



Mistral

Project Update, OpenStack Summit Vancouver
Brad P. Crochet, thrash, brad@redhat.com

What does Mistral do?

- Workflow (as a) Service
- Provides a mechanism to define tasks and workflows without writing code, manages and stores workflow state while executing them in the cloud environment.



Project background

- Founded during the Havana release of OpenStack
- Around 70 contributors to the latest release

Latest user survey adoption numbers:

- Deployed: 5% (clouds in production or test phase that indicate they are using this project in production or test phase)
- Used in all TripleO deployments since Newton



OpenStack Queens Features

- ➔ Mistral Custom Actions API
 - This initiative was started in Pike and the API stabilized in Queens. Custom workflow actions should now be written with the mistral-lib library
- ➔ New YAML AND JSON expression function for parsing and dumping data
- ➔ Workflow and Action names can now be set dynamically with a YAQL or Jinja2 expression
- ➔ Workflow execution creation can be idempotent by providing Mistral with the ID
- ➔ As per the Queens community goals Mistral now supports policy in code and Zuul v3
- ➔ Add workflow namespaces, which allow workflows and workflow executions to be grouped together. This allows workflows with the same name to exist in their own namespaces

OpenStack Queens Features

- ➔ Cron Trigger execution interval
 - The minimum interval in the cron trigger subsystem can now be configured. Previously it was every second which added unnecessary load on the host. See the cron trigger `execution_interval` setting.
- ➔ Start a new workflow execution based on a previous one
 - `mistral execution-create -s $EXECUTION_ID`
- ➔ Sub-workflow executions can now be filtered by their root execution ID
 - `mistral execution-list --filter root_execution_id=$ROOT_EXECUTION_ID`

OpenStack Rocky - Already Merged

- ➔ Execution events and notifications. A new plugin system for notifying
- ➔ New OpenStack actions for Swift and the OpenStack projects Vitrage, Qinling and Zun
- ➔ Task defaults can now set the safe-rerun policy
- ➔ Added support for py_mini_racer, an easier to install library for using V8 JavaScript engine
- ➔ Workflow executions can't be deleted while they are still running unless --force is provided. This makes workflow deletion much safer by default.
- ➔ Refactored and greatly improved Docker development container
- ➔ Performance Improvements
 - Configurable action definition caching
 - Reduced memory footprint and increased performance by optimising the database access

OpenStack Rocky - In Progress

- ➔ Use Keystone trusts by default in OpenStack actions
- ➔ Workflow Error Analysis
- ➔ Example project with custom Mistral actions
- ➔ Initial version of mistral-extra, the Mistral actions library with support for Ansible actions and a new home for the OpenStack actions
- ➔ Documentation improvements and refactoring

Beyond Rocky

- ➡ Further integration with other services. Maybe Kubernetes or Ansible AWX
- ➡ Stability, performance and HA

Find us!

- New contributors are always welcome!
- General IRC discussion at `#openstack-mistral` (freenode)
- Weekly “office hours” on IRC in `#openstack-mistral` (freenode)
 - Monday at 1500 UTC
 - Wednesday at 0300 UTC
 - Friday at 0800 UTC

Q&A

Thank you!



openstack



@OpenStack



openstack



OpenStackFoundation