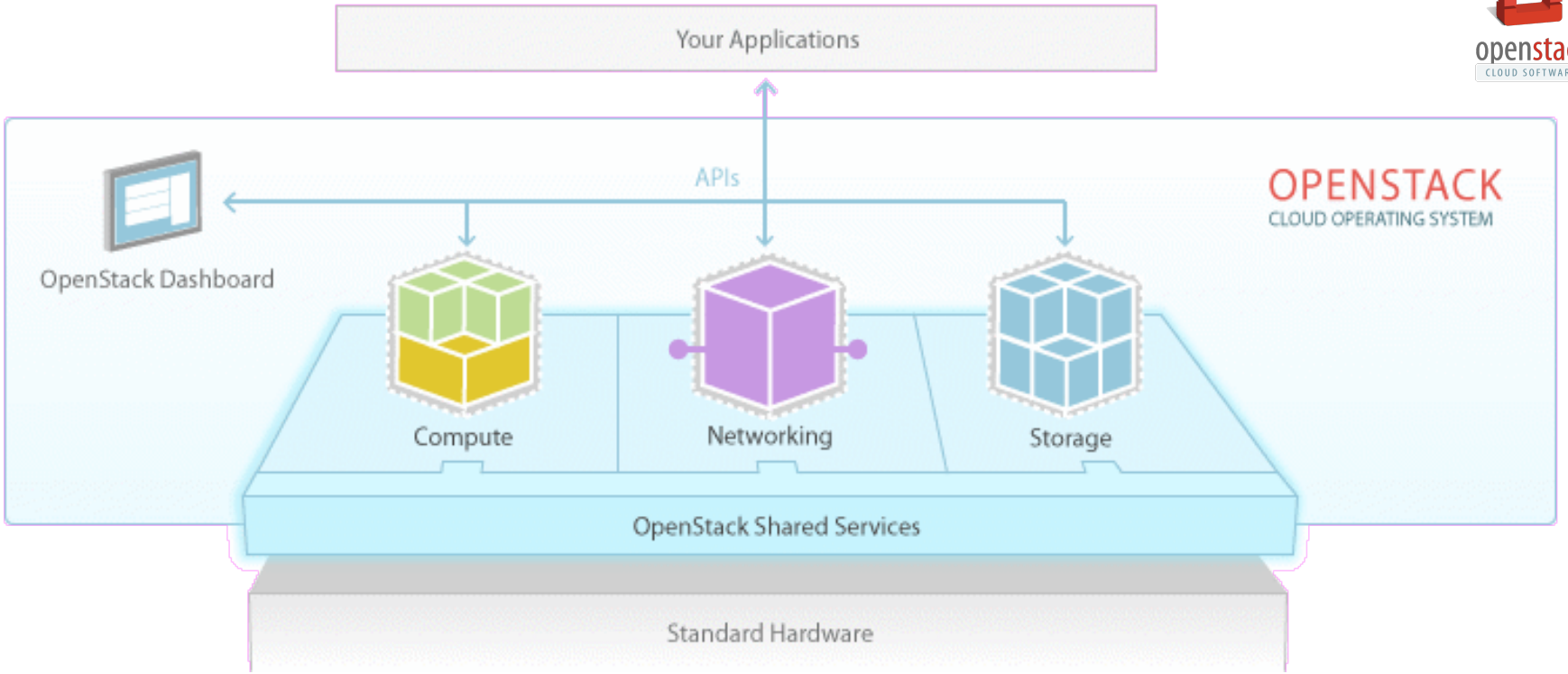
