13:26:51Z",
                "lastUpdateTime": "2020-09-29T13:27:02Z",
                "message": "ReplicaSet \"test-release-packetgen-7557cb58bb\" has successfully
progressed.",
                "reason": "NewReplicaSetAvailable",
                "status": "True",
                "type": "Progressing"
            }
        ],
        "observedGeneration": 1,
        "readyReplicas": 1,
        "replicas": 1,
        "updatedReplicas": 1
    }
}
},
{
    "name": "test-release-sink",
    "GVK": {
        "Group": "apps",
        "Version": "v1",
        "Kind": "Deployment"
    },
    "status": {
        "apiVersion": "apps/v1",
        "kind": "Deployment",
        "metadata": {
            "annotations": {
                "deployment.kubernetes.io/revision": "1"
            },
            "creationTimestamp": "2020-09-29T13:26:51Z",
            "generation": 1,
            "labels": {
                "app": "sink",
                "chart": "sink",
                "k8splugin.io/rb-instance-id": "stoic_wright",
                "release": "test-release"
            }
        }
    }
}

```

```

    },
    "name": "test-release-sink",
    "namespace": "plugin-tests-namespace",
    "resourceVersion": "10719538",
    "selfLink": "/apis/apps/v1/namespaces/plugin-tests-namespace/deployments/test-release-sink",
    "uid": "2200dbb3-5d83-4f0c-9e3a-95f1bd0abb46"
  },
  "spec": {
    "progressDeadlineSeconds": 600,
    "replicas": 1,
    "revisionHistoryLimit": 10,
    "selector": {
      "matchLabels": {
        "app": "sink",
        "release": "test-release"
      }
    },
    "strategy": {
      "rollingUpdate": {
        "maxSurge": "25%",
        "maxUnavailable": "25%"
      },
      "type": "RollingUpdate"
    },
    "template": {
      "metadata": {
        "annotations": {
          "k8s.plugin.opnfv.org/nfn-network": "{ \"type\": \"ovn4nfv\", \"interface\": [
{ \"name\": \"protected-private-net\", \"ipAddress\": \"192.168.20.3\", \"interface\": \"eth1\", \"
defaultGateway\": \"false\" }, { \"name\": \"onap-private-net-test\", \"ipAddress\": \"10.10.100.4\", \"
interface\": \"eth2\", \"defaultGateway\": \"false\" } ] }",
          "k8s.v1.cni.cncf.io/networks": "[{ \"name\": \"ovn-networkobj\", \"namespace\":
\"default\" }]"
        },
        "creationTimestamp": null,
        "labels": {
          "app": "sink",
          "k8splugin.io/rb-instance-id": "stoic_wright",
          "release": "test-release"
        }
      },
      "spec": {
        "containers": [
          {
            "envFrom": [
              {
                "configMapRef": {
                  "name": "sink-configmap"
                }
              }
            ],
            "image": "rtsood/onap-vfw-demo-sink:0.2.0",
            "imagePullPolicy": "IfNotPresent",
            "name": "sink",
            "resources": {},
            "securityContext": {
              "privileged": true
            },
            "stdin": true,
            "terminationMessagePath": "/dev/termination-log",
            "terminationMessagePolicy": "File",
            "tty": true
          },
          {
            "image": "electrocucaracha/darkstat:latest",
            "imagePullPolicy": "IfNotPresent",
            "name": "darkstat",
            "ports": [
              {
                "containerPort": 667,
                "protocol": "TCP"
              }
            ]
          }
        ]
      }
    }
  }
}

```

```

        }
        ],
        "resources": {},
        "stdin": true,
        "terminationMessagePath": "/dev/termination-log",
        "terminationMessagePolicy": "File",
        "tty": true
    }
    ],
    "dnsPolicy": "ClusterFirst",
    "restartPolicy": "Always",
    "schedulerName": "default-scheduler",
    "securityContext": {},
    "terminationGracePeriodSeconds": 30
}
},
"status": {
    "availableReplicas": 1,
    "conditions": [
        {
            "lastTransitionTime": "2020-09-29T13:27:00Z",
            "lastUpdateTime": "2020-09-29T13:27:00Z",
            "message": "Deployment has minimum availability.",
            "reason": "MinimumReplicasAvailable",
            "status": "True",
            "type": "Available"
        },
        {
            "lastTransitionTime": "2020-09-29T13:26:51Z",
            "lastUpdateTime": "2020-09-29T13:27:00Z",
            "message": "ReplicaSet \"test-release-sink-766684c678\" has successfully
progressed.",
            "reason": "NewReplicaSetAvailable",
            "status": "True",
            "type": "Progressing"
        }
    ],
    "observedGeneration": 1,
    "readyReplicas": 1,
    "replicas": 1,
    "updatedReplicas": 1
}
},
{
    "name": "test-release-firewall",
    "GVK": {
        "Group": "apps",
        "Version": "v1",
        "Kind": "Deployment"
    },
    "status": {
        "apiVersion": "apps/v1",
        "kind": "Deployment",
        "metadata": {
            "annotations": {
                "deployment.kubernetes.io/revision": "1"
            },
            "creationTimestamp": "2020-09-29T13:26:51Z",
            "generation": 1,
            "labels": {
                "app": "firewall",
                "chart": "firewall",
                "k8splugin.io/rb-instance-id": "stoic_wright",
                "release": "test-release"
            },
            "name": "test-release-firewall",
            "namespace": "plugin-tests-namespace",
            "resourceVersion": "10719556",
            "selfLink": "/apis/apps/v1/namespaces/plugin-tests-namespace/deployments/test-release-

```

