Mapping and integrating unknown storage systems

Rohan Kanade, 25 May 2017

tendrl

→ Tendrl.org

→ Github.com/Tendrl

→ twitter.com/Tendrl

What is Tendrl

- Provision, monitor and manage multiple software defined distributed storage systems (currently Ceph and Gluster) through a modern web interface.
- Collection of namespaces like common, node, integrations.$sds, provisioning, $SDS.
- Eg: ceph.flows.CreatePool, tendrl.objects.Disk, tendrl.flows.ImportCluster.

Tendrl Components

- tendrl-node-agents on all tendrl managed nodes.
- Tendrl SDS agents on all tendrl managed storage nodes.
- Stateless tendrl-api.
- tendrl-performance/node-monitoring, monitors via collectd plugins and feeds to graphite.
- Tendrl-dashboard as the web UI.
- All the above components talk to each other via Tendrl’s distributed key-value etcd cluster.

Why another sds management system?

- Sysadmins/devops have Ansible.
- Storage Admins have ….?
- Tendrl aims to provide sds abstractions for the storage admin which are equivalent to platform abstractions provided by Ansible.

Definitions driven development

- Back to the Ansible>Sysadmins, Tendrl->Storage Admins analogy
- Tendrl sees SDS systems as objects (like ceph pool, osds, crushmaps, gluster volumes, snapshots).
- https://goo.gl/20Pgnd (Tendrl Ceph definitions)
- https://goo.gl/HRh9Iv (Tendrl platform definitions)
- Definitions driven approach must provide abstractions to define attributes, valid states, valid actions, valid thresholds on such custom objects.

How are definitions implemented

- Tendrl commons framework enables developers to define building blocks like “object”, “attribute”, “atom” (atomic action on the object), “flow” (collection of “atoms” associated with different objects).
- All Tendrl Objects belong to namespaces (namespace.tendrl, namespace.node_agent, namespace.ceph, namespace.gluster etc).
- Namespaces enable uniform access to objects, flows without dependence on the programming language specific ways of handling/accessing objects.
- Eg: ceph.objects.Osd, tendrl.flows.CreateCluster

Who benefits from definitions

- SDS system developer trying to integrate with Tendrl.
- SDS system admin/expert trying to use Tendrl to manage and monitor their favourite SDS.
- IT/Ops manager trying to implement SLAs, Business rules around the SDS landscape using Tendrl.