OpenStack in Action!

Emilien Macchi
Software Engineer @eNovance by Red Hat
Who am I?

- Emilien Macchi
- French citizen residing in Montreal
- Software Engineer at eNovance (by Red Hat)
- Active contributor involved in OpenStack since 2011
- Specialized in Automation & Product development
- Passionated by traveling, running and learning new things
Your Data-center is changing.
Back in old-days

- Plan the project
- Estimate resources we need
- Submit a ticket
- Wait
- Wait a bit more...
- Get what you need (sometimes)
Change the model

- Still users/devs
- Ticket System to API's
- Server/Network/Storage admins to Cloud admins that feed resource pools
- Agile, scalable, automated, predictable
OpenStack
What is OpenStack?

- Open-Source IaaS Cloud Software
- Launched by NASA and Rackspace in 2010
- Massively scalable
- Managed by the OpenStack Foundation
- Fast grow and large adoption
- Release every 6 months
- 10 releases (current stable is Juno)
- Upgrades supported
Who is doing OpenStack?

- 130 organizations
- 1,420 devs
- 18,704 commits

Source: activity.openstack.org
OpenStack Dashboard
Use-cases

• SaaS and Web vendors
  – Delivering a SaaS (ERP) application with Swift
• Service Providers
  – Provides SIP-based call control for voice and video
• Enterprise Private Cloud & IT Operations
  – Private Cloud for HPC Cancer Research
• Infrastructure technology suppliers
  – Provide Storage driver for Cinder
How does it work?
You may want a product.
What you need

- Reference Architecture
- Supportability
- Stability & Code maturity
- Lifecycle
- Certified Hardware
- Certified Operating System
- Storage, Network, Virtualization, Ecosystem, Security
- Scalable
- Highly available
- Flexible
- Secure
Scalable bare-metal deployment

- Image-based deployment
- Build images in advance
  - Automated with Jenkins
  - Once built and archived => reproducible at wish
- Philosophy:
  - Reusable images for all customers, so no specific to configuration
  - Configuration will be done by an config management system like Puppet
OpenStack Images

- **Install Server**
  - Puppet master
  - Bare-metal provisioning server
  - Upgrade server
  - CI server

- **OpenStack Full**
  - All OpenStack & Ceph services with none activated
  - Puppet client
eDeploy

- Bare-metal provisioner
- Based on standard protocols (PXE / iPXE, HTTP and rsync)
- Automatic Hardware matching
- No auto-install description: powerful configuration by Python scripts to do only the hardware config.
Hardware Validator: AHC

- Automatic Health Check
- Check that hardware is ready to receive OpenStack
- Validate machine components individually
- Validate the set of machines
- Detect black sheeps
- Integrated in eDeploy
Step-by-step deployment

- Based on 100% upstream Puppet modules (~50)
- OpenStack services configuration for HA architectures
- Flexible to many use cases and customer specifics settings
- Evolve with deployments
- Fully unit tested
- Deployment as a 5 steps scenario where each step:
  - Is composed of Puppet classes managed by Hiera
  - Is validated by integration tests (serverspec)
  - Can be debugged easily (TDD style)
Step-by-step workflow
Sanity

- Validate an OpenStack deployment
- Based on Tempest to validate API / CLI (1600 tests)
- Javelin to test resource surviving after an upgrade
- Smoke scripts to create resources within a scenario
Deployment by Jenkins Jobs

5 jobs:
- AHC: hardware validation
- eDeploy: bootstrap / hardware management
- Puppet: configuration
- Sanity: validation
- Upgrade: upgrade the platform to the last release
Upgrade process

- **eDeploy image-base:**
  - Predictible and reproducible
  - Faster than package upgrades (rsync vs packaging)
- Orchestrated by Ansible playbook
- Configuration updated by Puppet
- Validated by Tempest (javelin + functional tests)
- Works only from n to n+1 (n to n+2 not possible)
Upgrade preparation

- Detect packages updates (eDeploy)
- Design Ansible playbooks

- OpenStack Icehouse to Juno + Kernel
- Smart orchestration to manage services upgrades
Upgrade execution

1. **Version N**
   - Orchestrate the upgrade (Ansible)
   - Upgrade by roles:
     - Controller (example):
       - stop services
       - eDeploy upgrade
       - Wait for reboot
       - OpenStack DB upgrades
       - start services
     - Order:
       - computes (with VM evacuation), network, storage, load-balancers and controllers
       - Short downtime

2. **Update the configuration (Puppet + Serverspec)**
   - Validating steps:
     - Validating resources surviving
     - Running

3. **Validate the upgrade (Tempest)**
   - with

4. **Version N+1**
EVERYTHING IS OPEN-SOURCE.
Do it yourself

• OpenStack core projects from upstream
• OpenStack Puppet modules from Stackforge & Puppetlabs
• eDeploy & configuration tools on Github / eNovance
  – https://github.com/enovance/edeploy
  – https://github.com/enovance/edeploy-roles
  – https://github.com/enovance/config-tools
  – https://github.com/stackforge/puppet-openstack-cloud/
  – https://github.com/enovance/openstack-yaml-infra
  – http://spinalstack.enovance.com
THANK YOU

emilien@redhat.com

@EmilienMacchi