```

firewall",
    "uid": "a9c3c5b0-f793-49be-a8cd-2a83d0905643"
  },
  "spec": {
    "progressDeadlineSeconds": 600,
    "replicas": 1,
    "revisionHistoryLimit": 10,
    "selector": {
      "matchLabels": {
        "app": "firewall",
        "release": "test-release"
      }
    },
    "strategy": {
      "rollingUpdate": {
        "maxSurge": "25%",
        "maxUnavailable": "25%"
      },
      "type": "RollingUpdate"
    },
    "template": {
      "metadata": {
        "annotations": {
          "VirtletCloudInitUserData": "ssh_pwauth: True\nusers:\n- name: admin\n  gecos: User\n  primary-group: admin\n  groups: users\n  sudo: ALL=(ALL) NOPASSWD:ALL\n  lock_passwd: false\n  passwd: \"$6$rounds=4096$QA5OCKHTE41$jRACivoPMJcoJLRgx13t.AMfU7LhCFwOWv2z66CQX.TSxBy50JoYtycJXSpr2JceG.8Tq/82QN9QYt3euYEZW/\n\nrunCmd:\n  - export demo_artifacts_version=1.5.0\n  - export vfw_private_ip_0=192.168.10.3\n  - export vsn_private_ip_0=192.168.20.3\n  - export protected_net_cidr=192.168.20.0/24\n  - export dcae_collector_ip=10.0.4.1\n  - export dcae_collector_port=8081\n  - export protected_net_gw=192.168.20.100/24\n  - export protected_private_net_cidr=192.168.10.0/24\n  - wget -O - https://git.onap.org/multicloud/k8s/plain/kud/tests/vFW/firewall | sudo -E bash\n",
          "VirtletLibvirtCPUSetting": "mode: host-model\n",
          "VirtletRootVolumeSize": "5Gi",
          "k8s.plugin.opnfv.org/nfn-network": "{ \"type\": \"ovn4nfv\", \"interface\": [ { \"name\": \"unprotected-private-net\", \"ipAddress\": \"192.168.10.3\", \"interface\": \"eth1\", \"defaultGateway\": \"false\" }, { \"name\": \"protected-private-net\", \"ipAddress\": \"192.168.20.2\", \"interface\": \"eth2\", \"defaultGateway\": \"false\" }, { \"name\": \"onap-private-net-test\", \"ipAddress\": \"10.10.100.3\", \"interface\": \"eth3\", \"defaultGateway\": \"false\" } ] }",
          "k8s.v1.cni.cncf.io/networks": "[ { \"name\": \"ovn-networkobj\", \"namespace\": \"default\" } ]",
          "kubernetes.io/target-runtime": "virtlet.cloud"
        },
        "creationTimestamp": null,
        "labels": {
          "app": "firewall",
          "k8splugin.io/rb-instance-id": "stoic_wright",
          "release": "test-release"
        }
      },
      "spec": {
        "affinity": {
          "nodeAffinity": {
            "requiredDuringSchedulingIgnoredDuringExecution": {
              "nodeSelectorTerms": [
                {
                  "matchExpressions": [
                    {
                      "key": "extraRuntime",
                      "operator": "In",
                      "values": [
                        "virtlet"
                      ]
                    }
                  ]
                }
              ]
            }
          }
        }
      }
    },
    "containers": [
      {

```

```

        "image": "virtlet.cloud/ubuntu/16.04:latest",
        "imagePullPolicy": "IfNotPresent",
        "name": "firewall",
        "resources": {
            "limits": {
                "memory": "4Gi"
            }
        },
        "stdin": true,
        "terminationMessagePath": "/dev/termination-log",
        "terminationMessagePolicy": "File",
        "tty": true
    }
},
"dnsPolicy": "ClusterFirst",
"restartPolicy": "Always",
"schedulerName": "default-scheduler",
"securityContext": {},
"terminationGracePeriodSeconds": 30
}
},
"status": {
    "availableReplicas": 1,
    "conditions": [
        {
            "lastTransitionTime": "2020-09-29T13:27:03Z",
            "lastUpdateTime": "2020-09-29T13:27:03Z",
            "message": "Deployment has minimum availability.",
            "reason": "MinimumReplicasAvailable",
            "status": "True",
            "type": "Available"
        },
        {
            "lastTransitionTime": "2020-09-29T13:26:51Z",
            "lastUpdateTime": "2020-09-29T13:27:03Z",
            "message": "ReplicaSet \"test-release-firewall-74fd65984f\" has successfully
progressed.",
            "reason": "NewReplicaSetAvailable",
            "status": "True",
            "type": "Progressing"
        }
    ],
    "observedGeneration": 1,
    "readyReplicas": 1,
    "replicas": 1,
    "updatedReplicas": 1
}
},
{
    "name": "onap-private-net-test",
    "GVK": {
        "Group": "k8s.plugin.opnfv.org",
        "Version": "v1alpha1",
        "Kind": "Network"
    },
    "status": {
        "apiVersion": "k8s.plugin.opnfv.org/v1alpha1",
        "kind": "Network",
        "metadata": {
            "creationTimestamp": "2020-09-29T13:26:51Z",
            "finalizers": [
                "nfnCleanUpNetwork"
            ],
            "generation": 2,
            "labels": {
                "k8splugin.io/rb-instance-id": "stoic_wright"
            },
            "name": "onap-private-net-test",
            "namespace": "plugin-tests-namespace",

```

```

        "resourceVersion": "10719506",
        "selfLink": "/apis/k8s.plugin.opnfv.org/v1alpha1/namespaces/plugin-tests-namespace/networks/onap-private-net-test",
        "uid": "dd04cc6b-e5a5-4b60-a055-b1d147d164b4"
    },
    "spec": {
        "cniType": "ovn4nfv",
        "dns": {},
        "ipv4Subnets": [
            {
                "gateway": "10.10.0.1/16",
                "name": "subnet1",
                "subnet": "10.10.0.0/16"
            }
        ]
    },
    "status": {
        "state": "Created"
    }
}
},
{
    "name": "protected-private-net",
    "GVK": {
        "Group": "k8s.plugin.opnfv.org",
        "Version": "v1alpha1",
        "Kind": "Network"
    },
    "status": {
        "apiVersion": "k8s.plugin.opnfv.org/v1alpha1",
        "kind": "Network",
        "metadata": {
            "creationTimestamp": "2020-09-29T13:26:51Z",
            "finalizers": [
                "nfnCleanupNetwork"
            ],
            "generation": 2,
            "labels": {
                "k8splugin.io/rb-instance-id": "stoic_wright"
            },
            "name": "protected-private-net",
            "namespace": "plugin-tests-namespace",
            "resourceVersion": "10719508",
            "selfLink": "/apis/k8s.plugin.opnfv.org/v1alpha1/namespaces/plugin-tests-namespace/networks/protected-private-net",
            "uid": "ae7e3a73-a33b-4c27-869b-9465b7e67348"
        },
        "spec": {
            "cniType": "ovn4nfv",
            "dns": {},
            "ipv4Subnets": [
                {
                    "gateway": "192.168.20.100/24",
                    "name": "subnet1",
                    "subnet": "192.168.20.0/24"
                }
            ]
        },
        "status": {
            "state": "Created"
        }
    }
}
},
{
    "name": "unprotected-private-net",
    "GVK": {
        "Group": "k8s.plugin.opnfv.org",
        "Version": "v1alpha1",
        "Kind": "Network"
    },
    "status": {

```

