TripleO Undercloud deep-dive

James Slagle (jslagle@redhat.com)

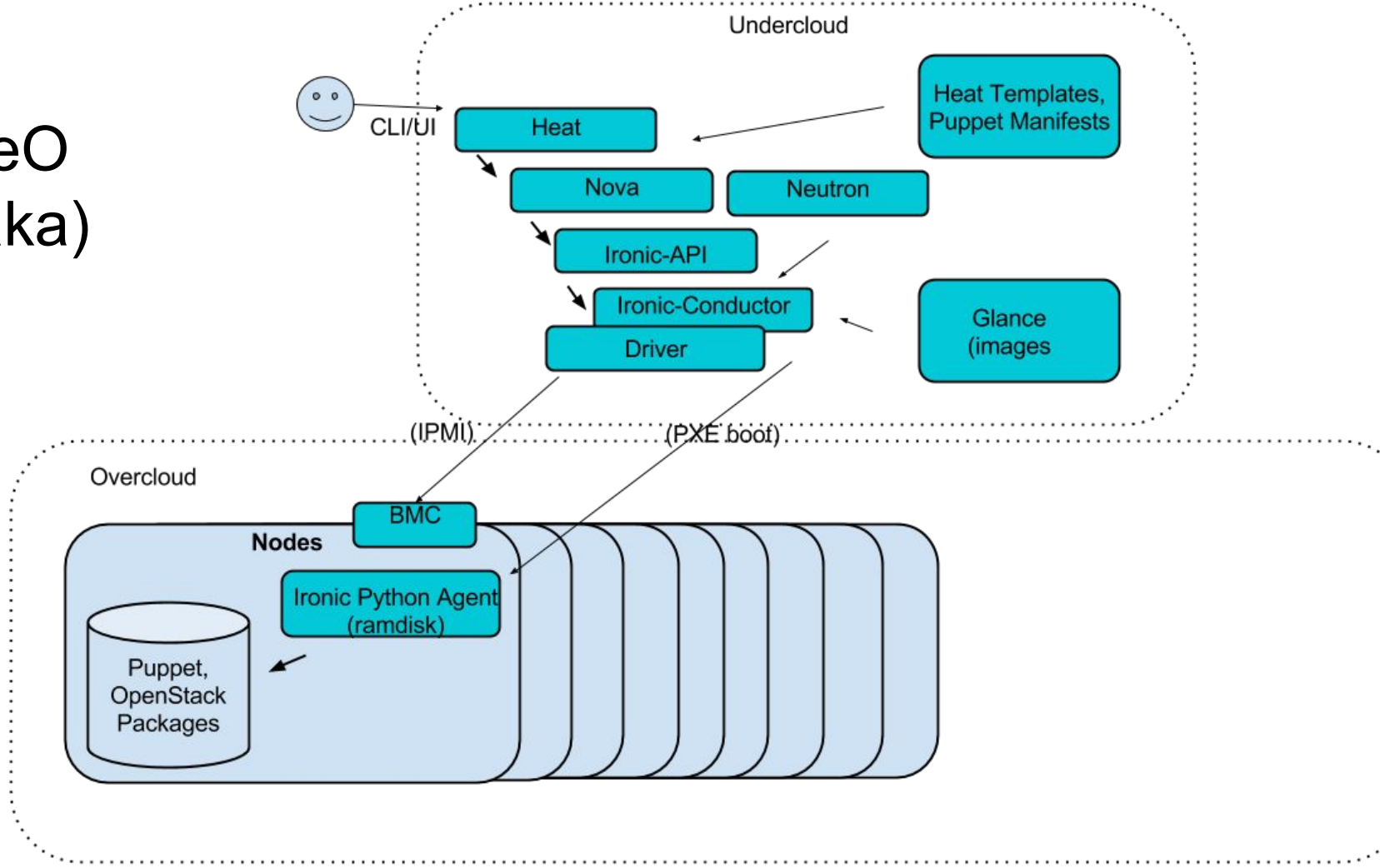
Steve Hardy (shardy@redhat.com)

