KubeCon | CloudNativeCon

Europe 2020

Virtual

Live Migration of Production Workloads from Apache Mesos PaaS to Kubernetes

Maria Camacho & Gufran Lutful, Nokia

Who we are

Maria Camacho

Gufran Lutful

"A picture is worth a thousand words"

Nokia has a comprehensive portfolio of network equipment, software, services and licensing opportunities across the globe for communications service providers.

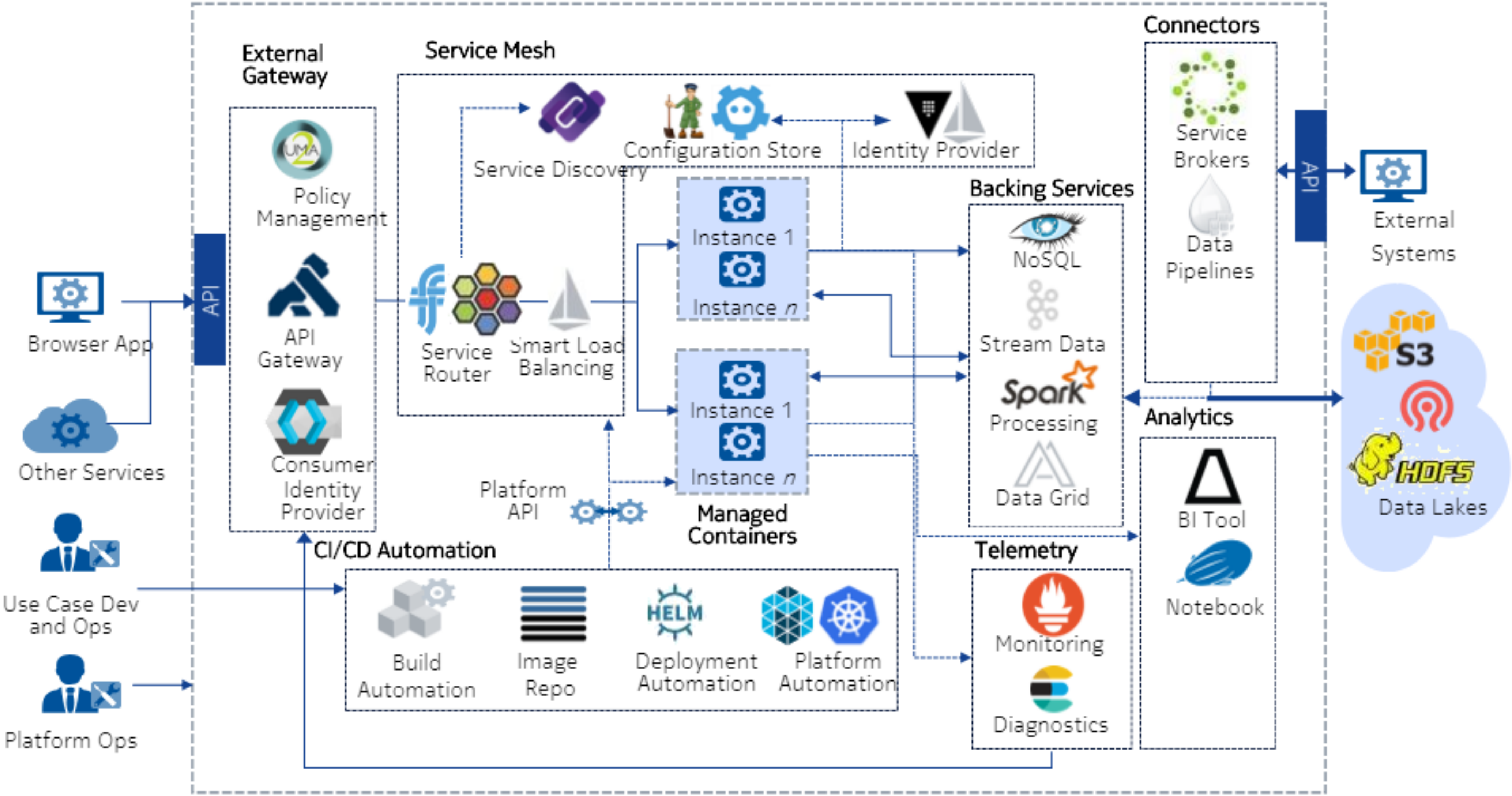
With its commitment to innovation, driven by the award-winning Nokia Bell Labs, Nokia is a leader in the development and deployment of 5G networks.

Nokia is still connecting people ;)

The story

©priestmangoode

About the project



Adoption of Kubernetes

By 2018...

Kubernetes has become:

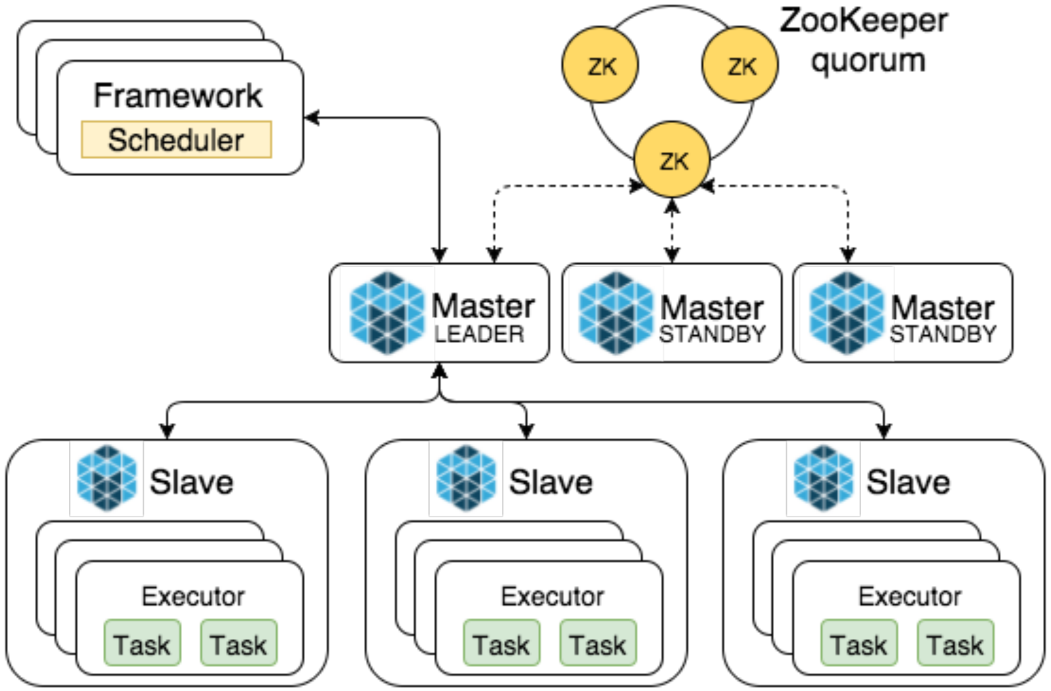
- ✓ Industry leading container orchestrator
- ✓ One of the top projects on GitHub: in a top position in stars, and No. 1 in terms of activity
- ✓ The centre of a growing community
- ✓ Quickly reaching production-level maturity

But there were limitations too:

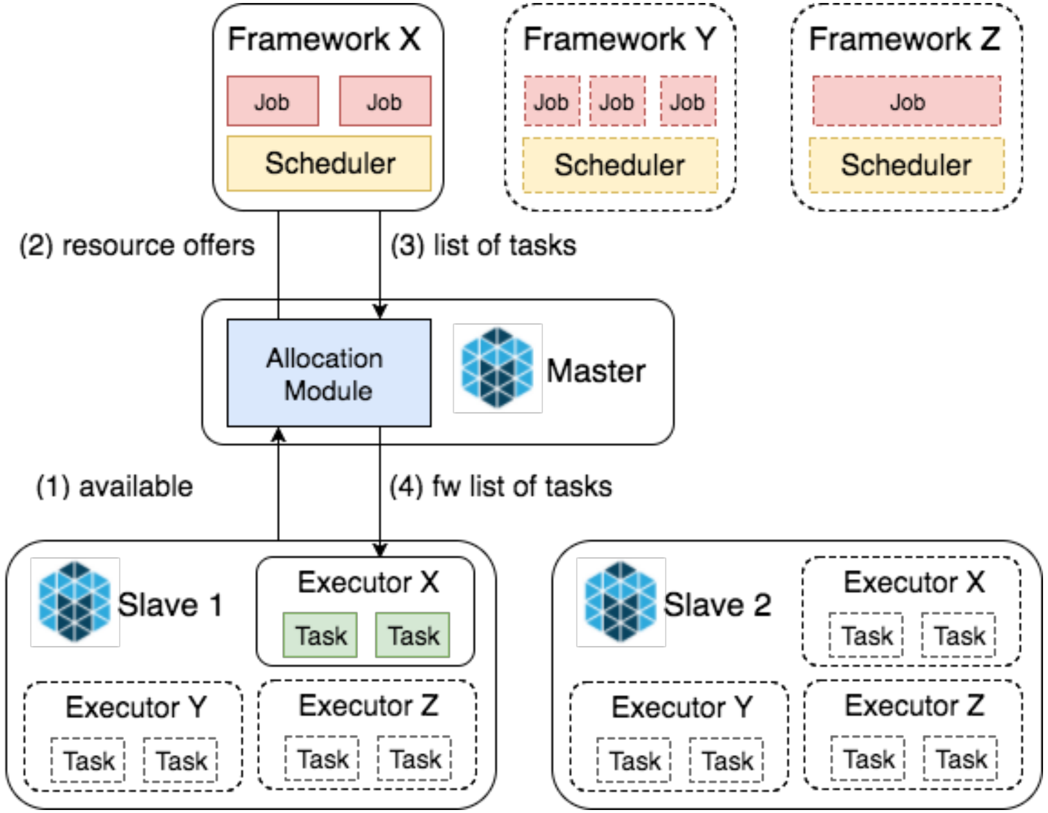
- ✓ Hard to run big data workloads with Apache Spark
- ✓ Not possible to seamlessly manage LPVs

