Community-driven Production OSM Operations

Mark Shuttleworth - Canonical
Readiness assessment

• Feedback from customers
• Feedback from GSIs and NEPs
• “OSM in Production” session
Upstream OSM and Distributions
Production considerations for OSM

- **Availability**
  - OSM components - NBI, LCM, RO, VCA, MON, POL
  - HA, geo-redundancy, backups and disaster recovery
- **Integrations** - authentication, monitoring, ext. systems
- **Deployment** - K8s substrates, proxy/air-gap
- **Operations**
  - Capacity - sizing, planning, scaling
  - Upgrades and patches
- **Security** - ETSI NFV-SEC, CIS, NCSC, NIST
  - Secret storage
Shared open source is very cost-effective for everybody
Reality is messy and mixed
OSM VNFD

Metadata

Operations package - “Charm”
- Lifecycle
- Configuration
- Operation
- Integration

DECLARATIVE

CODE
Charms are packages of scripts to drive apps

**Lifecycle scripts**
- install
- config
- update
- remove
- scale

**“Action” scripts are OSM Primitives**
- “action: backup”
- “action: restore”
- “action: scan-viruses”
- “action: health-check”
- “action: add-repo”
- “action: ...”

**Integration scripts**
- relate-mysql
- relate-ldap
- relate-proxy
- relate-...

These are your operations primitives.
OSM Architecture
OSM is big topology of services

- Many components have significant operational dimensions
  - All of these have to be well done for OSM to have a great reputation
  - It is easier to collaborate on great ops if we share code
  - Currently this is hugely duplicated between distributions
- The VCA is always present in OSM
Example operations

scale-application kafka
run-action mysql/leader backup
relate osm-nbi nagios
upgrade-charm osm-ro
config osm-lcm image=opensourcemanono/lcm:7.1.0
Example OSM charm operations

mysql:
  - scale
  - upgrade
  - backup
  - restore

mongodb:
  - scale
  - upgrade
  - backup
  - restore

osm-ro:
  - scale
  - upgrade

prometheus:
  - scale
  - upgrade

osm-lcm:
  - scale
  - upgrade

osm-nbi:
  - scale
  - upgrade

Shared collection of OSM operations scripts across distributions.

Upstream maintenance and collaboration on operations.
High availability substrates
High availability OSM on Kubernetes
High Availability

- OSM services high availability
- Using LXD clusters for proxy charm deployments
- Using K8s clusters for proxy charm deployments
- Highly available proxy charms
- Highly available VCA
Better integration
Common Code and Operations

FCAPS
Identity
OSS/BSS
Logging
Community-driven roadmap
OSM charms are in devops repo

git clone https://osm.etsi.org/gerrit/osm/devops
cd installers/charm
Community-driven roadmap

• Upstream maintenance of charms by Canonical
• Distributions may use upstream or differentiated charms
• Upstream contributions create shared operations codebase
OSM 7.1 install with shared operations
$ ./install_osm.sh --charmed
[--k8s <kubeconfig>]
[--vca <controller-name>]
[--lxd <cluster-name>]
[--openstack <novarc>]
[--vsphere <vcfg>]

© ETSI 2020
Single-node experience

$ ./install_osm.sh --charmed
OSM upstream charmed install

- Unified single-node, cloud and high-availability deploy
- Substrate-agnostic (Metal, AWS, Azure, VMware, OpenStack)
- All OSM and support components (db, messaging, monitoring)
- Simplified and consistent post-deployment operations
Documentation

• Intro to Python Operator Framework
  
  https://discourse.juju.is/t/first-steps-with-the-operator-framework/3006

• R8 planned documentation:
  - Charmed OSM installation
  - Scaling OSM components
  - Upgrading OSM
  - Scaling VCA
Cooperation
Cooperation

- TNO - OSM improvements
- Simula - Charm support
- Metaswitch - VNFs
- Parallel Wireless, Altiostar (OpenRAN) - Orchestration
- Juniper - SDN
Upstream OSM and Distributions

VIM - OSM

NEP - OSM

ISV - OSM

GSI - OSM

Cloud - OSM

Telco - OSM

Common Code and Operations
Open Source MANO

Find the Canonical OSM team at:

tytus.kurek@canonical.com
alex.chalkias@canonical.com