MaaS as a Backend:

Provisioning Infrastructure for Teuthology Suites

Overview

The Current Landscape: FOG

- Teuthology actively and successfully uses the FOG Project for bare-metal imaging and provisioning.
- FOG remains a fully supported and heavily utilized backend for our testing infrastructure.

Expanding the Toolkit: Adding MaaS

- Introducing Metal as a Service (MaaS) as an additional supported backend option.
- Allows Teuthology to interface with labs that already use MaaS or prefer a different deployment model.

Why Add MaaS ?

API-Driven Architecture

- Offers a robust REST API that integrates cleanly and natively with the Teuthology.

Cloud-Init Native

- Dynamically configures the OS on the fly via cloud-init.

Cloud-Like Experience

- Treats physical servers as elastic, dynamically allocatable resources.

Workflow

Selection

- Teuthology allocates a node dynamically based on **machine_type**, OR a user explicitly requests to lock a specific node by name.

Request

- Teuthology instructs MaaS to allocate the chosen node(s) and deploy a specific OS.

Execution

- MaaS handles the hardware - power on, PXE boot, and cloud-init.

Testing

- Teuthology connects via SSH to deploy Ceph and run tests.

Cleanup

- Teuthology issues an unlock; MaaS wipes the node and returns it to the pool.

Prerequisites

MaaS Infrastructure

- A fully operational MaaS controller and rack controller(s).

Enlisted Nodes

- Target machines must be enlisted, commissioned, and in the "Ready" state in MaaS.

Network

- Proper routing between the Teuthology control node and the MaaS-managed subnets.

Teuthology Ecosystem

- A Teuthology control node, Paddles (backend database), and Pulpito (dashboard).

MaaS Server

Authentication

- Uses MaaS API Keys generated from the MaaS Web UI or CLI.

User Roles

- Best practice is to use a dedicated "Teuthology" user account in MaaS to track machine ownership.

Network Access

- Teuthology requires access to the MaaS API endpoint (usually port 5240).

Control Node

- Acts as the orchestrator running **teuthology-dispatcher** and **supervisor** processes.
- Requires SSH keypair generation (**teuthology user**) to be injected into MaaS nodes during provisioning.
- Contains the core Teuthology Python virtual environment and repository clone.

Configuration

- Add the below section to **~/.teuthology.yaml** file on the control node

```
maas:
  api_url: http://maas.example.com:5240/MAAS/api/2.0/
  api_key: <consumer_key>:<consumer_token>:<secret>
  machine_types: ['typeA', 'typeB']
  timeout: 1200
  user_data: teuthology/maas/user_data/maas-{os_type}-{os_version}-user-data.txt
```

Lock Operation

```
teuthology-lock --lock-many <count> --machine-type <type>
```

Teuthology Action

- Queries Paddles, allocate nodes.
- Makes a REST API call to MaaS to deploy selected nodes.

MaaS Action

- *New | Ready -> Allocated -> Deployed -> Lock*
- *Deployed -> Unlock -> Released -> Allocated -> Deployed -> Lock*

Unlock Operation

```
teuthology-lock --unlock <node_name>
```

Teuthology Action

- Teuthology queries Paddles and Validate node owner.
- Makes a REST API call to MaaS to release selected node.

MaaS Action

- *Ready | Allocated*
- *Deploying -> Abort -> Ready*
- *Deployed -> Release -> Unlock*

Launching Suite

```
teuthology-suite -v -s <suite_name> -c <ceph_branch> -m <machine_type>
```

Teuthology Action

- Evaluates available inventory and explicitly select the target nodes for the job.
- Issues a deploy command to the MaaS API for the chosen node, passing the required OS.

MaaS Action

- Follows same process as lock operation.

Monitoring and Results

MaaS Dashboard

- Track bare-metal deployment phases.
- Powering up, loading ephemeral environment, installing OS.

Pulpito Dashboard

- Track the Ceph test results and log outputs.

Troubleshooting

- MaaS console logs for PXE issues.
- Teuthology supervisor logs for integration test issues.

Demo (LXD Environment)

Acquire Node

```
teuthology-lock --lock-many 2 --machine-type lxd
teuthology-lock --lock saya01 --os-type ubuntu --os-version 22.04
```

Run Test

```
teuthology-suite -m lxd -d ubuntu -l 1 --subset 1/23 \
    -s smoke -c tentacle -S 29f6b181c6e73f86be5c859298e82fcff9b8210a
```

Release

```
teuthology-lock --unlock saya01
```

Resources

- Tracker - https://tracker.ceph.com/issues/72252
- PR - https://github.com/ceph/teuthology/pull/2105
- Proposal - https://pad.ceph.com/p/Maas-Teuthology-Integration

Thank You

Question ?