Mesos Overview

Architecture

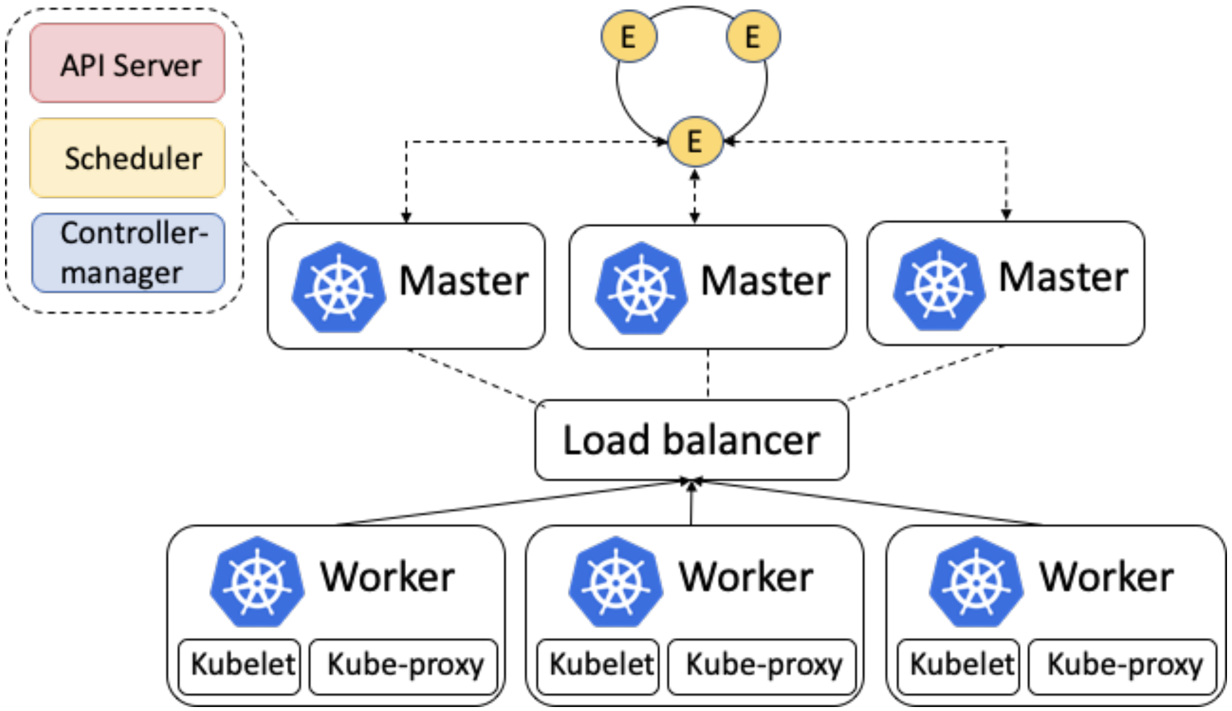


Scheduling

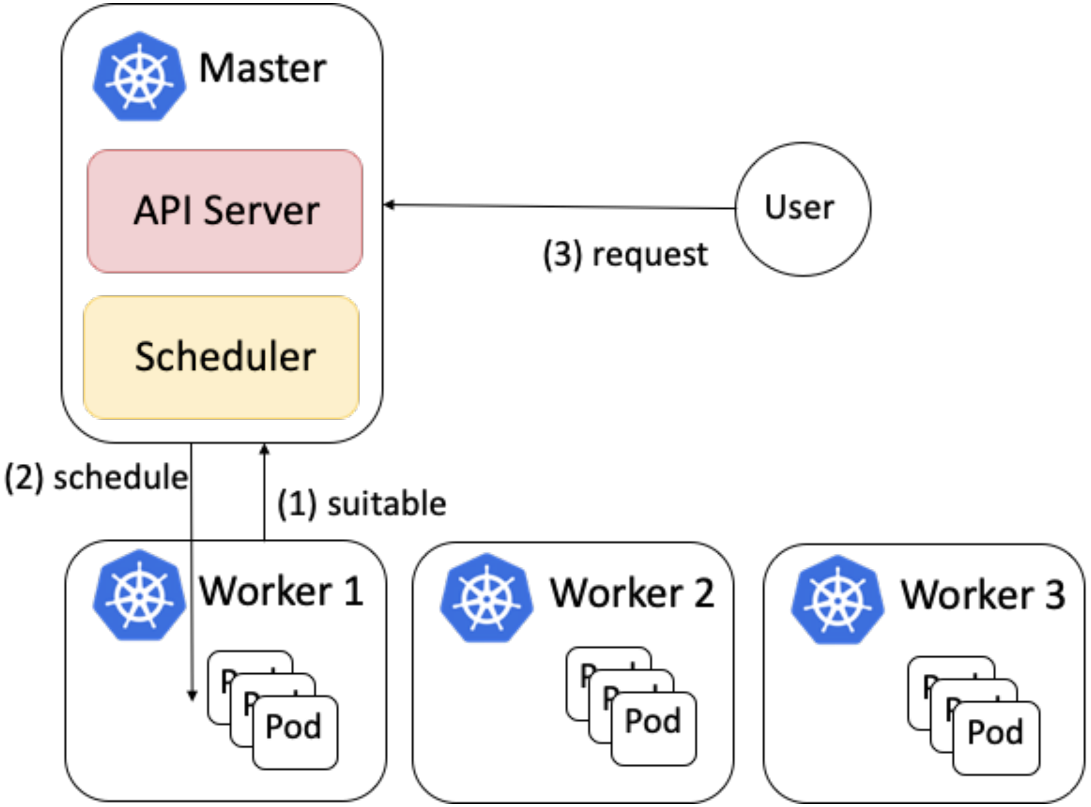


Kubernetes Overview

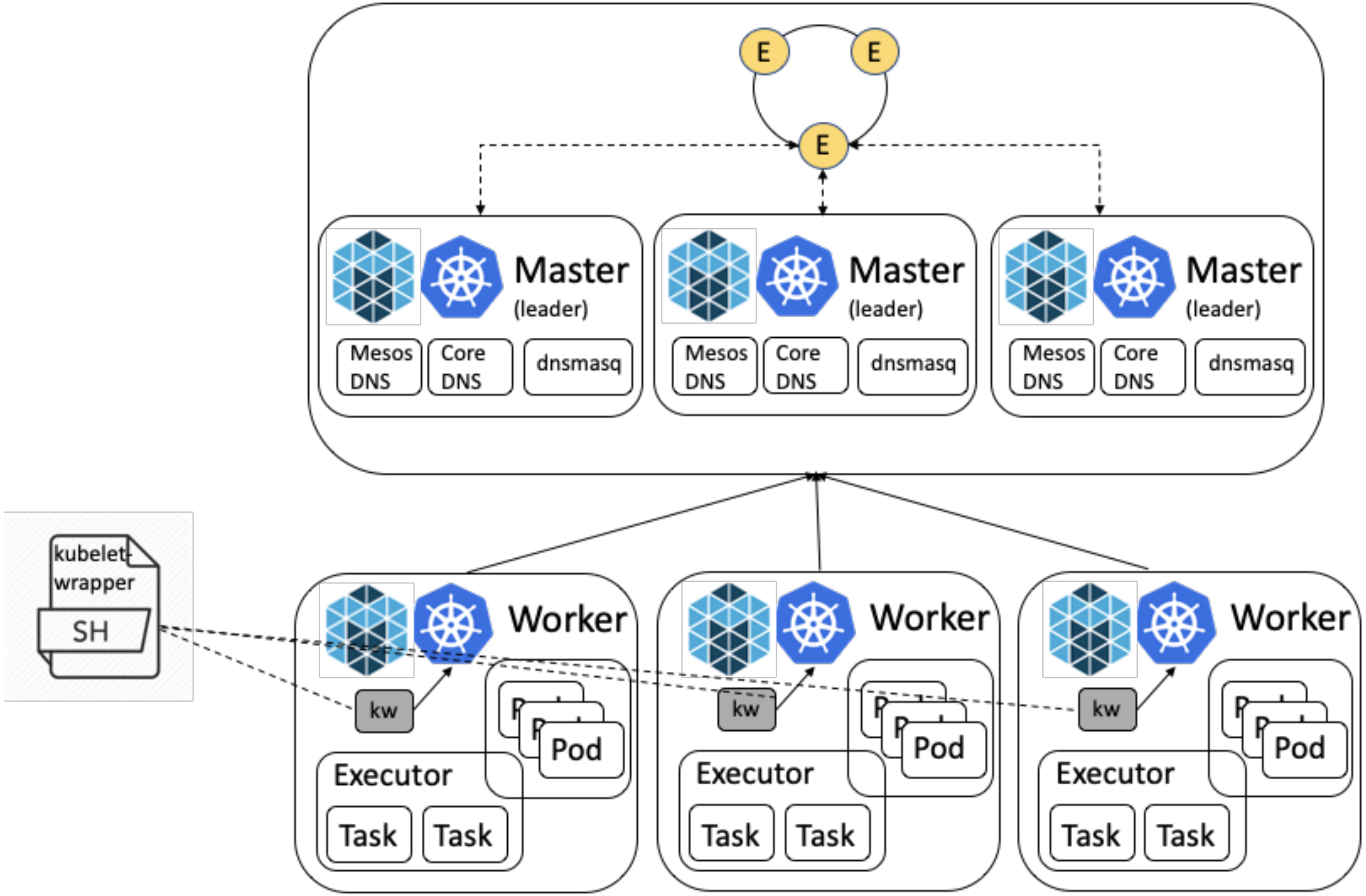
Architecture