```

"apiVersion": "k8s.plugin.opnfv.org/v1alpha1",
"kind": "Network",
"metadata": {
  "creationTimestamp": "2020-09-29T13:26:51Z",
  "finalizers": [
    "nfnCleanUpNetwork"
  ],
  "generation": 2,
  "labels": {
    "k8splugin.io/rb-instance-id": "stoic_wright"
  },
  "name": "unprotected-private-net",
  "namespace": "plugin-tests-namespace",
  "resourceVersion": "10719510",
  "selfLink": "/apis/k8s.plugin.opnfv.org/v1alpha1/namespaces/plugin-tests-namespace/networks/unprotected-private-net",
  "uid": "07f4fbe6-a58b-43ab-91a8-10619ab2bb3a"
},
"spec": {
  "cniType": "ovn4nfv",
  "dns": {},
  "ipv4Subnets": [
    {
      "gateway": "192.168.10.1/24",
      "name": "subnet1",
      "subnet": "192.168.10.0/24"
    }
  ]
},
"status": {
  "state": "Created"
}
},
{
  "name": "test-release-firewall-74fd65984f-rnwjd",
  "GVK": {
    "Group": "",
    "Version": "",
    "Kind": ""
  },
  "status": {
    "metadata": {
      "annotations": {
        "VirtletCloudInitUserData": "ssh_pwauth: True\nusers:\n- name: admin\n  gecos: User\n  primary-group: admin\n  groups: users\n  sudo: ALL=(ALL) NOPASSWD:ALL\n  lock_passwd: false\n  passwd:\n  \"$6$rounds=4096$QA50CKHTE41$jRACivoPMJcoJLRgxl3t.AMfU7LhCFwOWv2z66CQX.TSxBy50JoYtycJXSPR2JceG.8Tq\n  /82QN9QYt3euYEZW/\n\n  nruncmd:\n  - export demo_artifacts_version=1.5.0\n  - export vfw_private_ip_0=192.168.10.3\n  - export vsn_private_ip_0=192.168.20.3\n  - export protected_net_cidr=192.168.20.0/24\n  - export dcae_collector_ip=10.0.4.1\n  - export dcae_collector_port=8081\n  - export protected_net_gw=192.168.20.100\n  /24\n  - export protected_private_net_cidr=192.168.10.0/24\n  - wget -O - https://git.onap.org/multicloud/k8s/plain/kud/tests/vFW/firewall | sudo -E bash\n",
        "VirtletLibvirtCPUSetting": "mode: host-model\n",
        "VirtletRootVolumeSize": "5Gi",
        "k8s.plugin.opnfv.org/nfn-network": "{ \"type\": \"ovn4nfv\", \"interface\": [ { \"name\": \"unprotected-private-net\", \"ipAddress\": \"192.168.10.3\", \"interface\": \"eth1\", \"defaultGateway\": \"false\", { \"name\": \"protected-private-net\", \"ipAddress\": \"192.168.20.2\", \"interface\": \"eth2\", \"defaultGateway\": \"false\", { \"name\": \"onap-private-net-test\", \"ipAddress\": \"10.10.100.3\", \"interface\": \"eth3\", \"defaultGateway\": \"false\" } } ]\", \"k8s.plugin.opnfv.org/ovnInterfaces\": \"[{ \"ip_address\": \"192.168.10.3/24\", \"mac_address\": \"00:00:00:99:7c:2a\", \"gateway_ip\": \"192.168.10.1\", \"defaultGateway\": \"false\", \"interface\": \"eth1\", { \"ip_address\": \"192.168.20.2/24\", \"mac_address\": \"00:00:00:d1:a0:b0\", \"gateway_ip\": \"192.168.20.100\", \"defaultGateway\": \"false\", \"interface\": \"eth2\", { \"ip_address\": \"10.10.100.3/16\", \"mac_address\": \"00:00:00:af:84:ed\", \"gateway_ip\": \"10.10.0.1\", \"defaultGateway\": \"false\", \"interface\": \"eth3\" } } ]\", \"k8s.v1.cni.cncf.io/networks\": \"[{ \"name\": \"ovn-networkobj\", \"namespace\": \"default\" } ]\", \"k8s.v1.cni.cncf.io/networks-status\": \"[{ \n  \"name\": \"cni0\", \n  \"interface\": \"virtlet-eth0\", \n  \"ips\": [ \n    \"10.244.64.36\" \n  ], \n  \"mac\": \"0a:58:0a:f4:40:24\", \n  \"default\": true, \n  \"dns\": { } \n ], { \n  \"name\": \"ovn4nfv-k8s-plugin\", \n  \"interface\": \"eth3\", \n  \"ips\": [ \n    \"192.168.10.3\", \n    \"192.168.20.2\", \n    \"10.10.100.3\" \n  ], \n  \""

```