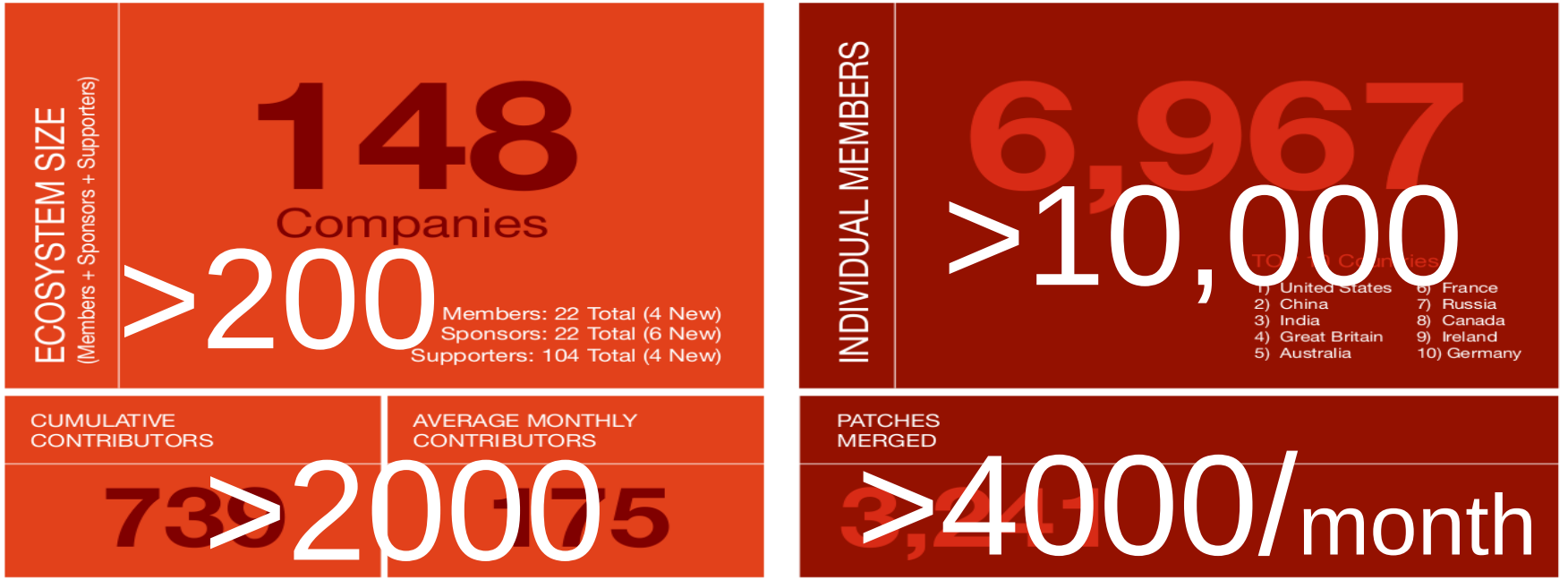
TripleO:
OpenStack on OpenStack