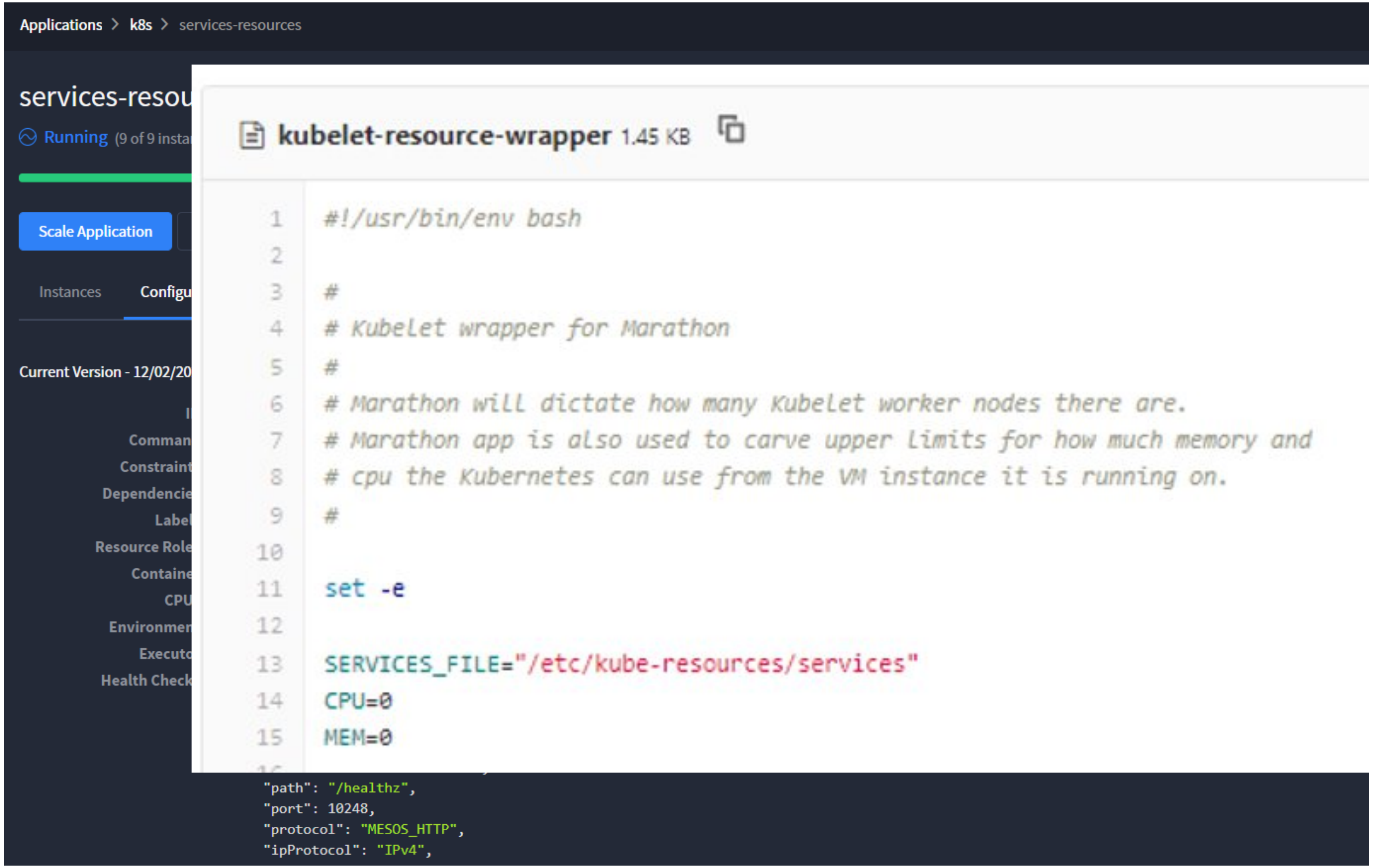
Scheduling



Mesos & Kubernetes Together



Kubelet Wrapper in Marathon



Applications > k8s > services-resources

services-resou

Running (9 of 9 insta