Undercloud
What’s “under the hood”

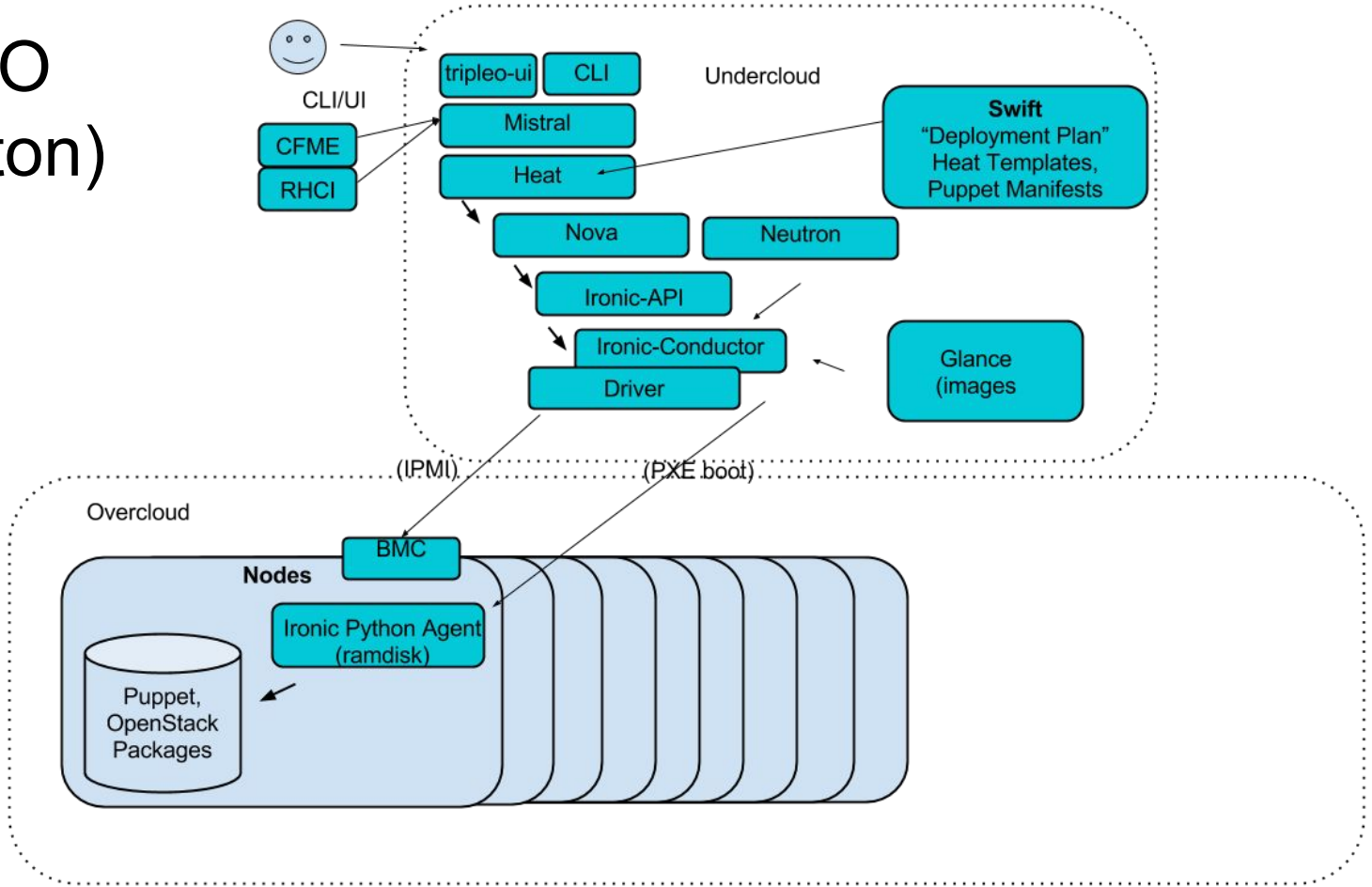
Undercloud components

- OpenStack on OpenStack
  - It’s “just” an all-in-one OpenStack node with Ironic enabled
- Configured via puppet (and a few scripts)
- **Heat** for Orchestration
- **Mistral** for Workflow/API
- **Zaqar** for messaging (UI/CLI uses this)
- **Nova** backed by **Ironic** for node deployment
- **Neutron** for networking (mostly IPAM)
- **Glance** stores images
- **Swift** stores data for overcloud deployments and undercloud services
- **Keystone** auth, but we don’t really do multi-user/tenant atm
- **Ironic Inspector** for node introspection (stores data in swift)