```

mac\": \"00:00:00:af:84:ed\", \n    \"dns\": { }\n}],
    \"kubernetes.io/target-runtime\": \"virtlet.cloud\"
  },
  \"creationTimestamp\": \"2020-09-29T13:26:51Z\",
  \"generateName\": \"test-release-firewall-74fd65984f-\",
  \"labels\": {
    \"app\": \"firewall\",
    \"k8splugin.io/rb-instance-id\": \"stoic_wright\",
    \"pod-template-hash\": \"74fd65984f\",
    \"release\": \"test-release\"
  },
  \"name\": \"test-release-firewall-74fd65984f-rnwjd\",
  \"namespace\": \"plugin-tests-namespace\",
  \"ownerReferences\": [
    {
      \"apiVersion\": \"apps/v1\",
      \"blockOwnerDeletion\": true,
      \"controller\": true,
      \"kind\": \"ReplicaSet\",
      \"name\": \"test-release-firewall-74fd65984f\",
      \"uid\": \"b21d7f15-c133-4bcf-b584-a2177eda82e7\"
    }
  ],
  \"resourceVersion\": \"10719554\",
  \"selfLink\": \"/api/v1/namespaces/plugin-tests-namespace/pods/test-release-firewall-74fd65984f-rnwjd\",
  \"uid\": \"0f469921-cc82-4808-9407-2f67c4801839\"
},
\"spec\": {
  \"affinity\": {
    \"nodeAffinity\": {
      \"requiredDuringSchedulingIgnoredDuringExecution\": {
        \"nodeSelectorTerms\": [
          {
            \"matchExpressions\": [
              {
                \"key\": \"extraRuntime\",
                \"operator\": \"In\",
                \"values\": [
                  \"virtlet\"
                ]
              }
            ]
          }
        ]
      }
    }
  },
  \"containers\": [
    {
      \"image\": \"virtlet.cloud/ubuntu/16.04:latest\",
      \"imagePullPolicy\": \"IfNotPresent\",
      \"name\": \"firewall\",
      \"resources\": {
        \"limits\": {
          \"memory\": \"4Gi\"
        },
        \"requests\": {
          \"memory\": \"4Gi\"
        }
      },
      \"stdin\": true,
      \"terminationMessagePath\": \"/dev/termination-log\",
      \"terminationMessagePolicy\": \"File\",
      \"tty\": true,
      \"volumeMounts\": [
        {
          \"mountPath\": \"/var/run/secrets/kubernetes.io/serviceaccount\",
          \"name\": \"default-token-9z6jn\",
          \"readOnly\": true
        }
      ]
    }
  ]
}

```

```

    ]
  },
  "dnsPolicy": "ClusterFirst",
  "enableServiceLinks": true,
  "nodeName": "localhost",
  "priority": 0,
  "restartPolicy": "Always",
  "schedulerName": "default-scheduler",
  "securityContext": {},
  "serviceAccount": "default",
  "serviceAccountName": "default",
  "terminationGracePeriodSeconds": 30,
  "tolerations": [
    {
      "effect": "NoExecute",
      "key": "node.kubernetes.io/not-ready",
      "operator": "Exists",
      "tolerationSeconds": 300
    },
    {
      "effect": "NoExecute",
      "key": "node.kubernetes.io/unreachable",
      "operator": "Exists",
      "tolerationSeconds": 300
    }
  ],
  "volumes": [
    {
      "name": "default-token-9z6jn",
      "secret": {
        "defaultMode": 420,
        "secretName": "default-token-9z6jn"
      }
    }
  ]
},
"status": {
  "conditions": [
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:26:51Z",
      "status": "True",
      "type": "Initialized"
    },
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:27:03Z",
      "status": "True",
      "type": "Ready"
    },
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:27:03Z",
      "status": "True",
      "type": "ContainersReady"
    },
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:26:51Z",
      "status": "True",
      "type": "PodScheduled"
    }
  ],
  "containerStatuses": [
    {
      "containerID": "docker://virtlet.cloud__9c3d1706-a8ec-5732-4a40-46d564a08304",
      "image": "virtlet.cloud/sha256:ac5c214712c4bdb809e73bcefb199569b18f94dd5d3c464b0f35bec01cdcc24d",
      "imageID": "sha256:ac5c214712c4bdb809e73bcefb199569b18f94dd5d3c464b0f35bec01cdcc24d",

```

```

        "lastState": {},
        "name": "firewall",
        "ready": true,
        "restartCount": 0,
        "started": true,
        "state": {
            "running": {
                "startedAt": "2020-09-29T13:27:02Z"
            }
        }
    },
    "hostIP": "192.168.255.3",
    "phase": "Running",
    "podIP": "10.244.64.36",
    "podIPs": [
        {
            "ip": "10.244.64.36"
        }
    ],
    "qosClass": "Burstable",
    "startTime": "2020-09-29T13:26:51Z"
}
},
{
    "name": "test-release-packetgen-7557cb58bb-q2ms9",
    "GVK": {
        "Group": "",
        "Version": "",
        "Kind": ""
    },
    "status": {
        "metadata": {
            "annotations": {
                "VirtletCloudInitUserData": "ssh_pwauth: True\nusers:\n- name: admin\n  gecos: User\n  primary-group: admin\n  groups: users\n  sudo: ALL=(ALL) NOPASSWD:ALL\n  lock_passwd: false\n  passwd: \n$6$rounds=4096$QA50CKHTE41$jRACivoPMJcOjLRgx13t.AMfU7LhCFwOWv2z66CQX.TSxBy50JoYtycJXSP2JceG.8Tq\n/82QN9QYt3euYEZW/\n\nrunCmd:\n  - export demo_artifacts_version=1.5.0\n  - export vfw_private_ip_0=192.168.10.3\n  - export vsn_private_ip_0=192.168.20.3\n  - export protected_net_cidr=192.168.20.0/24\n  - export dcae_collector_ip=10.0.4.1\n  - export dcae_collector_port=8081\n  - export protected_net_gw=192.168.20.100\n/24\n  - export protected_private_net_cidr=192.168.10.0/24\n  - wget -O - https://git.onap.org/multicloud/k8s/plain/kud/tests/vFW/packetgen | sudo -E bash\n",
                "VirtletLibvirtCPUSetting": "mode: host-model\n",
                "VirtletRootVolumeSize": "5Gi",
                "app": "packetgen",
                "k8s.plugin.opnfv.org/nfn-network": "{ \"type\": \"ovn4nfv\", \"interface\": { \"name\": \"unprotected-private-net\", \"ipAddress\": \"192.168.10.2\", \"interface\": \"eth1\", \"defaultGateway\": \"false\" }, { \"name\": \"onap-private-net-test\", \"ipAddress\": \"10.0.100.2\", \"interface\": \"eth2\", \"defaultGateway\": \"false\" } ] }",
                "k8s.plugin.opnfv.org/ovnInterfaces": "[ { \"ip_address\": \"192.168.10.2/24\", \"mac_address\": \"00:00:00:88:d6:fe\", \"gateway_ip\": \"192.168.10.1\", \"defaultGateway\": \"false\", \"interface\": \"eth1\" }, { \"ip_address\": \"10.0.100.2/16\", \"mac_address\": \"00:00:00:e5:a5:ef\", \"gateway_ip\": \"10.10.0.1\", \"defaultGateway\": \"false\", \"interface\": \"eth2\" } ]",
                "k8s.v1.cni.cncf.io/networks": "[ { \"name\": \"ovn-networkobj\", \"namespace\": \"default\" } ]",
                "k8s.v1.cni.cncf.io/networks-status": "[ { \"name\": \"cni0\", \"interface\": \"virtlet-eth0\", \"ips\": [ \"10.244.64.35\" ], \"mac\": \"0a:58:0a:f4:40:23\", \"default\": true, \"dns\": { } }, { \"name\": \"ovn4nfv-k8s-plugin\", \"interface\": \"eth2\", \"ips\": [ \"192.168.10.2\", \"10.0.100.2\" ], \"mac\": \"00:00:00:e5:a5:ef\", \"dns\": { } } ]",
                "kubernetes.io/target-runtime": "virtlet.cloud",
                "release": "test-release"
            }
        },
        "creationTimestamp": "2020-09-29T13:26:51Z",
        "generateName": "test-release-packetgen-7557cb58bb-",
        "labels": {
            "app": "packetgen",
            "k8splugin.io/rb-instance-id": "stoic_wright",
            "pod-template-hash": "7557cb58bb",
            "release": "test-release"
        }
    }
}

```

