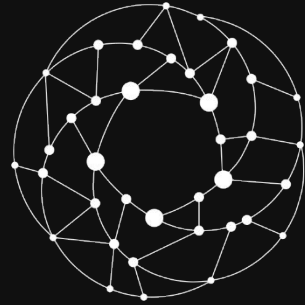


Open Source
MANO

Building operators from the ground up

David García(Canonical)





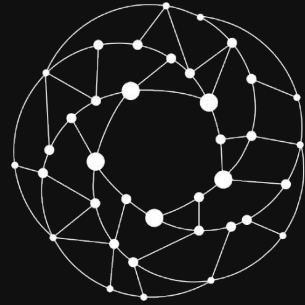
Open Source
MANO

Charm development guidelines

Do **one** thing **well**

Best practices

1. Use meaningful names (config, actions, endpoints)
2. Follow Python best practices
 - a. Write clean code
 - i. Formatting: black, flake8
 - ii. Meaningful names for vars and functions
 - iii. Functions with few/none parameters
 - iv. SRP: Single Responsibility Principle
 - v. Reuse code; avoid duplication
 - b. Test your code: harness
3. Understanding Juju:
 - a. Hooks/events: what is expected in each
4. Make the best operator



Open Source
MANO

Understanding hooks/events

Hooks are events generated by the Juju Controller

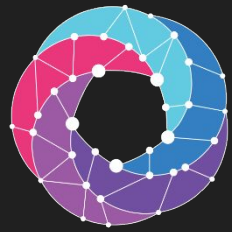
Hook → Code script

```
def _on_install(self, _):  
    subprocess.run(["snap", "install", "prometheus"])  
  
def _on_start(self, _):  
    subprocess.run(["service", "snap.prometheus.prometheus", "start"])
```


Rules writing hooks

1. Hooks are idempotent
2. Hooks are easy to read and understand
3. Where possible, hooks reuse common code
4. Hooks do not return errors... unless there a good reason

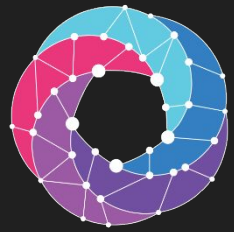
Use the **Operator Framework**



Open Source
MANO

Lifecycle hooks

- **install:**
 - When: At the beginning of the life cycle. Once only.
 - Purpose: Install prerequisite software.
- **config-changed:**
 - When: After *install*, *upgrade-charm*, or after configuration changes. At least once after the agent restarts.
 - Purpose:
 - Cannot assume that SW has started
 - Should not start stopped SW
 - Should restart running SW to update the configuration.
- **start:**
 - When: Immediately after the first *config-changed*.
 - Purpose: Should ensure the charm's software is running
- **stop:**
 - When: Immediately before the end of the unit destruction
 - Purpose:
 - Stop the application
 - Remove any files/configuration created during the application lifecycle
 - Prepare any backup(s)

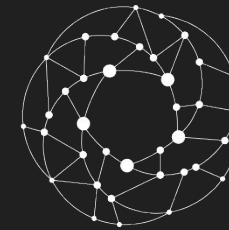


Open Source
MANO

Lifecycle hooks

- **upgrade-charm:**
 - When: Runs immediately after any upgrade operation
 - Purpose: used to reconcile local state written by some other version of the charm into whatever form it needs to take to be manipulated by the current version.
- **update_status:**
 - When: Run by Juju at regular intervals (default=5m)
 - Purpose: Provides constant feedback to the user about the status of the application
- **leader-elected:**
 - When: Run at least once to signify that Juju decided this unit is the leader
- **leader-elected-changed:**
 - When: Run when the leader has set values for other units to respond to

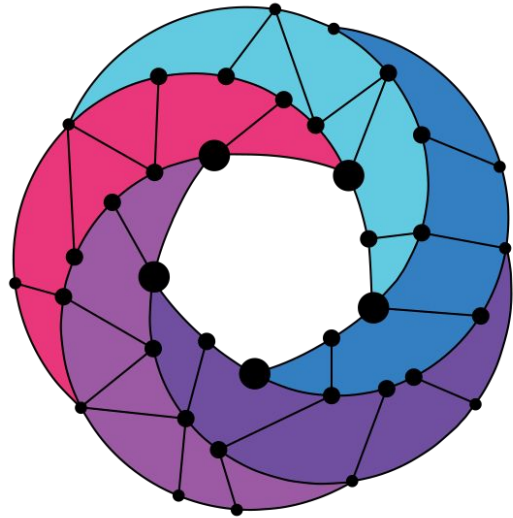
- **[name]-relation-joined:**
 - When: Run only when that remote unit is first observed by the unit.
 - Should not depend on any other relation setting more than the name of the joining unit and the remote private-address setting. If more information is needed, should wait for the relation-changed hook.
- **[name]-relation-changed:**
 - When: Always run after *-joined* and after the relation data changes.
 - Take into account:
 - The settings in the relation (relation data) are determined by the interface
 - If data is missing, do not raise errors, just wait until the data will eventually be there.
- **[name]-relation-departed:**
 - When: Run once only, when the remote unit is known to be leaving the relation
 - Purpose: Should be used to remove all references to the remote unit
- **[name]-relation-broken:**
 - When: Run after every necessary *-departed* hook has been run



Open Source
MANO

Relation hooks

- Juju: <https://juju.is/docs>
- Charms:
 - Charm hooks/events: <https://discourse.charmhub.io/t/charm-hooks>
 - Operator framework:
 - Github: <https://github.com/canonical/operator>
 - Api docs: <https://ops.readthedocs.io/en/latest/>
 - Talk to us: <https://discourse.charmhub.io/>
 - Examples:
 - Kubernetes Charm: <https://github.com/charmed-osm/squid-operator>



Open Source MANO

Find us at:

osm.etsi.org
osm.etsi.org/wikipub