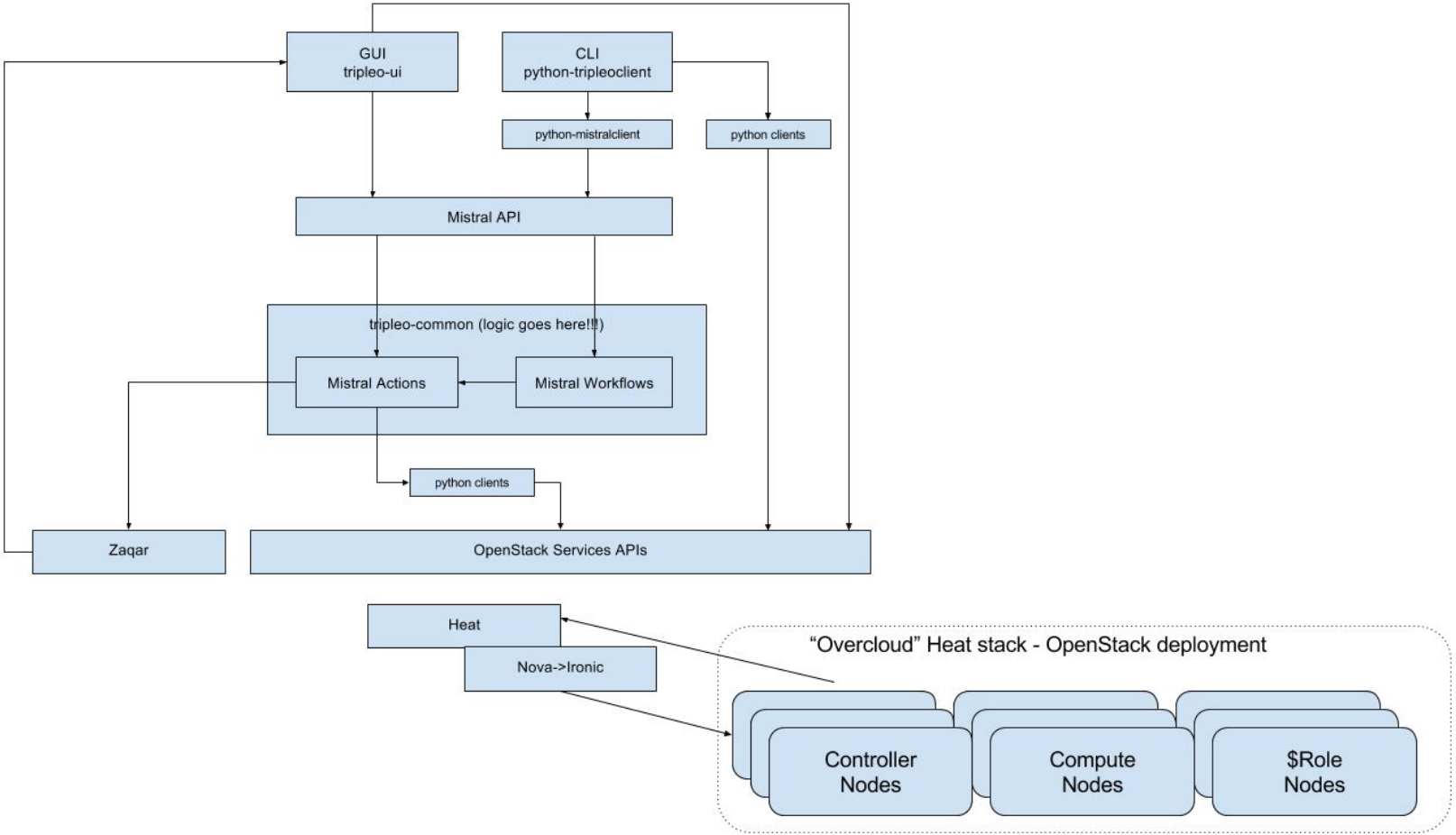
TripleO (Mitaka)