Clint Byrum – HP Cloud, Master Technologist
Twitter: @SpamapS
Email: clint@fewbar.com


Your Applications
APIs
OPENSTACK
CLOUD OPERATING SYSTEM
OpenStack Dashboard
Compute
Networking
Storage
OpenStack Shared Services
Standard Hardware

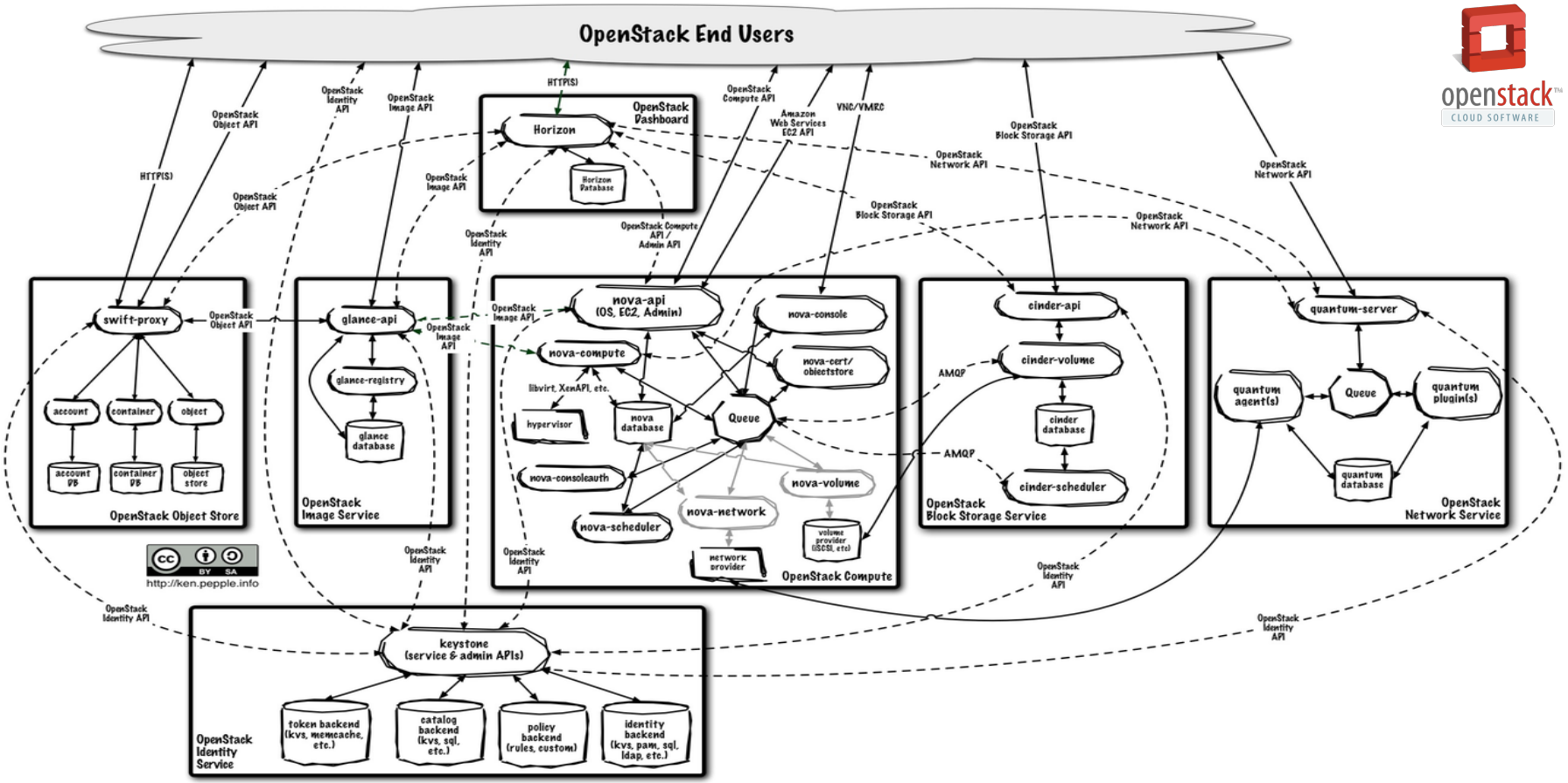
Community Size & Growth



openstack
CLOUD SOFTWARE

Velocity

→ Cloud Enables Increased Velocity
→ Cloud Drives Agility
→ Develop and Test and Deploy in Cloud

