Highly available, Production Ready
Blueprint for OSM
Mark Beierl
Canonical
Fault Domains

Single Point of Failure
Fault Domains
Software Fault Domains

- Kubernetes
- Databases
- OSM Core Software
- OSM VCA
K8s Cluster Domain

https://ubuntu.com/kubernetes/docs/high-availability
K8s Cluster Domain

Kubernetes Master

etcd

etcd

etcd

Kubernetes Master

etcd

etcd

Kubernetes Master
OSM Architecture

Integrated components for policy, fault, performance management and placement

E2E orchestration through Life Cycle Manager (LCM)

Unified Northbound Interface (SOL005-based), decoupled from LCM

Complete control through CLI and UI

Execution environments for VNF configuration

Data Fault Domain

Common Services

OSM IM
Common Database (NoSQL)
Object Storage
TSDB (Metrics)
Auth

Common Services

osmclient
GUI (ngui)

NBI

OSM IM
LCM
VCA
RO

Kafka bus

PLA
POL
MON

Message bus for async communications

Common DB, auth and object storage

Integrated components for policy, fault, performance management and placement

Execution environments for VNF configuration
Data Domain

Datacentre #1
- Common Database (MongoDB)
  - Voting, leader elected
  - Replica Sets
  - Common Database (MongoDB)
  - Common Database (MongoDB)

Datacentre #2
- Common Database (MongoDB)
  - No votes, no leaders
  - (Standby, not started)
  - Common Database (MongoDB)
  - Common Database (MongoDB)
  - Common Database (MongoDB)
OSM architecture

Integrated components for policy, fault, performance management and placement

E2E orchestration through Life Cycle Manager (LCM)

Unified Northbound Interface (SOL005-based), decoupled from LCM

Complete control through CLI and UI

Execution environments for VNF configuration
OSM Software Replicas

Ingress Controller

Calico VLAN

NBI/0

NBI/1

NBI/2

Single Point of Failure
Distributed VCA

Datacentre #1

Open Source MANO

LCM

N2VC

Single Point of Failure

Datacentre #2
High Available Charms

OSM

LXD

failed node

newly elected leader

LXD

HA CLUSTER

LXD

© ETSI
Resiliency in exactly the same way for K8s Charms

Use K8s native mechanisms for storage, resiliency
• No default password
• Random passwords are generated as 16 hexadecimal digit string for each installation
• Private keys are encrypted with symmetric AES before storing in DB
OSM containers run with unprivileged (non-root) users.

Running the containers with non-root user adds an extra layer of security.
Limits the processes that can be executed and who can execute them.
Charmed OSM

- Open Source, upstream installer
  - `--charmed` option
    
    https://osm.etsi.org/gitlab/osm/devops/-/blob/master/installers/charmed_install.sh

- OSM Services deployed as Charms

- Management of Kubernetes ingress

- Scales from single node to many with replicas for HA


  - Configured with replicas

```yaml
applications:
  zookeeper:
    charm: zookeeper-k8s
    channel: latest/edge
    scale: 3
    storage:
      data: 100M
```
Ease of Scaling Applications

juju scale-application -m osm nbi 3
Database Connection Strings

```
juju config -m osm nbi mongodb_url="mongodb://172.21.10.1"
```
Virtual Hosts for Ingress Controller

```
juju config -m osm nbi site_url=https://nbi.my-osm.home
```
Additional Infrastructure Support

Existing K8s cluster

Existing LXD cluster

./install_osm.sh --charmed --k8s kubecfg.yaml --lxd lxd.yaml
Thank you!