TripleO (Newton)



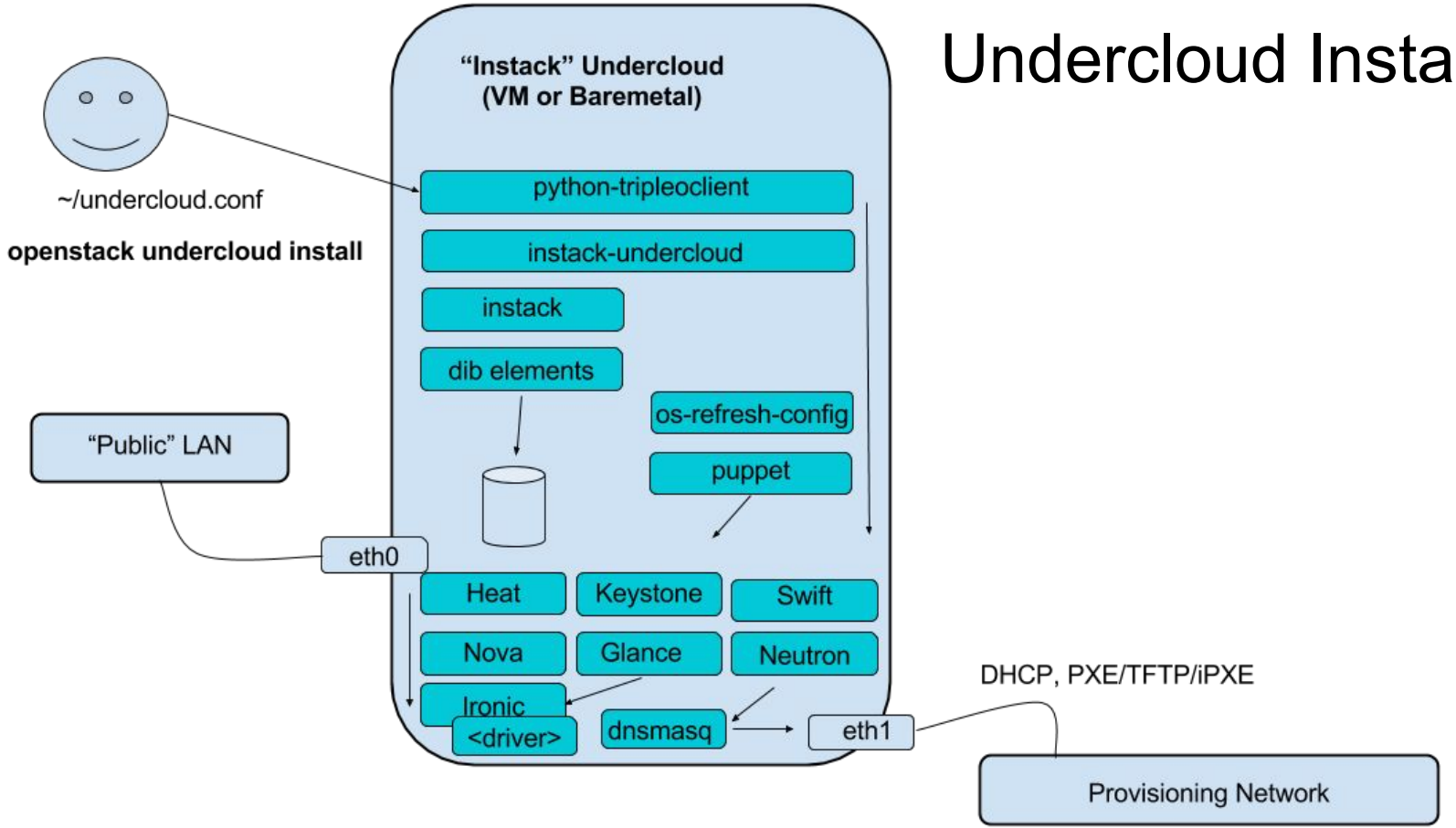
Mistral


GUI
tripleo-ui
CLI
python-tripleoclient
python-mistralclient
python clients
Mistral API
tripleo-common (logic goes here!!!)
Mistral Actions
Mistral Workflows
python clients
Zaqar
OpenStack Services APIs
Heat
Nova->Ironic
“Overcloud” Heat stack - OpenStack deployment
Controller Nodes
Compute Nodes
$Role Nodes

Mistral 101

- mistral action-list
  - Built-in actions based on python-*client
  - Tripleo specific actions in tripleo-common/actions
  - epi group show mistral.actions (pip install entry_point_inspector)
  - mistral-db-manage populate to add new actions
    - Puppet runs this during undercloud install
- mistral workflow-list
  - /usr/share/tripleo-common/workbooks/
  - Undercloud install deletes all workbooks and re-creates them
  - /usr/libexec/os-refresh-config/post-configure.d/98-undercloud-setup
- Testing is possible direct via mistralclient
  - mistral run-action tripleo.list_plans
  - mistral execution-create tripleo.plan_management.v1.create_default_deployment_plan '{"container":"overcloud_test"}'
  - swift list

Undercloud Install



Undercloud Install

- Undercloud install code is in the instack-undercloud repo
  - instack_undercloud/undercloud.py - defines options and main logic
  - elements/puppet-stack-config - defines puppet manifest and hiera
    - Pystache us used to translate options to hiera values
- A few things (e.g ipxe) are still configured via elements/*
  - Moving away from this in favor of puppet
- **/usr/libexec/os-refresh-config/configure.d/50-puppet-stack-config**
  - You can run this manually (and add --debug) to figure out puppet problems
- **/usr/libexec/os-refresh-config/post-configure.d/98-undercloud-setup**
  - Some post-install things are scripted outside of puppet here