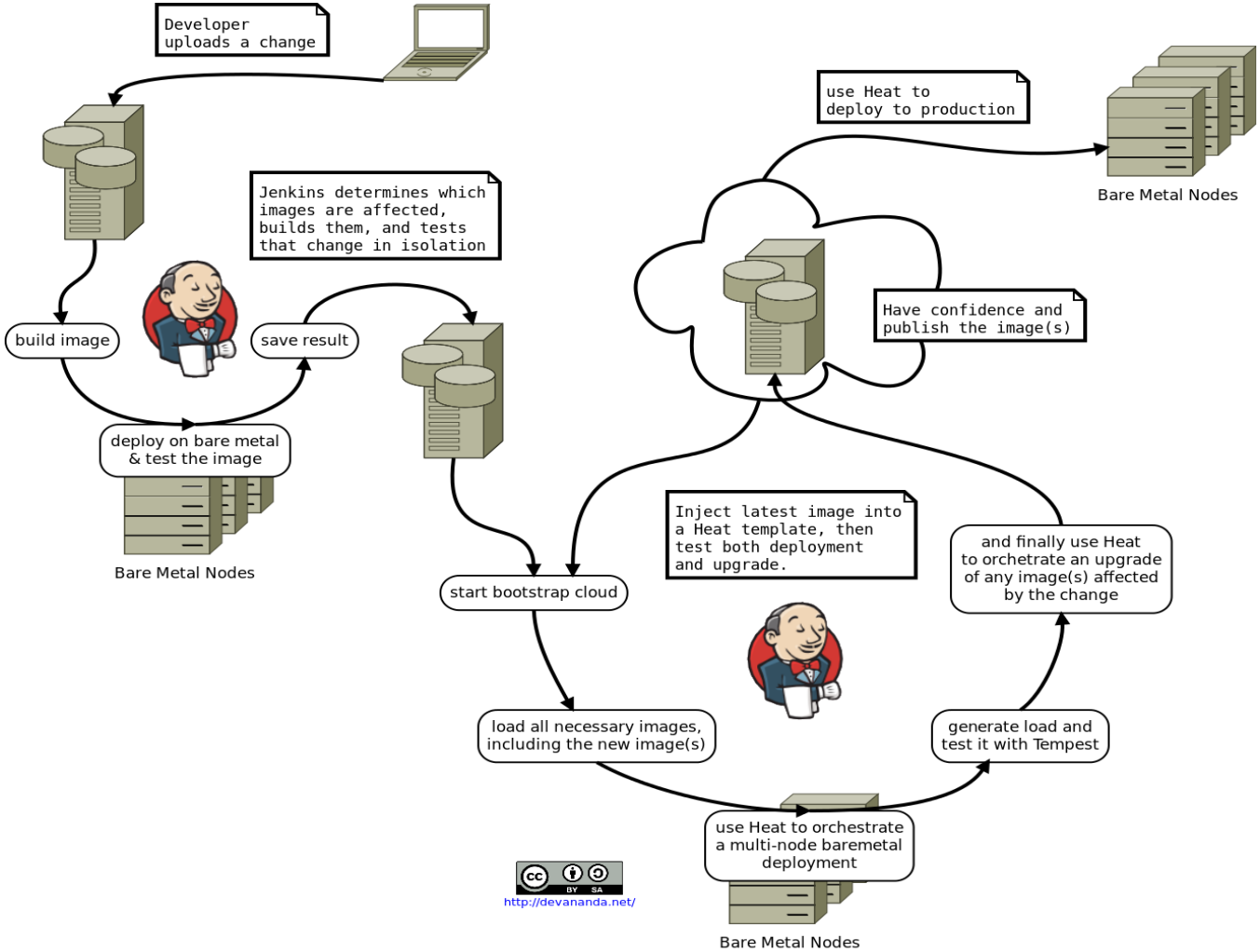

OpenStack End Users
HTTP(S)
OpenStack Dashboard
Horizon
Horizon Database
OpenStack Object API
OpenStack Identity API
OpenStack Image API
OpenStack Compute API
Amazon Web Services EC2 API
VNC/VMRC
OpenStack Block Storage API
OpenStack Network API
OpenStack Compute API / Admin API
swift-proxy
account
container
object
account DB
container DB
object store
OpenStack Object Store
glance-api
glance-registry
glance database
OpenStack Image Service
nova-api (OS, EC2, Admin)
nova-compute
libvirt, XenAPI, etc.
hypervisor
nova database
Queue
nova-console
nova-cert/ objectstore
nova-consoleauth
nova-volume
nova-scheduler
nova-network
network provider
volume provider (iSCSI, etc)
OpenStack Compute
cinder-api
cinder-volume
cinder database
cinder-scheduler
AMQP
OpenStack Block Storage Service
quantum-server
quantum agent(s)
Queue
quantum plugin(s)
quantum database
OpenStack Network Service
keystone (service & admin APIs)
token backend (kvs, memcache, etc.)
catalog backend (kvs, sql, etc.)
policy backend (rules, custom)
identity backend (kvs, pam, sql, ldap, etc.)
OpenStack Identity Service
http://ken.pepple.info

