SAMSUNG

Towards production-grade Database as a Service in OpenStack

Bartosz Żurkowski, Przemysław Godek

Samsung R&D Institute Poland

Nov 15, 2018

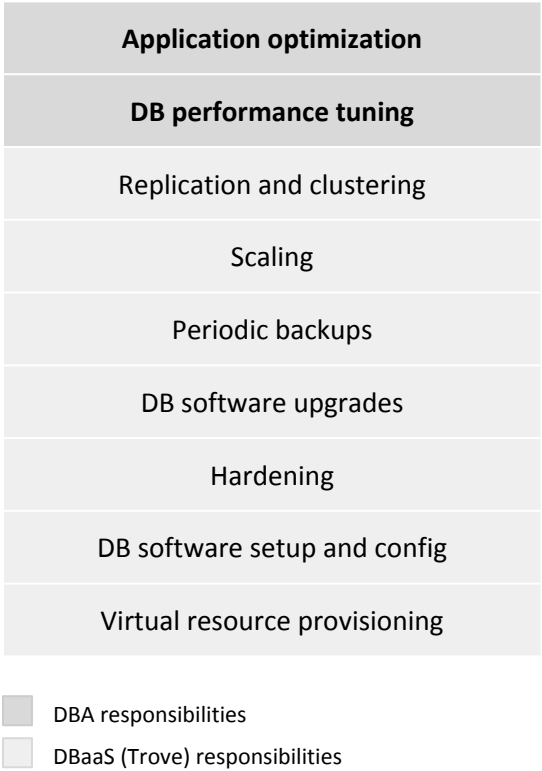
OpenStack Summit, Berlin

Outline

- Trove introduction
- Core features overview
- Production case-study
  - Automated lifecycle management
  - Clustered databases
  - Database proxy
  - Monitoring
  - Autonomic control loop
- Demo
- Q&A

What is Trove?

- Database as a Service for OpenStack
- Provides full database lifecycle management
  - Provisioning, configuration, backups, scaling
- Multi-datastore support
  - 11 database engines
  - Relational, non-relational
  - Single-instance and clustered deployments
- Unified management interface
- Built entirely on OpenStack
  - Synergy of Nova, Cinder, Swift, Glance and Neutron



Core features overview

- Instance provisioning
- Instance resizing (volume, flavor)
- Database and user management
- Configuration groups
- Backups (full, incremental, scheduled)
- Datastore upgrades
- Logs (guest, database)
- Security groups management
- Flavors management

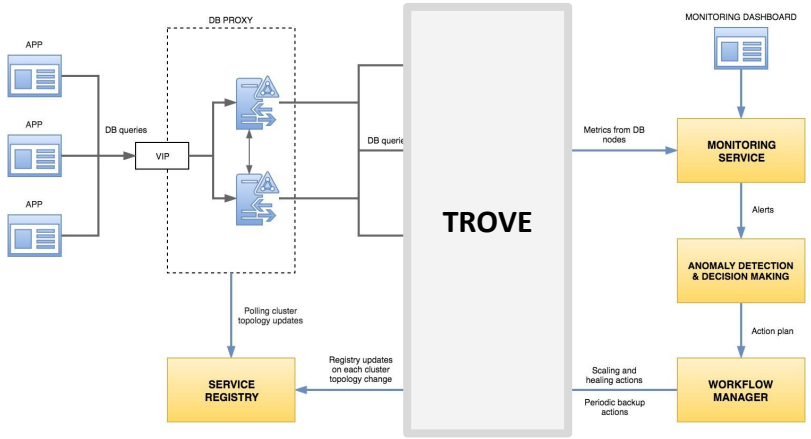
- Cluster provisioning
- Cluster sizing (grow, shrink)
- Replication setup
- Replication failover
  (promote read replica, eject source)

Production case-study

- Automated lifecycle management ✓
- Clustered databases ✓

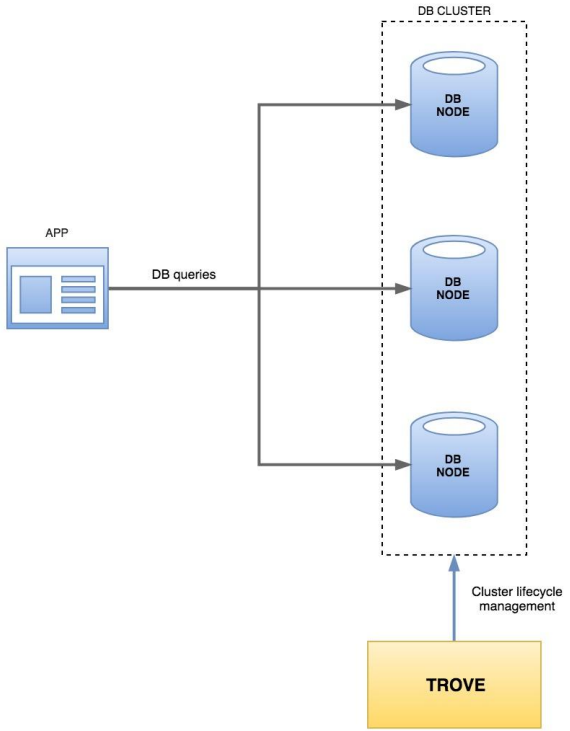
Provided by Trove out of the box

- Database proxy ✗
- Monitoring ✗
- Autonomic control loop ✗
  - Scaling, healing, protection, ...