```

    },
    "name": "test-release-packetgen-7557cb58bb-q2ms9",
    "namespace": "plugin-tests-namespace",
    "ownerReferences": [
      {
        "apiVersion": "apps/v1",
        "blockOwnerDeletion": true,
        "controller": true,
        "kind": "ReplicaSet",
        "name": "test-release-packetgen-7557cb58bb",
        "uid": "2ba50d08-d524-4d7f-ae63-9f172d98d38a"
      }
    ],
    "resourceVersion": "10719548",
    "selfLink": "/api/v1/namespaces/plugin-tests-namespace/pods/test-release-packetgen-7557cb58bb-q2ms9",
    "uid": "e9e935d5-8b92-4f8f-ae63-9f172d98d38a"
  },
  "spec": {
    "affinity": {
      "nodeAffinity": {
        "requiredDuringSchedulingIgnoredDuringExecution": {
          "nodeSelectorTerms": [
            {
              "matchExpressions": [
                {
                  "key": "extraRuntime",
                  "operator": "In",
                  "values": [
                    "virtlet"
                  ]
                }
              ]
            }
          ]
        }
      }
    },
    "containers": [
      {
        "image": "virtlet.cloud/ubuntu/16.04:latest",
        "imagePullPolicy": "IfNotPresent",
        "name": "packetgen",
        "resources": {
          "limits": {
            "memory": "4Gi"
          },
          "requests": {
            "memory": "4Gi"
          }
        },
        "stdin": true,
        "terminationMessagePath": "/dev/termination-log",
        "terminationMessagePolicy": "File",
        "tty": true,
        "volumeMounts": [
          {
            "mountPath": "/var/run/secrets/kubernetes.io/serviceaccount",
            "name": "default-token-9z6jn",
            "readOnly": true
          }
        ]
      }
    ],
    "dnsPolicy": "ClusterFirst",
    "enableServiceLinks": true,
    "nodeName": "localhost",
    "priority": 0,
    "restartPolicy": "Always",
    "schedulerName": "default-scheduler",
    "securityContext": {},

```

```

"serviceAccount": "default",
"serviceAccountName": "default",
"terminationGracePeriodSeconds": 30,
"tolerations": [
  {
    "effect": "NoExecute",
    "key": "node.kubernetes.io/not-ready",
    "operator": "Exists",
    "tolerationSeconds": 300
  },
  {
    "effect": "NoExecute",
    "key": "node.kubernetes.io/unreachable",
    "operator": "Exists",
    "tolerationSeconds": 300
  }
],
"volumes": [
  {
    "name": "default-token-9z6jn",
    "secret": {
      "defaultMode": 420,
      "secretName": "default-token-9z6jn"
    }
  }
]
},
"status": {
  "conditions": [
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:26:51Z",
      "status": "True",
      "type": "Initialized"
    },
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:27:02Z",
      "status": "True",
      "type": "Ready"
    },
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:27:02Z",
      "status": "True",
      "type": "ContainersReady"
    },
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:26:51Z",
      "status": "True",
      "type": "PodScheduled"
    }
  ],
  "containerStatuses": [
    {
      "containerID": "docker://virtlet.cloud__c3130190-d31f-5fa3-6c6b-bde8e0222aa2",
      "image": "virtlet.cloud/sha256:ac5c214712c4bdb809e73bcefb199569b18f94dd5d3c464b0f35bec01cdcc24d",
      "imageID": "sha256:ac5c214712c4bdb809e73bcefb199569b18f94dd5d3c464b0f35bec01cdcc24d",
      "lastState": {},
      "name": "packetgen",
      "ready": true,
      "restartCount": 0,
      "started": true,
      "state": {
        "running": {
          "startedAt": "2020-09-29T13:27:01Z"
        }
      }
    }
  ]
}

```