Scale Application

Instances Configu

Current Version - 12/02/20

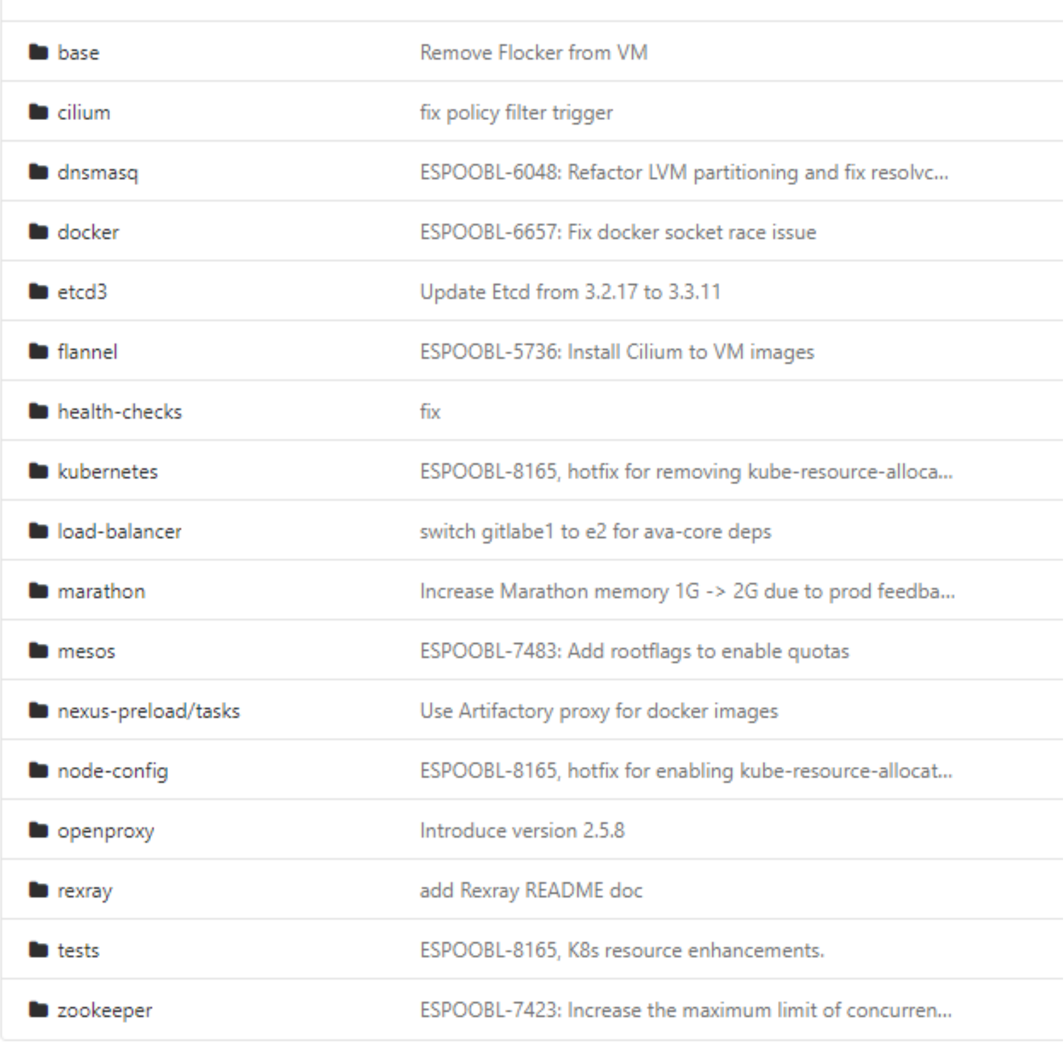
I
Comman
Constraint
Dependencie
Label
Resource Role
Containe
CPU
Environmen
Executo
Health Check