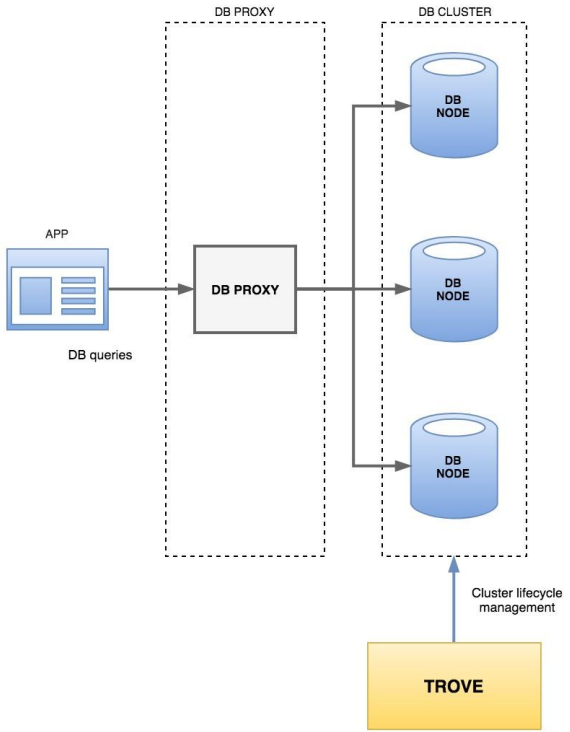
Initial deployment

- Trove provides a functional database cluster
- Applications require a heavy client
  - Multiple contact points
  - Topology awareness
  - Load balancing
  - Failover
  - Read/write splitting
  - …



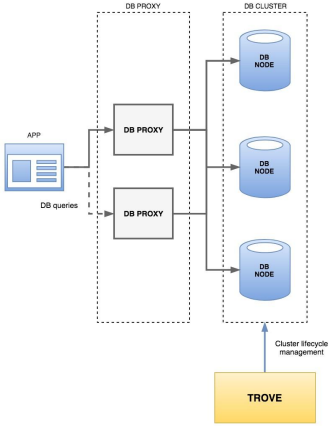
Database proxy

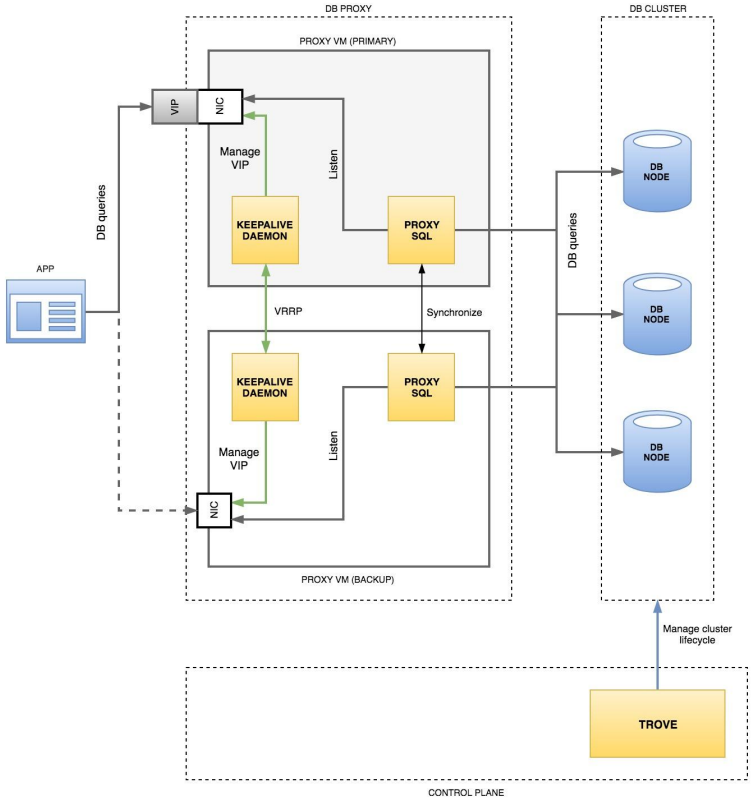
- Minimizes requirements put on the client
- Takes over large part of the communication responsibilities from both the client and database service
  - Failover, load balancing
  - Read/write splitting, sharding
- Provides single endpoint to connect to the database service
- ProxySQL, MaxScale, HAProxy, twemproxy, ...



Database proxy: HA

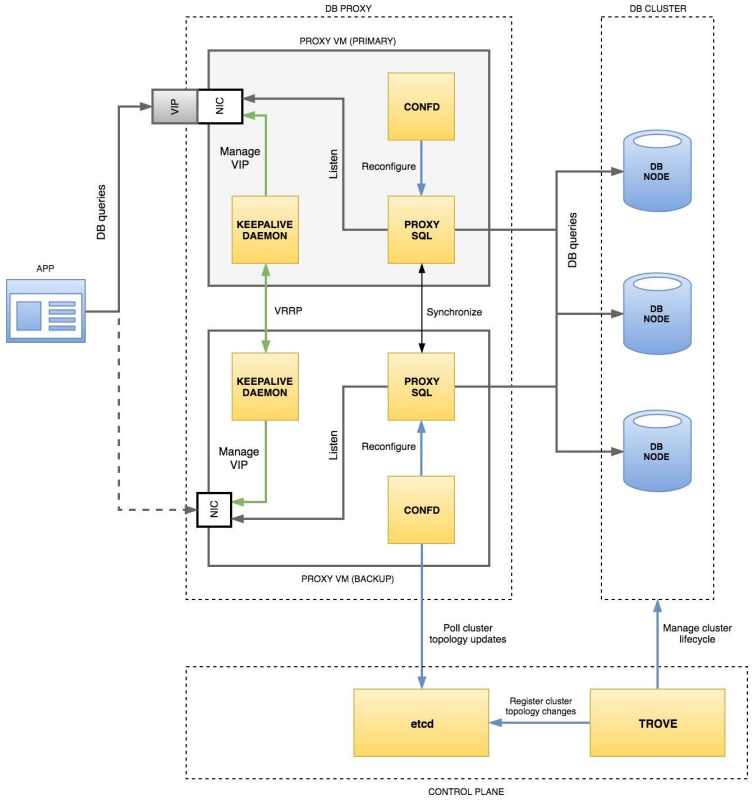
- In order to eliminate the SPoF in proxy layer we introduce backup proxy instances
- Active-passive setup (VRRP, keepalived)
- Single endpoint for applications provided via VIP