```

    }
    ],
    "hostIP": "192.168.255.3",
    "phase": "Running",
    "podIP": "10.244.64.35",
    "podIPs": [
        {
            "ip": "10.244.64.35"
        }
    ],
    "qosClass": "Burstable",
    "startTime": "2020-09-29T13:26:51Z"
}
},
{
    "name": "test-release-sink-766684c678-5rtgs",
    "GVK": {
        "Group": "",
        "Version": "",
        "Kind": ""
    },
    "status": {
        "metadata": {
            "annotations": {
                "k8s.plugin.opnfv.org/nfn-network": "{ \"type\": \"ovn4nfv\", \"interface\": [ { \"name\": \"protected-private-net\", \"ipAddress\": \"192.168.20.3\", \"interface\": \"eth1\", \"defaultGateway\": \"false\" }, { \"name\": \"onap-private-net-test\", \"ipAddress\": \"10.10.100.4\", \"interface\": \"eth2\", \"defaultGateway\": \"false\" } ] }, \"k8s.plugin.opnfv.org/ovnInterfaces\": \"[{ \"ip_address\": \"192.168.20.3/24\", \"mac_address\": \"00:00:00:44:1e:ab\", \"gateway_ip\": \"192.168.20.100\", \"defaultGateway\": \"false\", \"interface\": \"eth1\" }, { \"ip_address\": \"10.10.100.4/16\", \"mac_address\": \"00:00:00:f2:17:da\", \"gateway_ip\": \"10.10.0.1\", \"defaultGateway\": \"false\", \"interface\": \"eth2\" } ]\", \"k8s.v1.cni.cncf.io/networks\": \"[{ \"name\": \"ovn-networkobj\", \"namespace\": \"default\" } ]\", \"k8s.v1.cni.cncf.io/networks-status\": \"[{ \\n    \\\"name\\\": \\\"cni0\\\", \\n    \\\"interface\\\": \\\"eth0\\\", \\n    \\\"ips\\\": [ \\n        \\\"10.244.64.37\\\" \\n    ], \\n    \\\"mac\\\": \\\"0a:58:0a:f4:40:25\\\", \\n    \\\"default\\\": true, \\n    \\\"dns\\\": { } \\n }, { \\n    \\\"name\\\": \\\"ovn4nfv-k8s-plugin\\\", \\n    \\\"interface\\\": \\\"eth2\\\", \\n    \\\"ips\\\": [ \\n        \\\"192.168.20.3\\\", \\n        \\\"10.10.100.4\\\" \\n    ], \\n    \\\"mac\\\": \\\"00:00:00:f2:17:da\\\", \\n    \\\"dns\\\": { } \\n } ]\"
            },
            "creationTimestamp": "2020-09-29T13:26:51Z",
            "generateName": "test-release-sink-766684c678-",
            "labels": {
                "app": "sink",
                "k8splugin.io/rb-instance-id": "stoic_wright",
                "pod-template-hash": "766684c678",
                "release": "test-release"
            },
            "name": "test-release-sink-766684c678-5rtgs",
            "namespace": "plugin-tests-namespace",
            "ownerReferences": [
                {
                    "apiVersion": "apps/v1",
                    "blockOwnerDeletion": true,
                    "controller": true,
                    "kind": "ReplicaSet",
                    "name": "test-release-sink-766684c678",
                    "uid": "8eb55965-78b9-48e6-baa4-109d63af22b3"
                }
            ],
            "resourceVersion": "10719535",
            "selfLink": "/api/v1/namespaces/plugin-tests-namespace/pods/test-release-sink-766684c678-5rtgs",
            "uid": "99e70f0a-0a48-4308-ba27-a37d13465061"
        },
        "spec": {
            "containers": [
                {
                    "envFrom": [

```

```

        "configMapRef": {
            "name": "sink-configmap"
        }
    },
    "image": "rtsood/onap-vfw-demo-sink:0.2.0",
    "imagePullPolicy": "IfNotPresent",
    "name": "sink",
    "resources": {},
    "securityContext": {
        "privileged": true
    },
    "stdin": true,
    "terminationMessagePath": "/dev/termination-log",
    "terminationMessagePolicy": "File",
    "tty": true,
    "volumeMounts": [
        {
            "mountPath": "/var/run/secrets/kubernetes.io/serviceaccount",
            "name": "default-token-9z6jn",
            "readOnly": true
        }
    ]
},
{
    "image": "electrocucaracha/darkstat:latest",
    "imagePullPolicy": "IfNotPresent",
    "name": "darkstat",
    "ports": [
        {
            "containerPort": 667,
            "protocol": "TCP"
        }
    ],
    "resources": {},
    "stdin": true,
    "terminationMessagePath": "/dev/termination-log",
    "terminationMessagePolicy": "File",
    "tty": true,
    "volumeMounts": [
        {
            "mountPath": "/var/run/secrets/kubernetes.io/serviceaccount",
            "name": "default-token-9z6jn",
            "readOnly": true
        }
    ]
}
],
"dnsPolicy": "ClusterFirst",
"enableServiceLinks": true,
"nodeName": "localhost",
"priority": 0,
"restartPolicy": "Always",
"schedulerName": "default-scheduler",
"securityContext": {},
"serviceAccount": "default",
"serviceAccountName": "default",
"terminationGracePeriodSeconds": 30,
"tolerations": [
    {
        "effect": "NoExecute",
        "key": "node.kubernetes.io/not-ready",
        "operator": "Exists",
        "tolerationSeconds": 300
    },
    {
        "effect": "NoExecute",
        "key": "node.kubernetes.io/unreachable",
        "operator": "Exists",
        "tolerationSeconds": 300
    }
]
}

```

```

],
"volumes": [
  {
    "name": "default-token-9z6jn",
    "secret": {
      "defaultMode": 420,
      "secretName": "default-token-9z6jn"
    }
  }
]
},
"status": {
  "conditions": [
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:26:51Z",
      "status": "True",
      "type": "Initialized"
    },
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:27:00Z",
      "status": "True",
      "type": "Ready"
    },
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:27:00Z",
      "status": "True",
      "type": "ContainersReady"
    },
    {
      "lastProbeTime": null,
      "lastTransitionTime": "2020-09-29T13:26:51Z",
      "status": "True",
      "type": "PodScheduled"
    }
  ],
  "containerStatuses": [
    {
      "containerID":
"docker://e63bae29d79a82c9c1ba2cf6038ec439c7e5fbc1fe584a0d721cfaab36ea3816",
      "image": "electrocucaracha/darkstat:latest",
      "imageID": "docker-pullable://electrocucaracha/darkstat@sha256:
a6764fcc2e15f6156ac0e56f1d220b98970f2d4da9005bae99fb518cfd2f9c25",
      "lastState": {},
      "name": "darkstat",
      "ready": true,
      "restartCount": 0,
      "started": true,
      "state": {
        "running": {
          "startedAt": "2020-09-29T13:26:59Z"
        }
      }
    },
    {
      "containerID":
"docker://696a651658d9a199cced1982cb9b629607dde899d6af5bddb12c2b2232196347",
      "image": "rtsood/onap-vfw-demo-sink:0.2.0",
      "imageID": "docker-pullable://rtsood/onap-vfw-demo-sink@sha256:
15b7abb0b67a3804ea5f954254633f996fc99c680b09d86a6cf15c3d7b14ab16",
      "lastState": {},
      "name": "sink",
      "ready": true,
      "restartCount": 0,
      "started": true,
      "state": {
        "running": {
          "startedAt": "2020-09-29T13:26:59Z"
        }
      }
    }
  ]
}

```


Configuration API

Configuration Template API

Example Contents of Tar File

