Open Source MANO

OSM#9 Hackfest
OSM in Production
Alex Chalkias, Eduardo Sousa ( Canonical)
Session goals

- Clarify the current state of the art
- Understand any new issues from the field
- Discuss further enhancements within the OSM community
Production considerations for OSM

• **Availability**
  • OSM components - NBI, LCM, RO, VCA, MON, POL
  • HA, geo-redundancy, backups and disaster recovery
• **Integrations** - syslog, other monitoring
• **Deployment** - K8s substrates, proxy/air-gap
• **Operations**
  • Capacity - sizing, planning, scaling
  • Upgrades and patches
• **Security** - ETSI NFV-SEC, CIS, NCSC, NIST
  • Secret storage
NBI, LCM, RO, POL

- Stateless services on Kubernetes
- High availability is supported
- Data stores are Mongo and MySQL with standard HA
- Shared files provided by Mongo
MON

- MON is currently not scalable nor highly available
  - Future work to spread NFVI & VNF metric collection across multiple instances
- No framework for complex VNF monitoring
  - Very challenging to monitor for e.g. GNMI-based VNFs
• Juju controller
  • High availability with 3 clustered Juju instances
  • Handles thousands of charms on modest capacity (32GB RAM, 4 cores)
  • Roadmap OSM R8 to handle failover automatically
• LXD
  • High availability with 3 clustered LXD nodes
  • Juju already handles failover automatically
• Proxy Charms
  • Roadmap OSM R8 allow control of scaling to 2+ units
  • Need guidelines for HA Proxy Charms
Backup

• Databases - well and widely understood
  • Mongodb
  • MySQL

• VCA
  • Juju controller has built-in backup/restore capability
  • Proxy charm containers snapshot via LXD or underlying filesystem
  • Could standardise backup primitives, e.g

  juju run-action magma-o/leader osm-backup
Geo-redundancy and disaster recovery

- Active/Standby strategy
- Active stack is running normally
- Standby stack is receiving data
- Charms handle data replication
- Transition from standby to active made by the operations team

Unclear if it makes sense to remotely replicate control of local functions.
Integration

- Authentication
- External systems through NBI
  - RBAC policy definition
- MON & LMA:
  - OSM cluster + substrate monitoring
  - VNF workloads
- Export events to external systems (SNMP, Syslog, Prometheus, Graylog, etc)
Deployment

- Openstack cloud
  - Load balancing
  - Block storage backend
  - Pre-created K8s and VNF flavors
- Bare metal machines
  - Machine provisioning (e.g. MAAS)
  - Load balancing (e.g. MetalLB, F5)
- Networking
  - Access to external systems (e.g. LDAP, OSS/BSS, Monitoring)
- Proxied & air-gapped environments
Operations

• Capacity planning
  • Sizing
  • Scaling
• Resource monitoring
  • LXD
  • K8s cluster
  • OSM components
• Cluster scale-out
  • Is my capacity planning correct? How to address alerts?
• Upgrades and patching
  • Any issue that needs urgent fixing? How to enable new feature foo?
Security

- FIPS / CIS hardening for the substrate
- Monitoring of dependencies for vulnerabilities
- CVE patching of upstream OSM Docker images
- ETSI NFV-SEC? NCSC? NIST? Which are important?
- Kubernetes security
  - Authorization Mode: AlwaysAllow or stricter, e.g. RBAC?
  - Resource quota per pod
  - Security contexts
Secrets storage

- Different secrets in use:
  - Database/message queue/external systems credentials
  - SSL certificates
  - Encryption keys

- Currently OSM does not have a coherent approach for secret storage:
  - Some stored in mongodb, others shared in docker environments

- New mechanism for certs/private keys
  - Vault