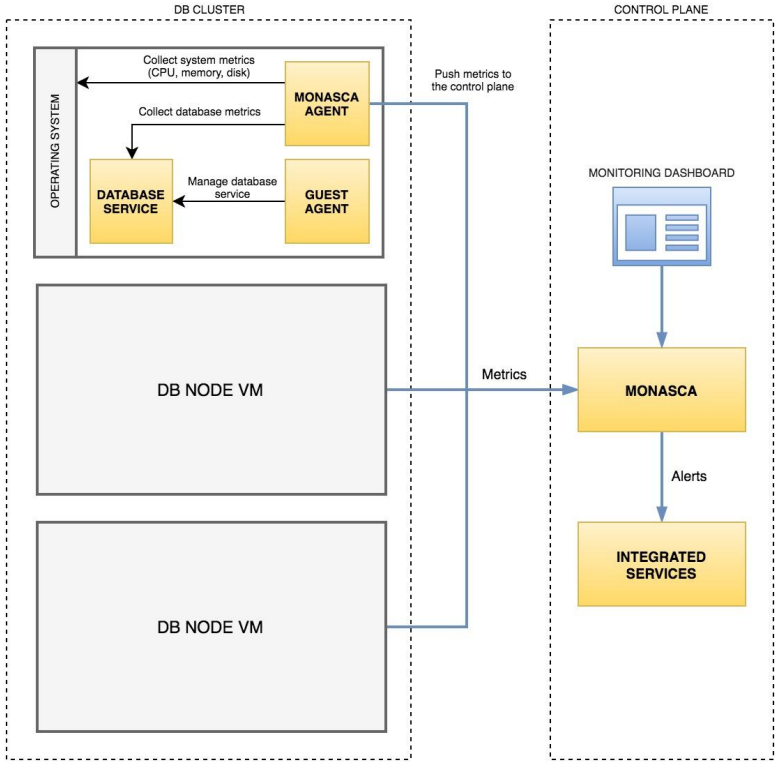
Database proxy: Cluster discovery

- Manual reconfiguration of the proxy layer on each cluster topology change may be troublesome – we may want to automate this process
- For example, we can introduce a service discovery component
  - Cluster nodes register to service registry (Consul, etcd, …)
  - Proxy layer polls registry updates (confd) and reconfigures cluster topology accordingly



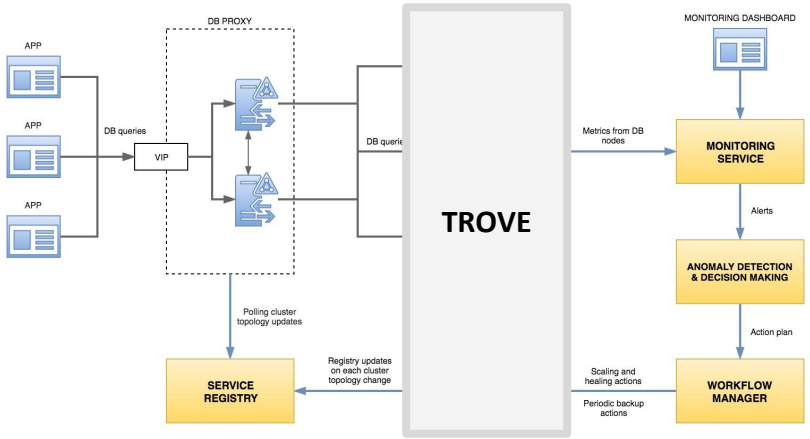
Monitoring

- Production-grade deployments require continuous insight into database performance and behaviour
- Monasca agent installed in cluster nodes collects system and database metrics
- Metrics are forwarded to the Monasca monitoring sink in the control plane
- Based on static thresholds and predictions alarms are emitted for consumption by integrated services
- Monitoring dashboard (e.g. Grafana) provides metric visualizations



Production case-study

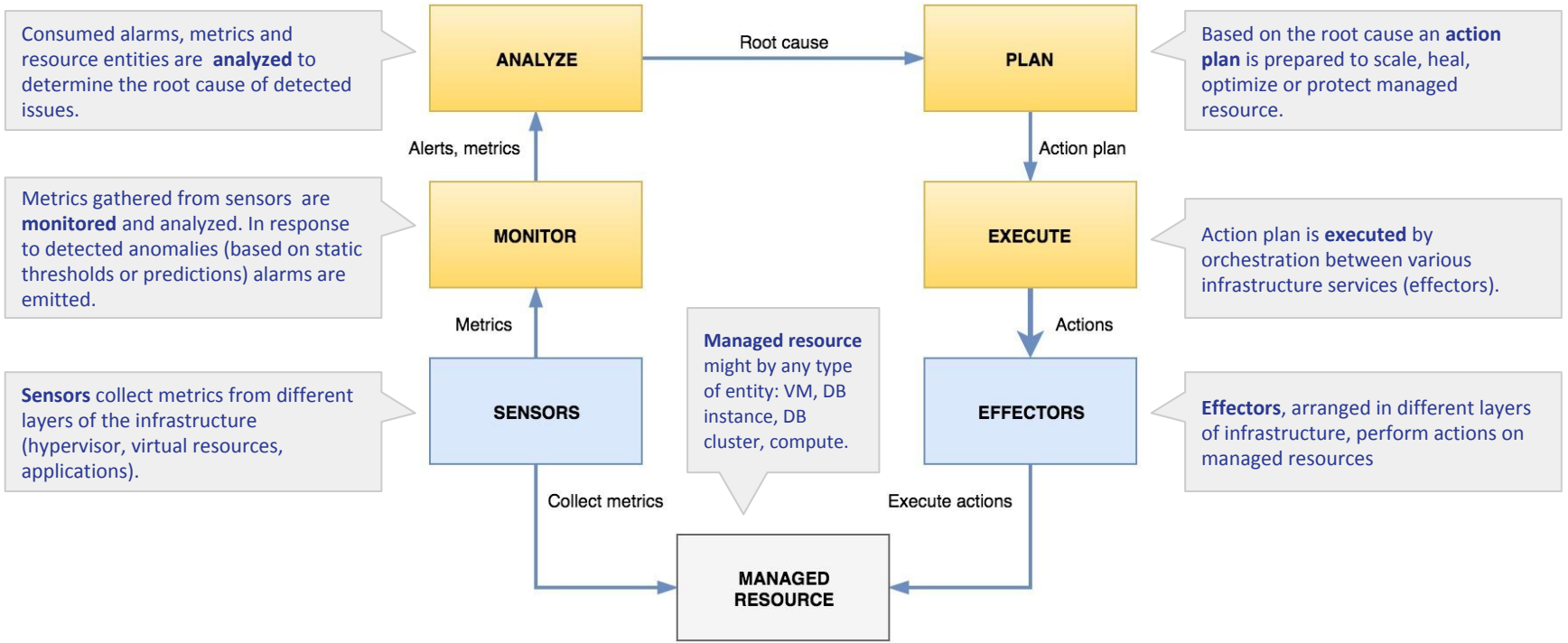
- Automated lifecycle management ✓
- Clustered databases ✓
- Database proxy ✓
- Monitoring ✓
- **Autonomic control loop** ✗
  - Scaling, healing, protection, ...