```
> cd demo/vnfs/DaaS
> find kafka-config
kafka-config/values.yaml
kafka-config/Chart.yaml
kafka-config/templates
kafka-config/templates/topic.yaml

#Create a tar.gz for upload
> tar -cf kafka-config.tar kafka-config
> gzip kafka-config.tar
```

Configuration Values API

In Day 2 apply configuration API each instance of configuration is identified by `rb_name`, `rb_version` and `profile_name`. The body of the API contains set of parameter and value list.

Command to **POST** (create) Configuration Values

```
curl -i -d @values.json -X POST http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/{name}/{version}/profile/{name}/config
```

With the following JSON content (values.json)

```
{
  "template-name": "kafka_token",
  "config-name": "kafka_token_1",
  "description": "testing Kafka Day 2 config configuration",
  "values": {
    "namespace": "kafka",
    "topic": {
      "name": "orders",
      "cluster": "my-cluster",
      "partitions": 10,
      "replicas": 2
    }
  }
}
```

This command returns the following JSON which contains config-version id.

```
{
  "rb-name": "kafka",
  "rb-version": "kafka-config-1",
  "profile-name": "1234567890",
  "template-name": "kafka-token",
  "config-name": "kafka_token_1",
  "config-version": "1"
}
```

Command to **GET** Configuration Values

```
# Get all Values [DOES NOT WORK]
curl -i http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/{name}/{version}/profile/{name}/config

# Get one Value with config name
curl -i http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/{name}/{version}/profile/{name}/config/{name}
```

Command to **DELETE** Configuration Values

```
curl -i -X DELETE http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/{name}/{version}/profile/{name}/config/{name}
```

Command to **PUT** (Modify) Configuration Values

```
curl -i -d @values.json -X PUT http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/{name}/{version}/profile/{name}/config/{name}
```

With the following JSON content (values.json)

```
{
  "template-name": "kafka_token"
  "config-name": "kafka_token_1"
  "description": "testing Kafka Day 2 config configuration",
  "values": {
    "namespace": "kafka",
    "topic": {
      "name": "orders",
      "cluster": "my-cluster",
      "partitions": 20,
      "replicas": 2
    }
  }
}
```

This command returns the following JSON which contains config-version.

```
{
  "rb-name": "kafka",
  "rb-version": "kafka-config-1",
  "profile-name": "1234567890",
  "template-name": "kafka-token"
  "config-name": "kafka_token_1"
  "config-version": "3"
}
```

Command to **TAG** configuration Values

This will add the tag name to the current config version

```
curl -i -d @values.json -X POST http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/{name}/{version}/profile/{name}/tagit
```

With the following JSON content (values.json)

```
{
  "tag-name": "my-tag"
}
```

Command to **ROLLBACK** configuration Values

Rollbacks configuration to a config version or a tag.

```
curl -i -d @values.json -X POST http://NODE_IP:30280/api/multicloud-k8s/v1/v1/rb/definition/{name}/{version}/profile/{name}/rollback
```

With the following JSON content (values.json)

```
{
  "anyOf": [
    {
      "config-version": "<value>"
    },
    {
      "config-tag": "<tag name>"
    }
  ]
}
```

Reachability/Connectivity Info API

API to support Reachability for Kubernetes Cloud

Command to **POST** Connectivity Info

```
{
  "cloud-region" : "<name>",    // Must be unique across
  "cloud-owner" : "<owner>",
  "other-connectivity-list" : {
    //Extendible list of name value pairs
    "connectivity-records" : [
      {
        "connectivity-record-name" : "<name>",    // example: OVN
        "FQDN-or-ip" : "<fqdn>",
        "ca-cert-to-verify-server" : "<contents of CA certificate to validate the OVN server>",
        "ssl-initiator" : "<true/false>",
        "user-name": "<user name>",    //valid if ssl-initiator is false
        "password" : "<password>",    // valid if ssl-initiator is false
        "private-key" : "<contents of private key in PEM>", // valid if ssl-initiator is true
        "cert-to-present" : "<contents of certificate to present to server>" , //valid if ssl-
initiator is true
      },
    ]
  }
}
```

This is a multipart upload and here is how you do the POST for this.

```
#Using a json file containing content as above;

curl -i -F "metadata=<jsonfile;type=application/json" -F file=@/home/ad_kkkamine/.kube/config -X POST http://NOD
E_IP:30280/api/multicloud-k8s/v1/v1/connectivity-info
```

Command to GET Connectivity Info

```
curl -i -X GET http://NODE_IP:30280/api/multicloud-k8s/v1/v1/connectivity-info/{name}
```

Command to DELETE Connectivity Info

```
curl -i -X DELETE http://NODE_IP:30280/api/multicloud-k8s/v1/v1/connectivity-info/{name}
```

Command to UPDATE/PUT Connectivity Info

```
curl -i -d @update.json -X PUT http://NODE_IP:30280/api/multicloud-k8s/v1/v1/connectivity-info/{name}
```

Infrastructure workload LCM API

APIs defined here conform to API defined by main multicloud broker service: <https://docs.onap.org/en/dublin/submodules/multicloud/framework.git/docs/MultiCloud-APIv1-Specification.html#infrastructure-workload-lcm>

Example request toward "Instantiate" API can look as follows:

```
POST http://msb-iag.onap:80/api/multicloud/v1/k8scloudowner4/k8sregionfour/infra_workload

{
  "generic-vnf-id": "6bfca5dc-993d-48f1-ad27-a7a9ea91836b",
  "oof_directives": {},
  "sdnc_directives": {
    "attributes": [
      {
        "attribute_name": "k8s-rb-profile-namespace",
        "attribute_value": "default"
      },
      {
        "attribute_name": "vnf_model_customization_uuid",
        "attribute_value": "366c007e-7684-4a0b-a2f4-9815174bec55"
      },
      {
        "attribute_name": "vfw_int_private2_ip_0",
```