kubelet-resource-wrapper 1.45 KB

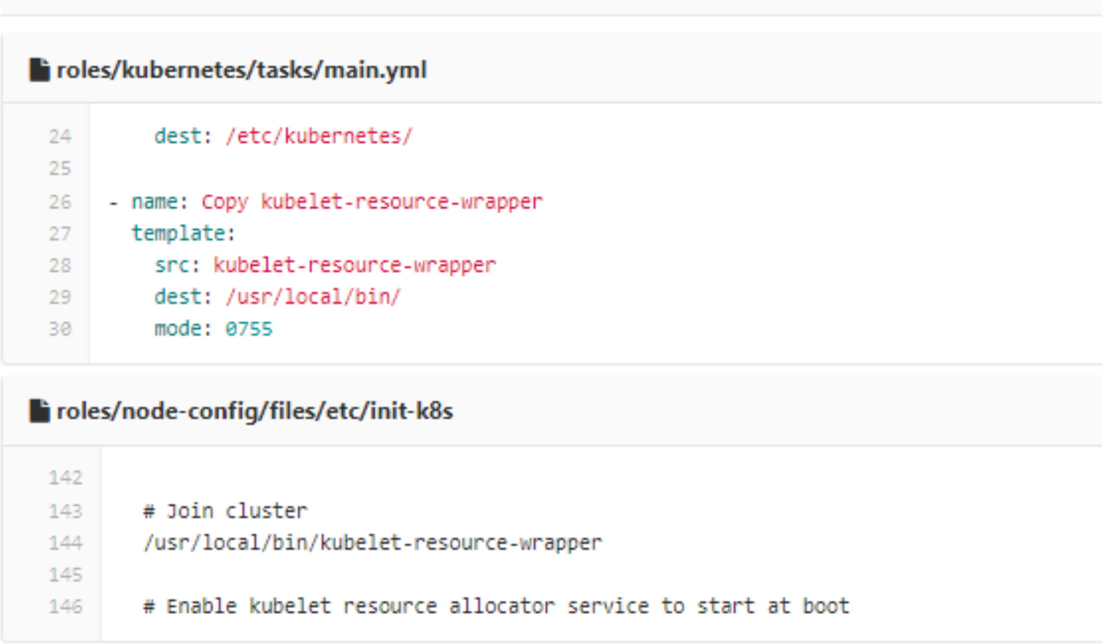
```
#!/usr/bin/env bash

#
# Kubelet wrapper for Marathon
#
# Marathon will dictate how many Kubelet worker nodes there are.
# Marathon app is also used to carve upper limits for how much memory and
# cpu the Kubernetes can use from the VM instance it is running on.
#

set -e

SERVICES_FILE="/etc/kube-resources/services"
CPU=0
MEM=0
```

```
"path": "/healthz",
"port": 10248,
"protocol": "MESOS_HTTP",
"ipProtocol": "IPv4",
```

Kubelet Wrapper in VM-Image



base	Remove Flocker from VM
cilium	fix policy filter trigger
dnsmasq	ESPOOBL-6048: Refactor LVM partitioning and fix resolvc...
docker	ESPOOBL-6657: Fix docker socket race issue
etcd3	Update Etcd from 3.2.17 to 3.3.11
flannel	ESPOOBL-5736: Install Cilium to VM images
health-checks	fix
kubernetes	ESPOOBL-8165, hotfix for removing kube-resource-alloca...
load-balancer	switch gitlabe1 to e2 for ava-core deps
marathon	Increase Marathon memory 1G -> 2G due to prod feedba...
mesos	ESPOOBL-7483: Add rootflags to enable quotas
nexus-preload/tasks	Use Artifactory proxy for docker images
node-config	ESPOOBL-8165, hotfix for enabling kube-resource-allocat...
openproxy	Introduce version 2.5.8
rexray	add Rexray README doc
tests	ESPOOBL-8165, K8s resource enhancements.
zookeeper	ESPOOBL-7423: Increase the maximum limit of concurren...



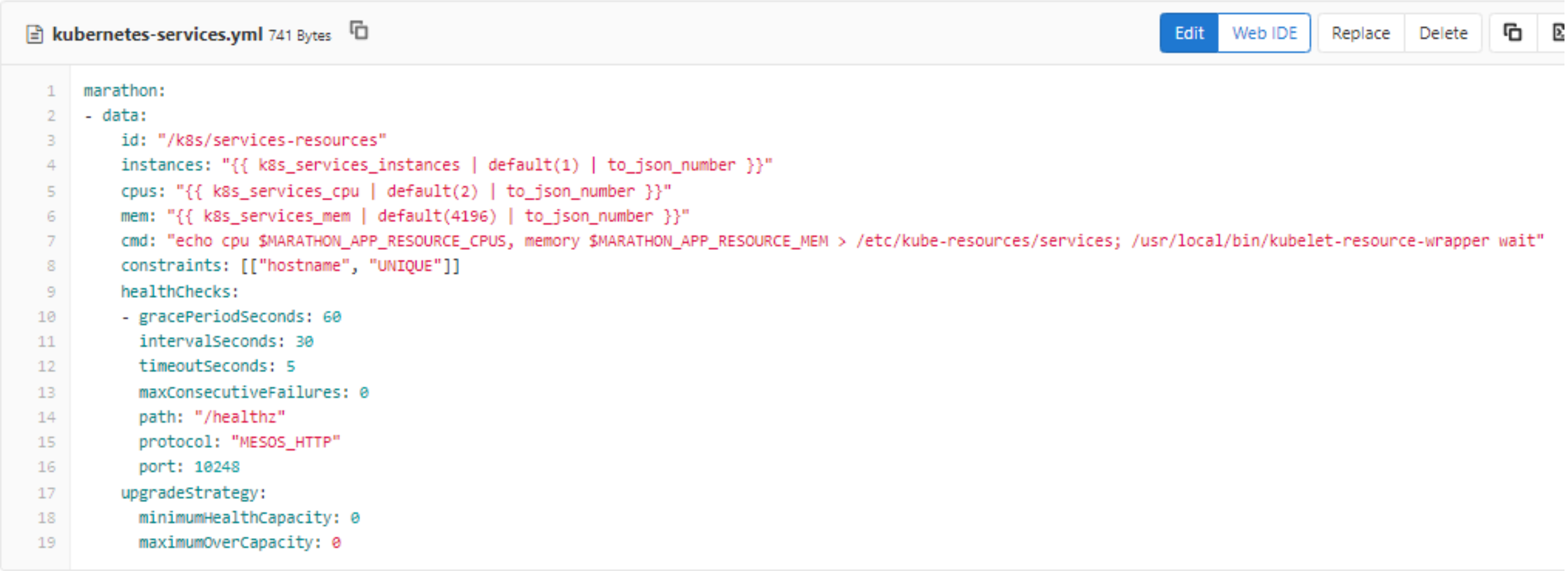
roles/kubernetes/tasks/main.yml

```yaml
    dest: /etc/kubernetes/

- name: Copy kubelet-resource-wrapper
  template:
    src: kubelet-resource-wrapper
    dest: /usr/local/bin/
    mode: 0755
```