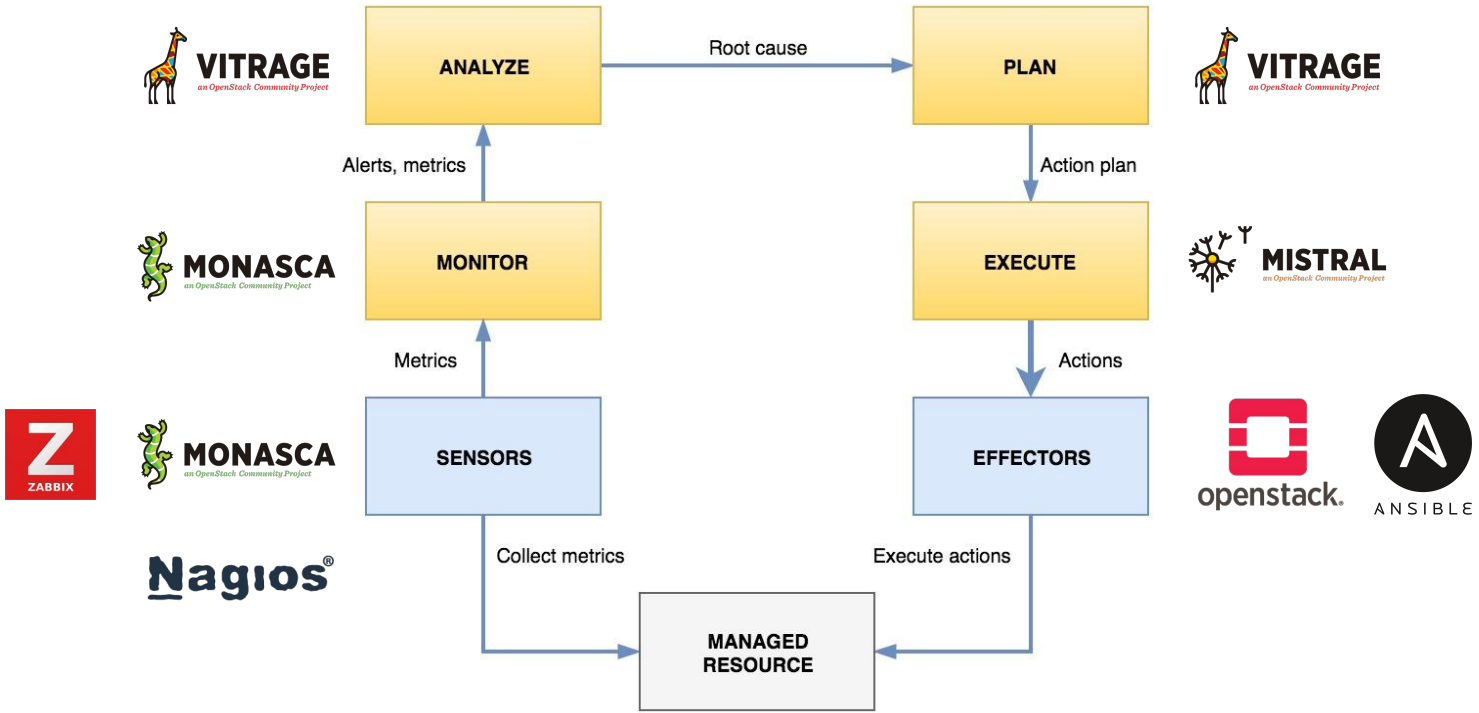
Autonomic control loop

- Concept initiated by IBM in 2001
- Addresses problem of growing complexity spreading in the field of computing systems
- Describes self-managing characteristics of such systems
  - self-configuration, self-healing, self-optimization, self-protection
- Aims to offload human operators from burden of administering large systems
- Compares to human organism and its capabilities to adapt to changing conditions (e.g. immune system, blood pressure, breathing)
- Very often modeled around MAPE architecture