```
    "attribute_value": "192.168.20.10"
  },
  {
    "attribute_name": "int_private1_subnet_id",
    "attribute_value": "6bfca5dc-993d-48f1-ad27-a7a9ea91836b-unprotected-network-subnet-1"
  },
  {
    "attribute_name": "public_net_id",
    "attribute_value": "est"
  },
  {
    "attribute_name": "onap_private_subnet_id",
    "attribute_value": "6bfca5dc-993d-48f1-ad27-a7a9ea91836b-namagement-network-subnet-1"
  },
  {
    "attribute_name": "vsn_int_private2_ip_0",
    "attribute_value": "192.168.20.11"
  },
  {
    "attribute_name": "aic-cloud-region",
    "attribute_value": "k8sregionfour"
  },
  {
    "attribute_name": "int_private1_gw_ip",
    "attribute_value": "192.168.10.11"
  },
  {
    "attribute_name": "sec_group",
    "attribute_value": "culpa"
  },
  {
    "attribute_name": "nfc-naming-code",
    "attribute_value": "base_template"
  },
  {
    "attribute_name": "onap_private_net_cidr",
    "attribute_value": "10.0.0.0/8"
  },
  {
    "attribute_name": "nexus_artifact_repo",
    "attribute_value": "https://nexus.onap.org"
  },
  {
    "attribute_name": "vpg_onap_private_ip_0",
    "attribute_value": "10.0.101.10"
  },
  {
    "attribute_name": "image_name",
    "attribute_value": "voluptatem"
  },
  {
    "attribute_name": "int_private2_gw_ip",
    "attribute_value": "192.168.20.9"
  },
  {
    "attribute_name": "flavor_name",
    "attribute_value": "ducimus"
  },
  {
    "attribute_name": "onap_private_gw_ip",
    "attribute_value": "10.0.101.9"
  },
  {
    "attribute_name": "vfw_int_private2_floating_ip",
    "attribute_value": "192.168.10.13"
  },
  {
    "attribute_name": "vf-module-model-invariant-uuid",
    "attribute_value": "750b39d0-7f99-4b7f-9a22-c15c7348221d"
  },
  {
    
```

```

    "attribute_name": "int_private2_net_cidr",
    "attribute_value": "192.168.20.0/24"
  },
  {
    "attribute_name": "int_private2_subnet_id",
    "attribute_value": "6bfca5dc-993d-48f1-ad27-a7a9ea91836b-protected-network-subnet-1"
  },
  {
    "attribute_name": "vf-naming-policy",
    "attribute_value": "SDNC_Policy.Config_MS_ONAP_VNF_NAMING_TIMESTAMP"
  },
  {
    "attribute_name": "net_attachment_definition",
    "attribute_value": "6bfca5dc-993d-48f1-ad27-a7a9ea91836b-ovn-nat"
  },
  {
    "attribute_name": "vf-module-model-version",
    "attribute_value": "8bb9fa50-3e82-4664-bd1c-a29267be726a"
  },
  {
    "attribute_name": "k8s-rb-profile-name",
    "attribute_value": "vfw-cnf-cds-base-profile"
  },
  {
    "attribute_name": "onap_private_net_id",
    "attribute_value": "6bfca5dc-993d-48f1-ad27-a7a9ea91836b-management-network"
  },
  {
    "attribute_name": "vf_module_label",
    "attribute_value": "base_template"
  },
  {
    "attribute_name": "service-instance-id",
    "attribute_value": "0362acff-38e7-4ecc-8ac0-4780161f3ca0"
  },
  {
    "attribute_name": "sdnc_artifact_name",
    "attribute_value": "vnf"
  },
  {
    "attribute_name": "demo_artifacts_version",
    "attribute_value": "et"
  },
  {
    "attribute_name": "aai-vf-module-put",
    "attribute_value": "SUCCESS"
  },
  {
    "attribute_name": "test.param",
    "attribute_value": "testValue"
  },
  {
    "attribute_name": "pub_key",
    "attribute_value": "aut"
  },
  {
    "attribute_name": "vnf-model-customization-uuid",
    "attribute_value": "366c007e-7684-4a0b-a2f4-9815174bec55"
  },
  {
    "attribute_name": "vfw_int_private1_ip_0",
    "attribute_value": "192.168.10.12"
  },
  {
    "attribute_name": "key_name",
    "attribute_value": "aut"
  },
  {
    "attribute_name": "sdnc_model_version",
    "attribute_value": "1.0.7"
  },

```

```

        {
            "attribute_name": "int_private1_net_cidr",
            "attribute_value": "192.168.10.0/24"
        },
        {
            "attribute_name": "install_script_version",
            "attribute_value": "suscipit"
        },
        {
            "attribute_name": "vf_module_type",
            "attribute_value": "Base"
        },
        {
            "attribute_name": "vpg_int_private1_ip_0",
            "attribute_value": "192.168.10.14"
        },
        {
            "attribute_name": "int_private2_net_id",
            "attribute_value": "6bfca5dc-993d-48f1-ad27-a7a9ea91836b-protected-network"
        },
        {
            "attribute_name": "sdnc_model_name",
            "attribute_value": "vFW_CNF_CDS"
        },
        {
            "attribute_name": "cloud_env",
            "attribute_value": "openstack"
        },
        {
            "attribute_name": "int_private1_net_id",
            "attribute_value": "6bfca5dc-993d-48f1-ad27-a7a9ea91836b-unprotected-network"
        }
    ]
},
"template_data": {
    "disable_rollback": true,
    "environment": "#snipped for readability",
    "parameters": {
        "int_private1_net_cidr": "192.168.10.0/24",
        "int_private2_net_cidr": "192.168.20.0/24",
        "pub_key": "aut",
        "vf_module_id": "34c190c7-e5bc-4e61-a0d9-5fd44416dd96",
        "vnf_id": "6bfca5dc-993d-48f1-ad27-a7a9ea91836b",
        "vnf_name": "FixedVFW 0"
    },
    "stack_name": "SharedPostman-21-base_template",
    "template": "#snipped for readability",
    "timeout_mins": 120
},
"template_type": "",
"user_directives": {},
"vf-module-id": "34c190c7-e5bc-4e61-a0d9-5fd44416dd96",
"vf-module-model-customization-id": "603eadfe-50d6-413a-853c-46f5a8e2ddc7",
"vf-module-model-invariant-id": "750b39d0-7f99-4b7f-9a22-c15c7348221d",
"vf-module-model-version-id": "8bb9fa50-3e82-4664-bd1c-a29267be726a"
}

```

Please note that following attributes from *_directives have special meaning: **"k8s-rb-profile-name"** (mandatory) and **"k8s-rb-instance-release-name"** (optional). They correspond to **"profile-name"** and **"release-name"** parameters from instance API.