roles/node-config/files/etc/init-k8s

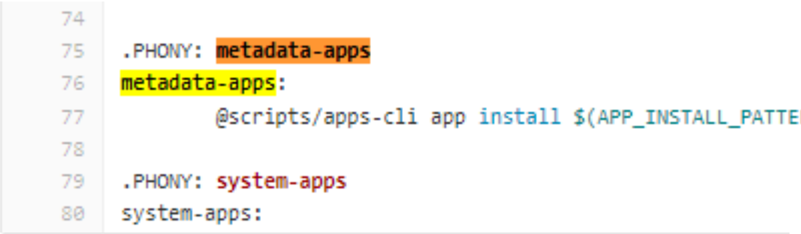
```
# Join cluster
/usr/local/bin/kubelet-resource-wrapper

# Enable kubelet resource allocator service to start at boot
```

Launching Kubelet Wrapper in Marathon

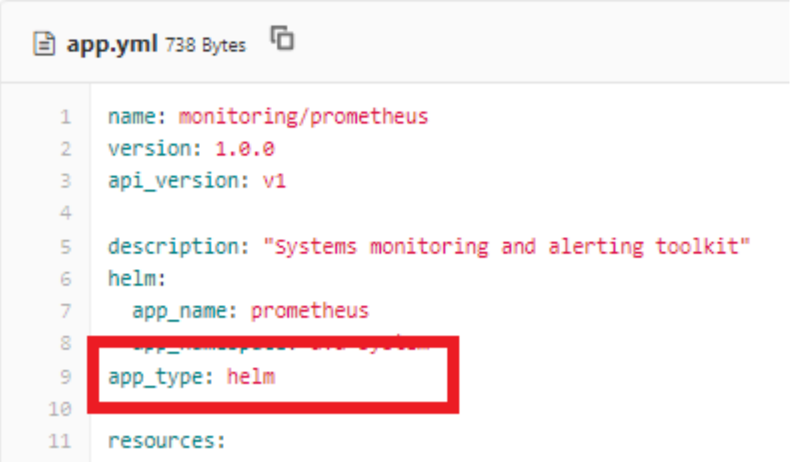


```yaml
marathon:
- data:
    id: "/k8s/services-resources"
    instances: "{{ k8s_services_instances | default(1) | to_json_number }}"
    cpus: "{{ k8s_services_cpu | default(2) | to_json_number }}"
    mem: "{{ k8s_services_mem | default(4196) | to_json_number }}"
    cmd: "echo cpu $MARATHON_APP_RESOURCE_CPUS, memory $MARATHON_APP_RESOURCE_MEM > /etc/kube-resources/services; /usr/local/bin/kubelet-resource-wrapper wait"
    constraints: [["hostname", "UNIQUE"]]
    healthChecks:
    - gracePeriodSeconds: 60
      intervalSeconds: 30
      timeoutSeconds: 5
      maxConsecutiveFailures: 0
      path: "/healthz"
      protocol: "MESOS_HTTP"
      port: 10248
    upgradeStrategy:
      minimumHealthCapacity: 0
      maximumOverCapacity: 0
```

Leveraging Existing Metadata Driven Deployment

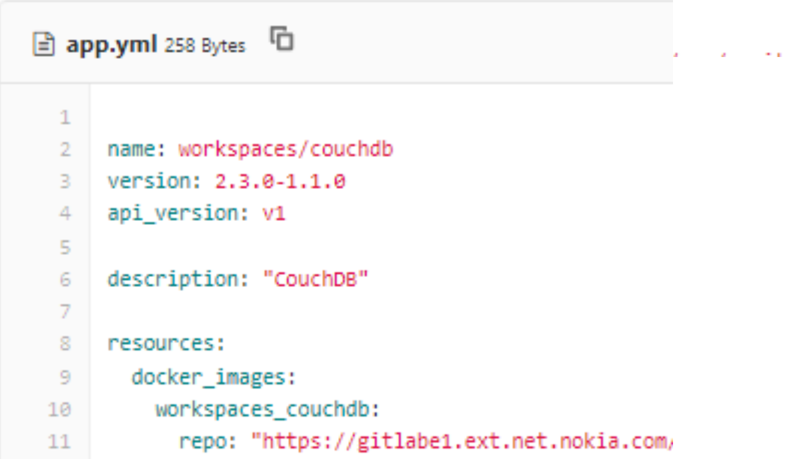


```
.PHONY: metadata-apps
metadata-apps:
	@scripts/apps-cli app install $(APP_INSTALL_PATTERN)

.PHONY: system-apps
system-apps:
```