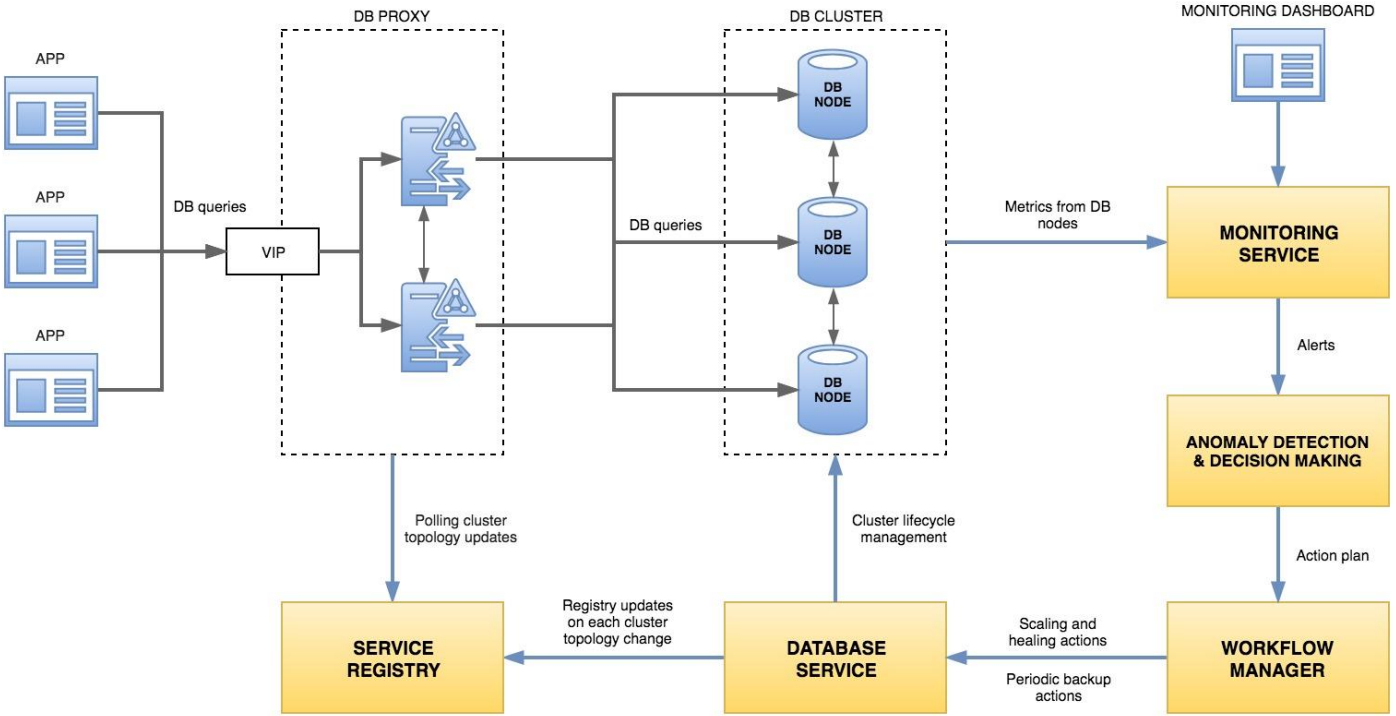
Monitor Analyze Plan Execute (MAPE)