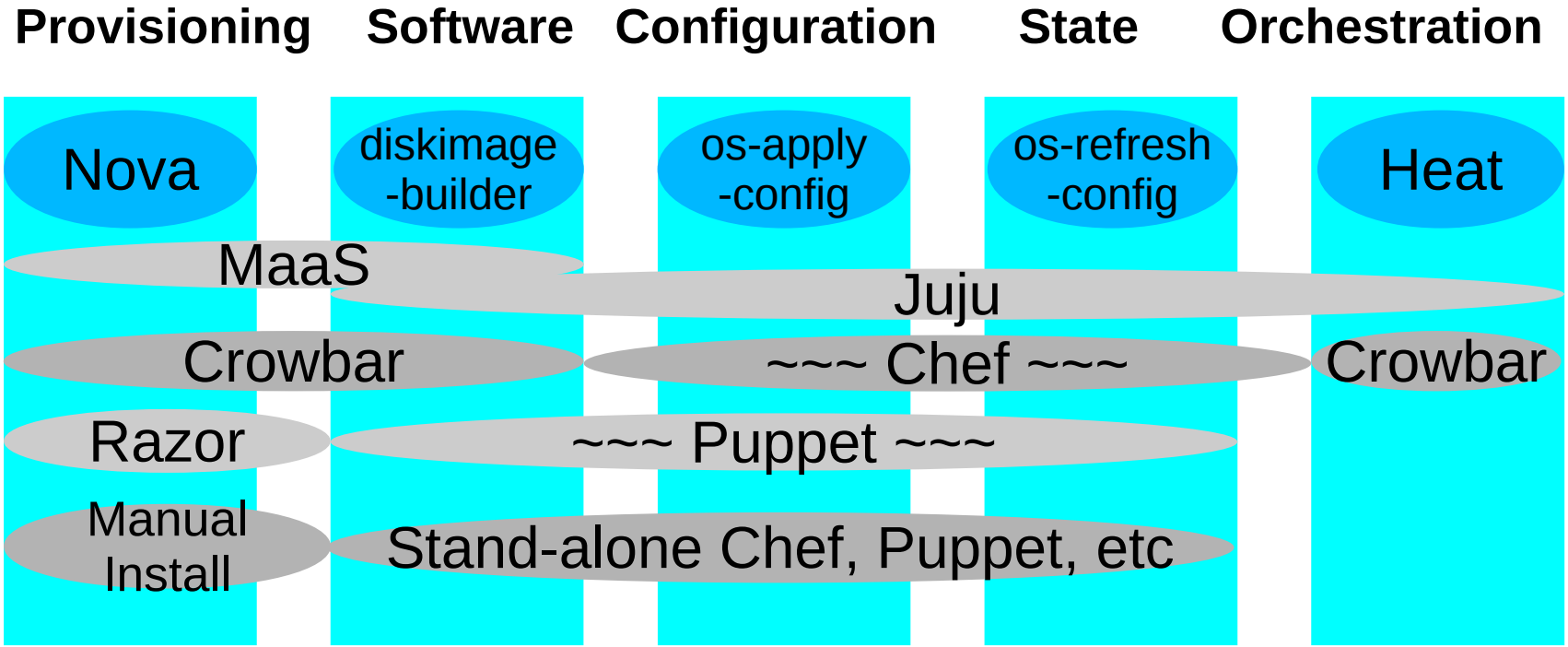
Install / Reconfigure / Upgrade		
Bugs	Cruft / Entropy	Hardware failure