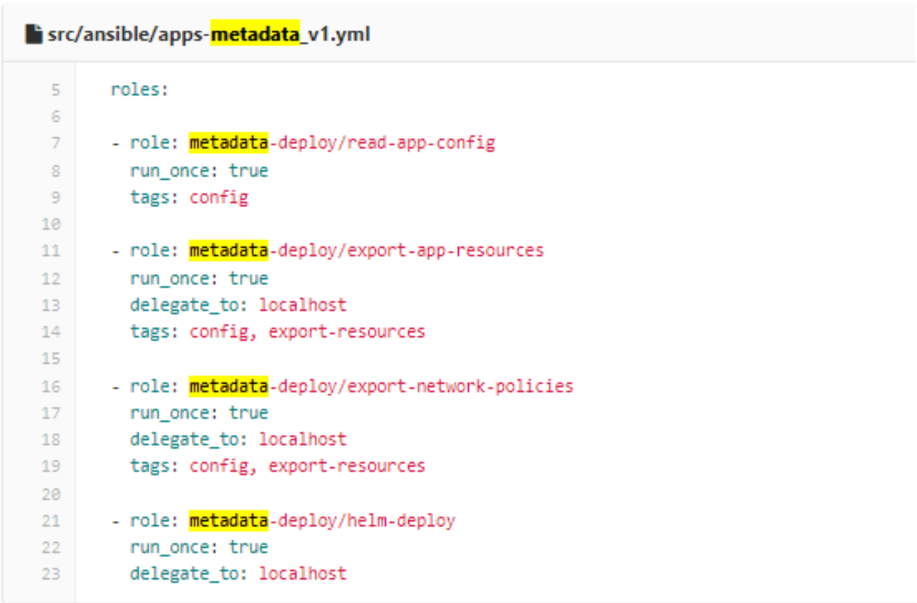
app.yml 738 Bytes

```
name: monitoring/prometheus
version: 1.0.0
api_version: v1

description: "Systems monitoring and alerting toolkit"
helm:
  app_name: prometheus
  app_namespace: kube-system
app_type: helm

resources:
```



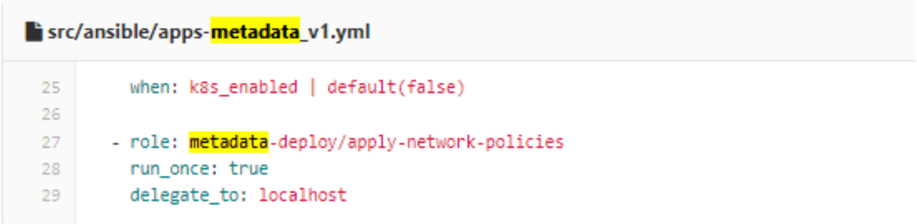
app.yml 258 Bytes

```

name: workspaces/couchdb
version: 2.3.0-1.1.0
api_version: v1

description: "CouchDB"

resources:
  docker_images:
    workspaces_couchdb:
      repo: "https://gitlabe1.ext.net.nokia.com/
```



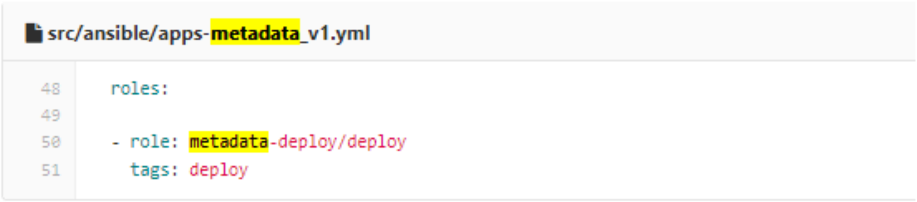
src/ansible/apps-metadata_v1.yml

```
roles:

- role: metadata-deploy/read-app-config
  run_once: true
  tags: config

- role: metadata-deploy/export-app-resources
  run_once: true
  delegate_to: localhost
  tags: config, export-resources

- role: metadata-deploy/export-network-policies
  run_once: true
  delegate_to: localhost
  tags: config, export-resources

- role: metadata-deploy/helm-deploy
  run_once: true
  delegate_to: localhost
```



src/ansible/apps-metadata_v1.yml

```
  when: k8s_enabled | default(false)

- role: metadata-deploy/apply-network-policies
  run_once: true
  delegate_to: localhost
```



src/ansible/apps-metadata_v1.yml

```
roles:

- role: metadata-deploy/deploy
  tags: deploy
```

Lessons learnt

The strategy:

- ✓ Spike to study the possible options of migration
- ✓ Follow KISS principle
- ✓ Less is more
- ✓ Favour a release-driven migration
- ✓ Have proper documentation/guidelines for dev teams
- ✓ Have a rollback strategy

The implementation:

- ✓ Metadata driven deployment
- ✓ K8s and Mesos can share same host resources
- ✓ Dimension your cluster properly, including system resources
- ✓ Use dedicated CIDRs for each orchestrator
- ✓ Kubelet can be run with no resources. Required for pod eviction

The benefits:

- ✓ Seamless sharing of resources between orchestrators
- ✓ Hosting selected workloads on each orchestrator
- ✓ Managing network traffic between orchestrators
- ✓ Internal DNS sharing
- ✓ Independent block storage management

…and much more

Bonus info

Some tools we used and loved:

GitLab

cilium

ANSIBLE

CI/CD

Grafana

Flocker™ by ClusterHQ™

HELM

KONG

KEYCLOAK

elasticsearch

Prometheus

Apache MESOS™

docker

HashiCorp Packer

kubernetes

swagger

etcd

Apache Zeppelin

cassandra

logstash

kibana

Spark

HashiCorp Vault

Nexus

HashiCorp Terraform

Jenkins

REX-Ray

NOKIA

KubeCon | CloudNativeCon

Europe 2020

Virtual

KEEP CLOUD NATIVE CONNECTED