Deep dive, instack, puppet, networking?

Configuration Flow

- Tooling same/similar to overcloud
  - DiB elements used, but applied locally
  - Puppet applied via os-refresh-config
- Puppet hieradata generated based on template
  - instack-undercloud/elements/puppet-stack-config/
- Tripleoclient calls instack-undercloud undercloud.py to install
- Scope for simplifying here?

Customizing Configuration

- undercloud.conf
- hieradata_override
  - Can set any hieradata to influence the configuration applied by puppet
  - No need to gratuitously add options to undercloud.conf
- net_config_override
  - Configure a different networking setup from the default (single nic joined to an ovs bridge)

os-net-config

- Python tool used to configure networking on the undercloud and overcloud
- Uses a declarative approach to describe the desired networking configuration (via yaml or JSON)
- Persists configuration via ifcfg or eni
- https://github.com/openstack/os-net-config/tree/master/etc/os-net-config/samples
- https://github.com/openstack/os-net-config

Where to get help?

- http://docs.openstack.org/developer/tripleo-docs
- http://docs.openstack.org/developer/heat/template_guide/
- #tripleo on Freenode
- #heat on Freenode
- openstack-dev ML
- Bugs (upstream) http://bugs.launchpad.net/tripleo

Thanks/Questions

- http://docs.openstack.org/developer/heat/template_guide/
- http://docs.openstack.org/developer/tripleo-docs/
- http://hardysteven.blogspot.co.uk/
- https://github.com/openstack/heat
- https://github.com/openstack/heat-templates
- https://github.com/openstack/tripleo-heat-templates
- https://github.com/hardys/presentations
- https://github.com/hardys/demo_templates/
- https://ask.openstack.org/en/questions/
- #heat on Freenode
- openstack@lists.openstack.org (OpenStack general/users ML)