TripleO: OpenStack on OpenStack

→ Continuous Integration and Delivery
→ Drive installation and maintenance costs down
→ Encapsulate the installation and upgrade process
→ Common API and infrastructure for above and below cloud

Install / Reconfigure / Upgrade		
Bugs	Cruft / Entropy	Hardware failure
CI/CD	Golden images	HA setup


Developer uploads a change
Jenkins determines which images are affected, builds them, and tests that change in isolation
build image
save result
deploy on bare metal & test the image
Bare Metal Nodes
use Heat to deploy to production
Bare Metal Nodes
Have confidence and publish the image(s)
Inject latest image into a Heat template, then test both deployment and upgrade.
start bootstrap cloud
and finally use Heat to orchetrate an upgrade of any image(s) affected by the change
load all necessary images, including the new image(s)
generate load and test it with Tempest
use Heat to orchestrate a multi-node baremetal deployment
Bare Metal Nodes
http://devananda.net/



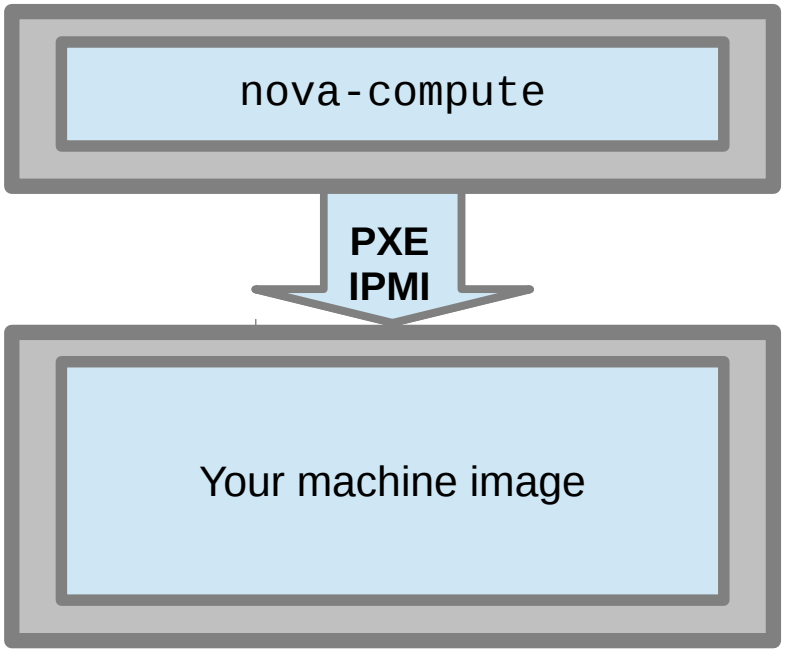
Components

→ Nova bare metal / Ironic™
→ Heat
→ diskimage-builder
→ os-apply-config
→ os-refresh-config
→ os-collect-config
→ tripleo-image-elements
→ tripleo-heat-templates

Deployment

→ Heat stack defines the cluster
→ Heat drives the Nova API to deliver images to machines
→ Virtual machines in developer test
→ Bare metal Nova for CI/CD and production deployment

Nova bare metal / Ironic™



Heat

→ Focus on orchestration
→ Supports any CM system within a machine
    → Use Puppet or Chef or Salt if you like
→ Delivers configuration metadata to machines
→ Accepts exported metadata from machines
→ OpenStack templates – tripleo-heat-templates