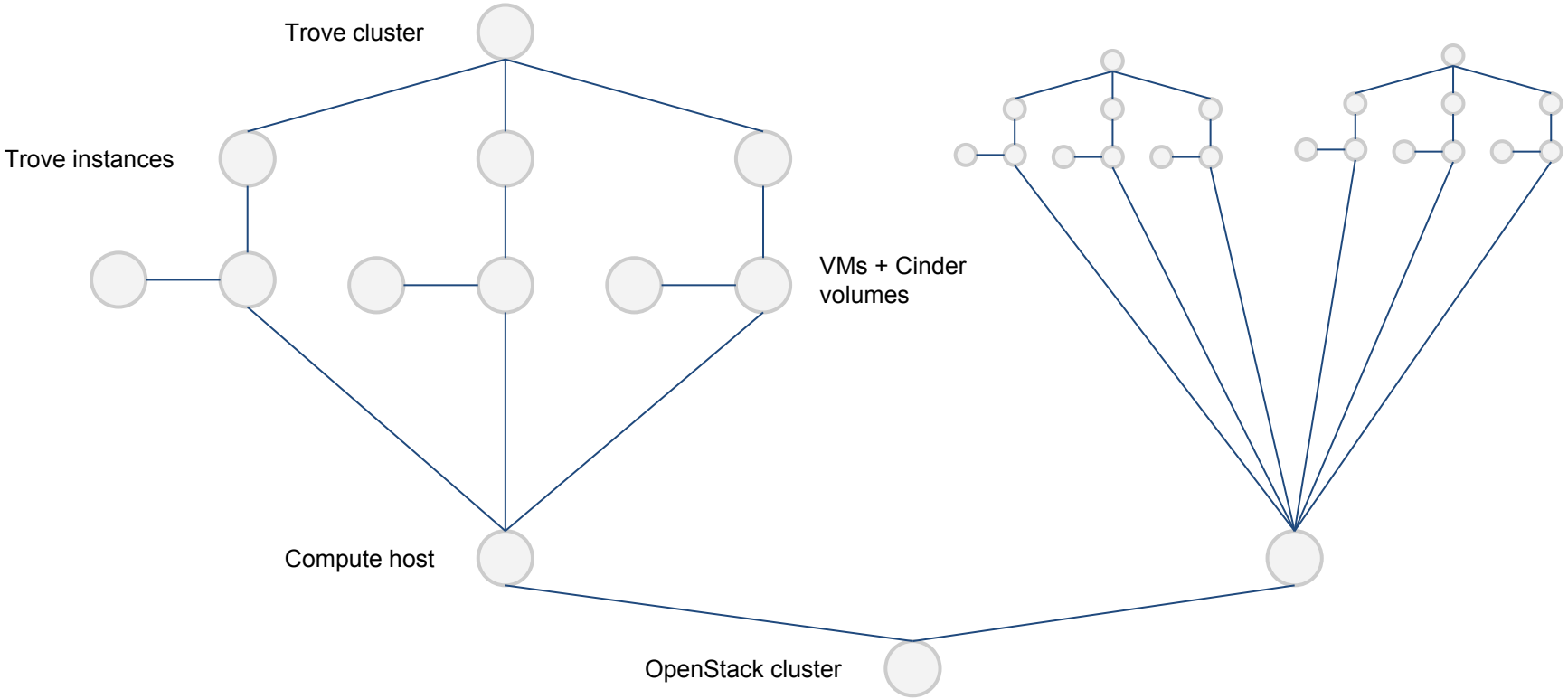
MAPE in OpenStack



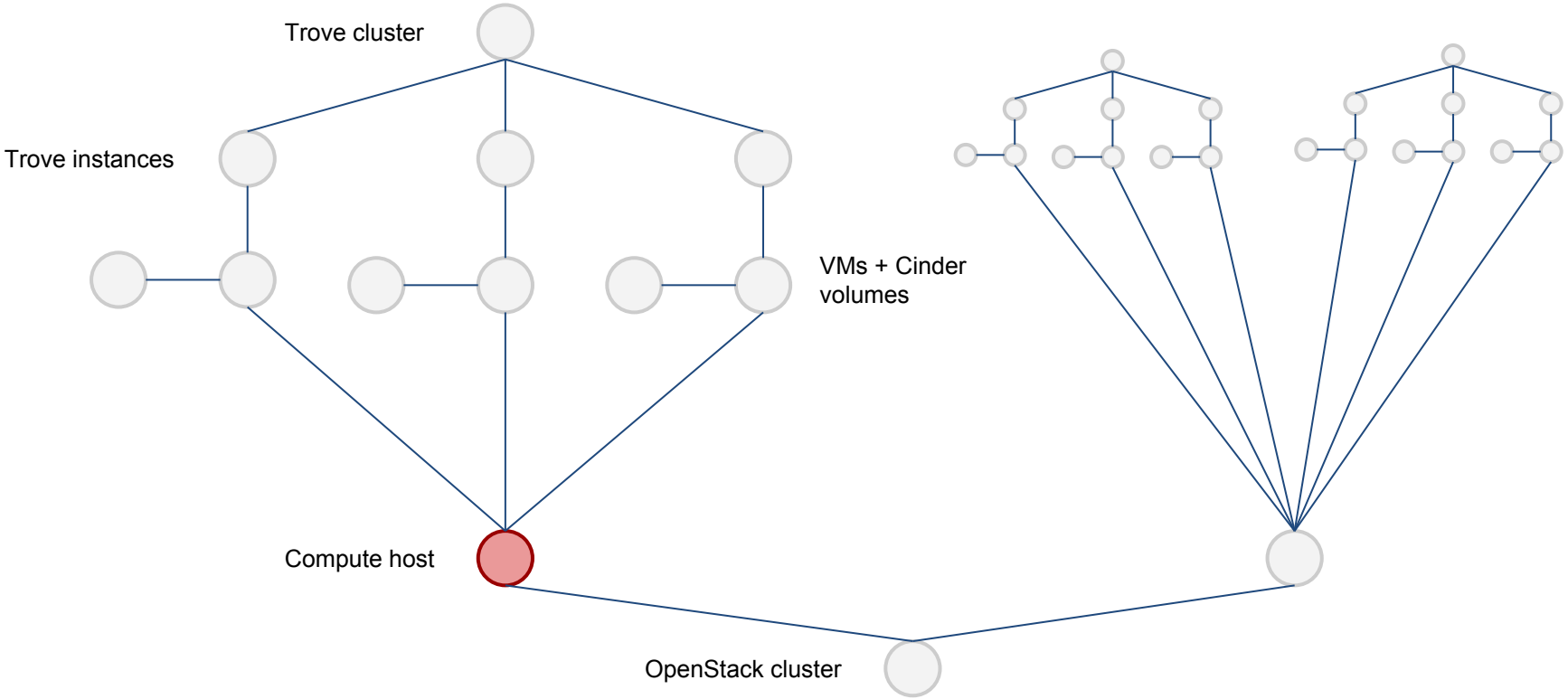
Overall solution architecture



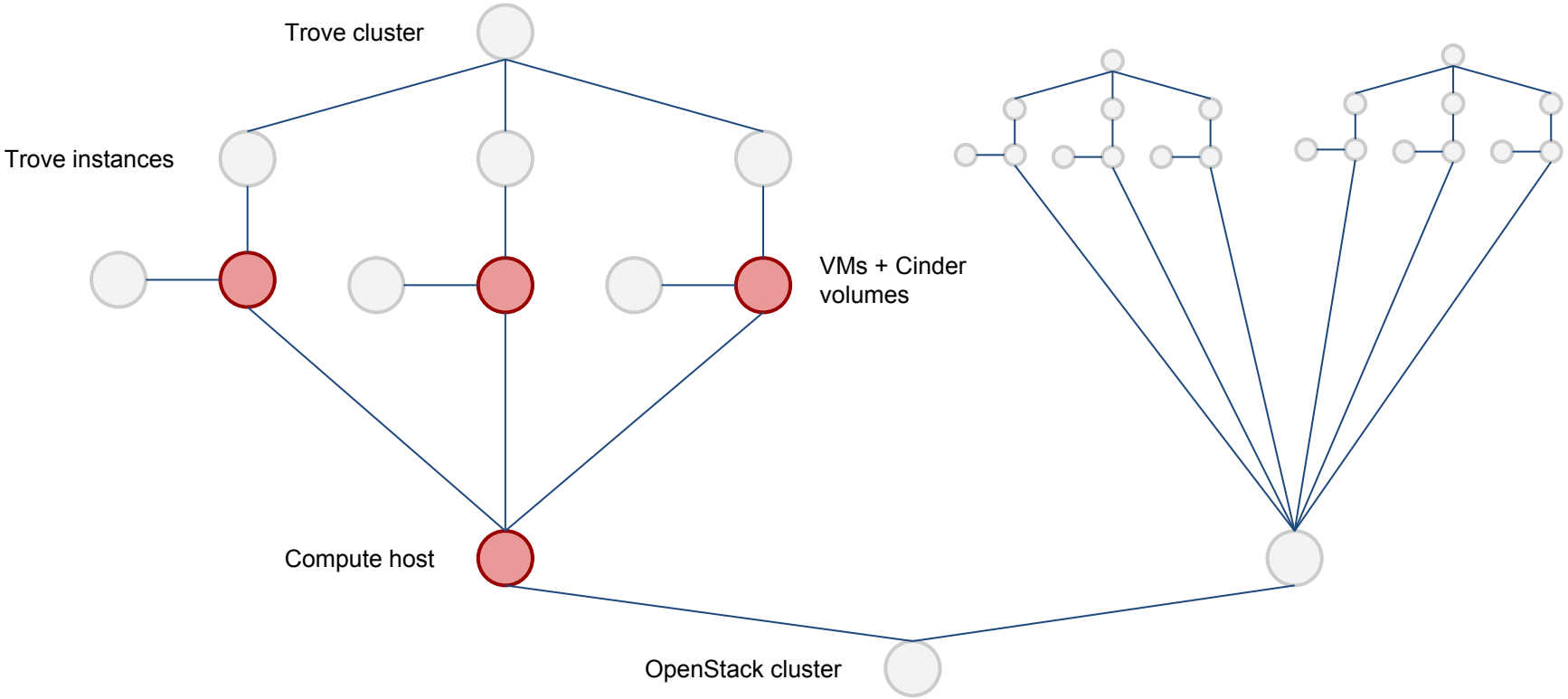
Vitrage example



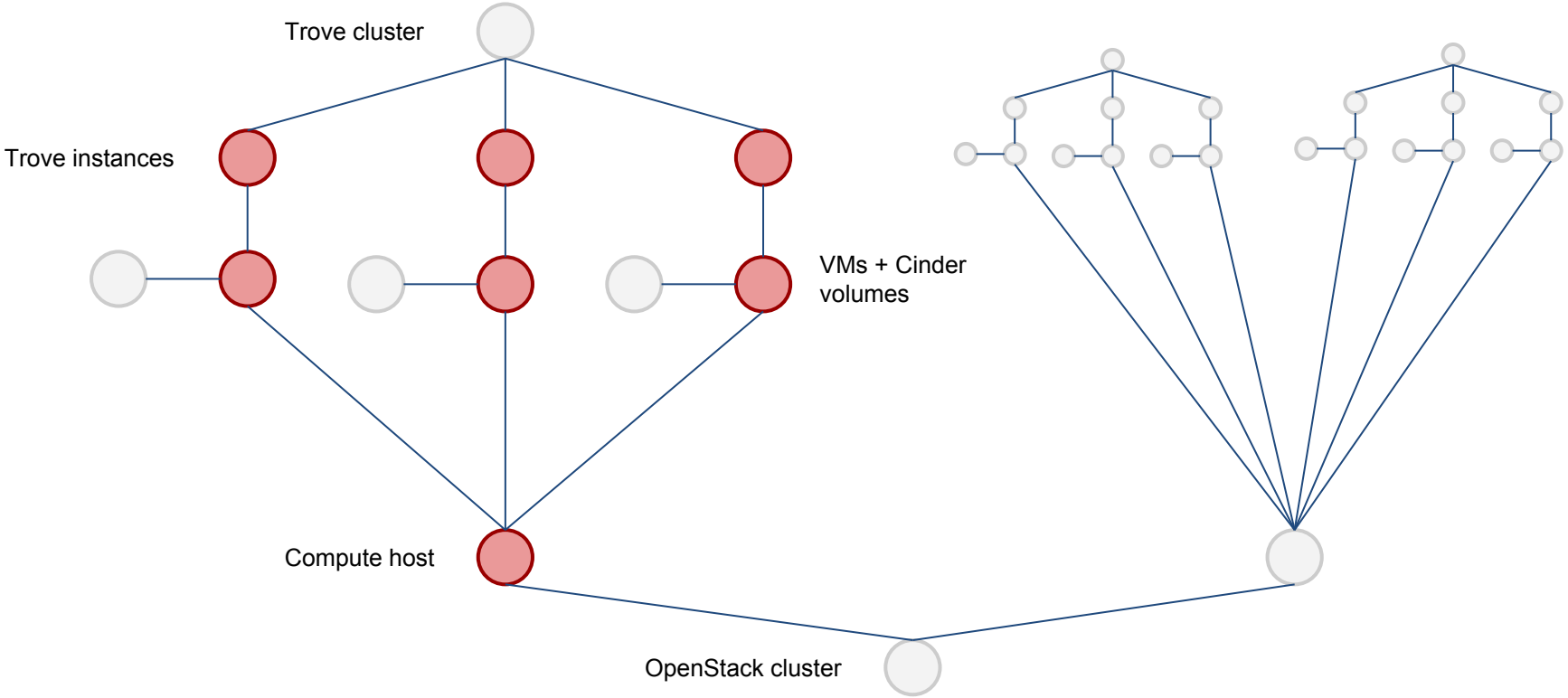
Vitrage example



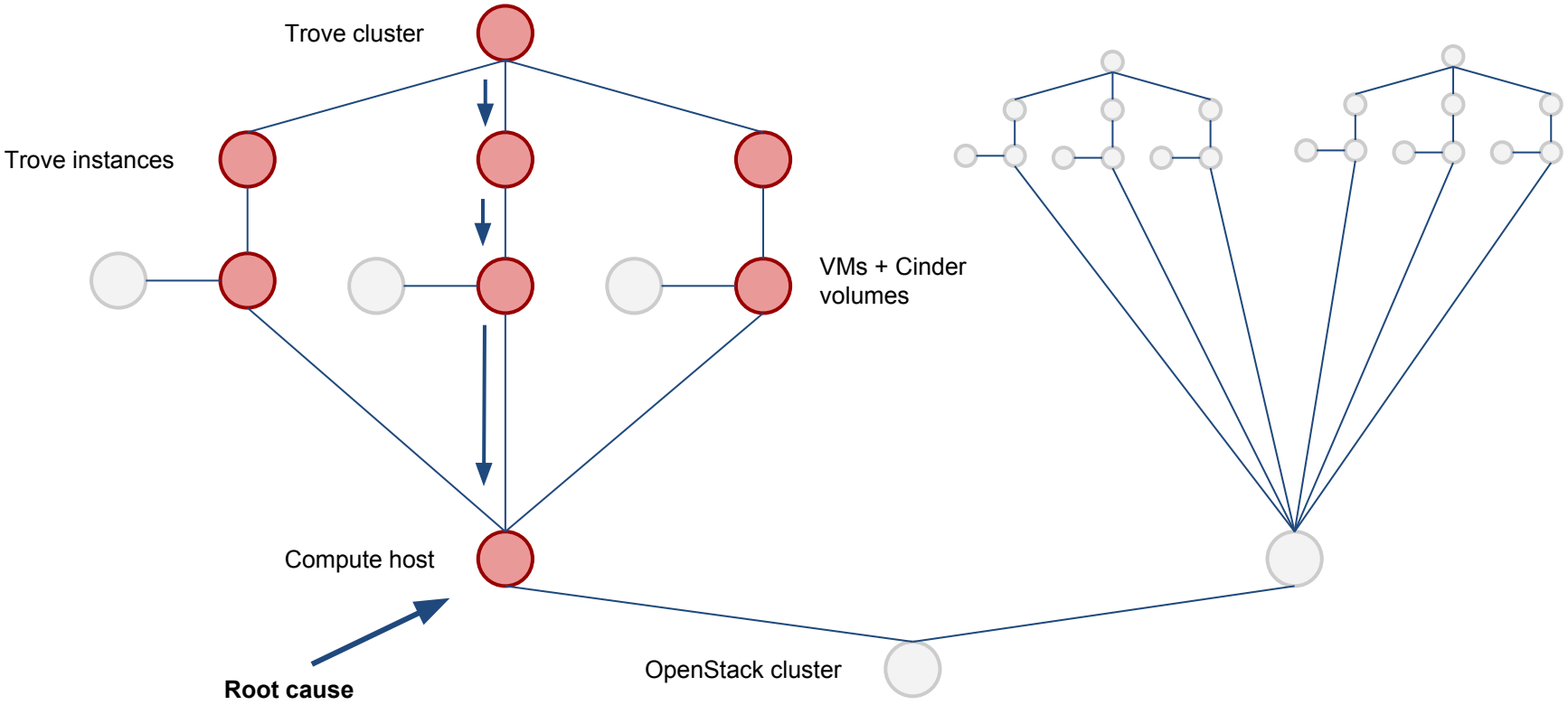
Vitrage example



Vitrage example



Vitrage example



Vitrage example

```
metadata:
  version: 2
  name: trove-cluster-capacity-auto-scaling
  ...
definitions:
  entities:
    - entity:
        category: RESOURCE
        type: trove.instance
        template_id: trove_instance
    - entity:
        category: ALARM
        name: trove-instance-low-disk-space
        template_id: low_disk_alarm_on_instance
    ...
  relationships:
    - relationship:
        source: low_disk_alarm_on_instance
        target: trove_instance
        relationship_type: on
        template_id: low_disk_on_instance
scenarios:
  ...
  - scenario:
      condition: low_disk_on_instance and trove_cluster_contains_instance and low_disk_on_cluster
      actions:
        - action:
            action_type: execute_mistral
            properties:
              workflow: trove_resize_cluster_volume
              input:
                cluster_id: get_attr(trove_cluster,id)
```

Mistral example

```
---
version: 2
trove_resize_cluster_volume:
  ...
  input:
    - cluster_id
  tasks:
    calculate_new_volume_size:
      action: trove.clusters_find id=<% $.cluster_id %> %>
      publish:
        new_volume_size: <% task(calculate_new_volume_size).result.instances.first().volume.size * 2 %>
      on-success:
        - resize_cluster_volume
    resize_cluster_volume:
      action: trove.clusters_resize_volume cluster=<% $.cluster_id %> volume_size=<% $.new_volume_size %> %>
      on-success:
        - wait_for_cluster_resize
    wait_for_cluster_resize:
      action: trove.clusters_get cluster=<% $.cluster_id %>
      retry:
        delay: 10
        count: 30
        continue-on: <% task(wait_for_cluster_resize).result.task.name != 'NONE' %>
```

Demo

SAMSUNG

Demo outline

- Create Galera cluster consisting of 3 nodes (Trove)
- Present monitoring dashboard in Grafana visualizing Monasca measurements gathered from cluster nodes
- Show "low disk space" alarm definitions in Monasca
- Present Vitrage Entity Graph
- Present Vitrage template with "low disk space" alarm scenario
- Present Mistral workflow responsible for scaling cluster capacity
- Run sysbench to populate data in the cluster
- Watch Monasca alarms in appearing in the Entity Graph
- Show triggering Mistral workflow by Vitrage
- Demonstrate that cluster has been successfully resized

What we didn’t show in the demo

- ProsxySQL setup
- Seamless database communication during capacity scaling
- Database failover scenarios
- Proxy reconfiguration on topology change
- Automation with Heat and Murano

Thanks to

- Ifat Afek and Muhamad Najjar
  - For identifying serious bug in Vitrage template during the Summit (yesterday)
- Daug Szumski
  - For support in deploying Monasca on top of Kolla
- Maciej Gaweł
  - For helping solve various networking and deployment issues

Q&A

SAMSUNG

Thank You

SAMSUNG