Golden Images

- → Encapsulate a known good set of software
- → Excludes *configuration* and *persistent state*
  - → these are placed on a separate state partition
  - → / is then mounted read-only
- → Equivalent of packages at a cluster level
- → Each image can be tested and then deployed as-is
  - → Because the configuration is not part of the image
- → Small focused toolchain to build images
  - → diskimage-builder
  - → tripleo-image-elements

os-*-config

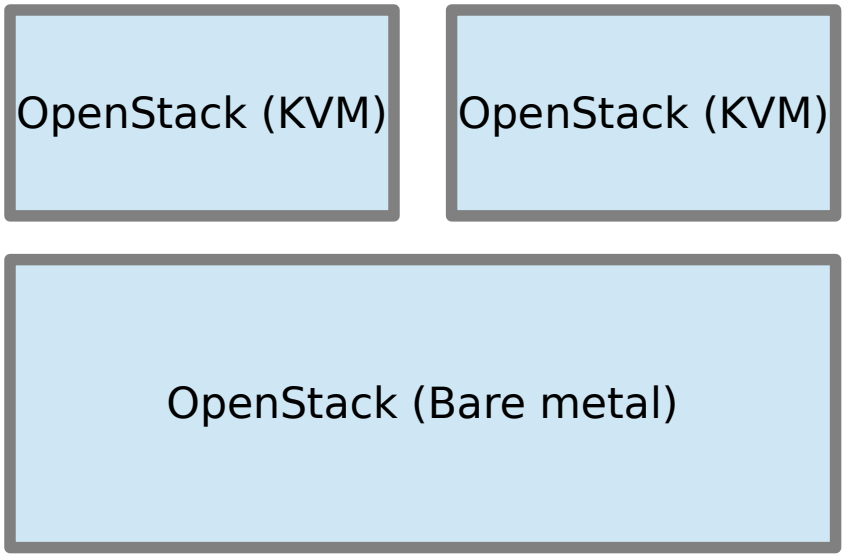
→ os-collect-config grabs new metadata from heat
os-refresh-config:
→ Quiesce fragile services
→ (If needed) Upgrade software from glance
→ (If needed) os-apply-config applies config files
→ (If needed) Reboot
→ Ensure required services are running and/or restarted
→ Perform any migrations (such as seeding initial data)
→ Notify heat that the deploy is complete on the machine

Performance

→ Installation code executes at Image Build time
→ FAST deployment

6 minutes from power off to working machine

Under and Over cloud



Nova cannot reliably run two different hypervisors in one cloud today

So we run two+ clouds:
→ the undercloud, a bare metal cloud that runs on, and owns, all the hardware
→ the overcloud, a regular VM based cloud running as a tenant on the bare metal cloud
→ additional VM clouds can run as parallel tenants on the undercloud (e.g. for testing).

Undercloud

→ Fully HA Bare metal OpenStack
→ Self hosted: nodes in the control plane are tenants within it
→ Aiming for as few as 2 machines for the control plane
→ All additional nodes are available for the overcloud tenant

Overcloud

→ Fully HA KVM based OpenStack hosted by the undercloud
→ Orchestrated by Heat running in the undercloud
→ Can (optionally) use the same disk images for most services

Installation

→ Special case of normal deployment
→ Run a collapsed cluster:
  – a single image with Heat + Nova Bare metal in a VM
→ Bridge that to the new data center network
→ Enroll the machines
→ Tell Heat that we want an HA configuration
→ Wait while it scale the undercloud out
→ Switch off the VM image
→ Tell Heat to recover from the loss of the VM node
  (by scaling out again)
→ Deploy the overcloud as a tenant

Upgrade, the simple version

→ HA/scale out services know how to react to dead nodes
→ Rolling upgrade:
  – heat deletes one node
  – heat creates new node with new version
  – lather, rinse, repeat

Upgrade, the complex version

→ Some nodes have precious data
→ Cloudify the install:
  – Node and root FS are ephemeral
  – Root FS specialization through cloud metadata
  – Precious data goes in attached volumes
→ Make the Root Image metadata passed in via heat
→ Upgrade changes the image id of the instance
→ takeovernode:
  – Download image
  – Mount in temp dir
  – Overwrite old Root FS with new Root FS
→ Services need to be restarted or node needs reboot:
  – Remember os-refresh-config?

Future engineering

→ cinder: Local volumes
→ neutron: Switch configuration
→ ironic: Support booting from newer kernels

Thank you