

ESI: Bringing multi-tenancy to Ironic

What is ESI?

“We want to create a set of services to permit multiple tenants to flexibly allocate baremetal machines from a pool of available hardware, create networks, attach baremetal nodes and networks, and to optionally provision an operating system on those systems.”

- Allow hardware owners to maintain control
- Allow hardware consumers flexible self-provisioning

Where are we trying to get to?

- Most OpenStack services are “multi-tenant”: resources are owned by a project and cannot be seen by members of other projects.
- Ironic has an “admin or nothing” model: a user with admin privileges can do everything, and a non-admin user can’t do anything.
- In order to support the workflows we envision for ESI, we need multi-tenancy at the hardware layer.

Multi-Tenant Ironic

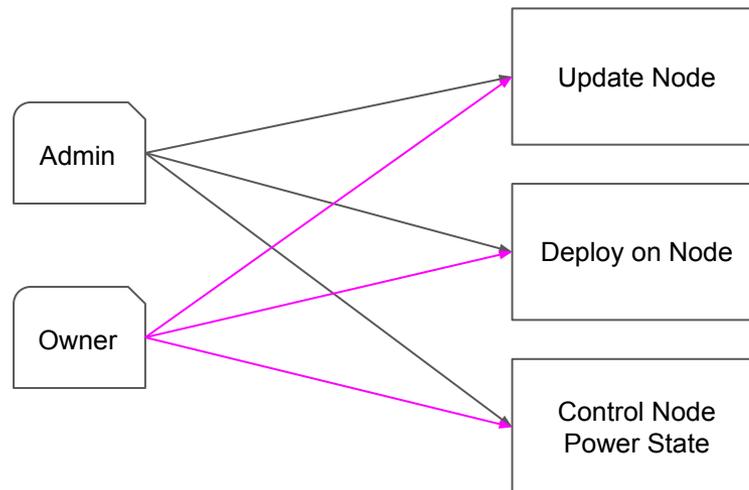
Working towards multi-tenant support

We are implementing support for multi-tenancy in Ironic in stages:

- Enable node owners to control nodes
- Introduce the concept of a node lessee to Ironic
- Tweak node deployment through the Ironic API
- Allow Ironic to reserve nodes based on owner/lessee

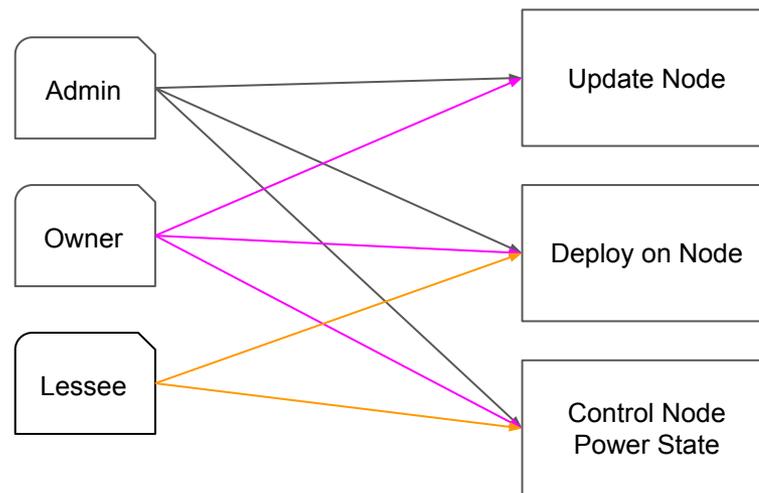
Enable node owner to control nodes

- Prior to our ESI work, Ironic had an “owner” field that was informational but did not have any operational significance.
- Our changes enable policy decisions based on the owner field
- Owner-based access control must be explicitly enabled
- Cloud operator can now delegate control of nodes to node owners



Introduce node lessee to Ironic

- Permits an owner to delegate some control of their nodes to another project.
- Our changes enable policy decisions based on the lessee field.
- A node lessee can have limited Ironic API access to leased hardware.



Example: Updated Ironic Policy File

```
# Owner of node

"is_node_owner": "project_id:%(node.owner)s"

# Lessee of node

"is_node_lessee": "project_id:%(node.lessee)s"

# Update Node records

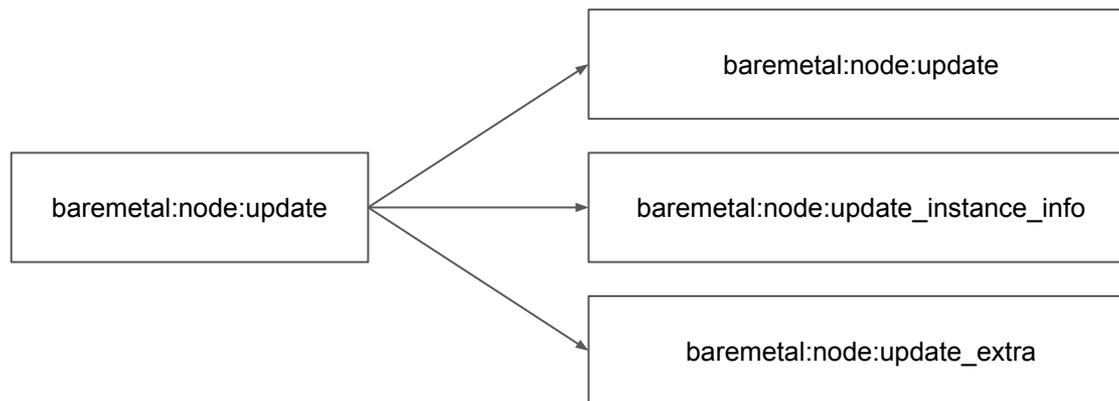
"baremetal:node:update": "rule:is_admin or rule:is_node_owner"

# Change Node power status

"baremetal:node:set_power_state": "rule:is_admin or
rule:is_node_owner or rule:is_node_lessee"
```

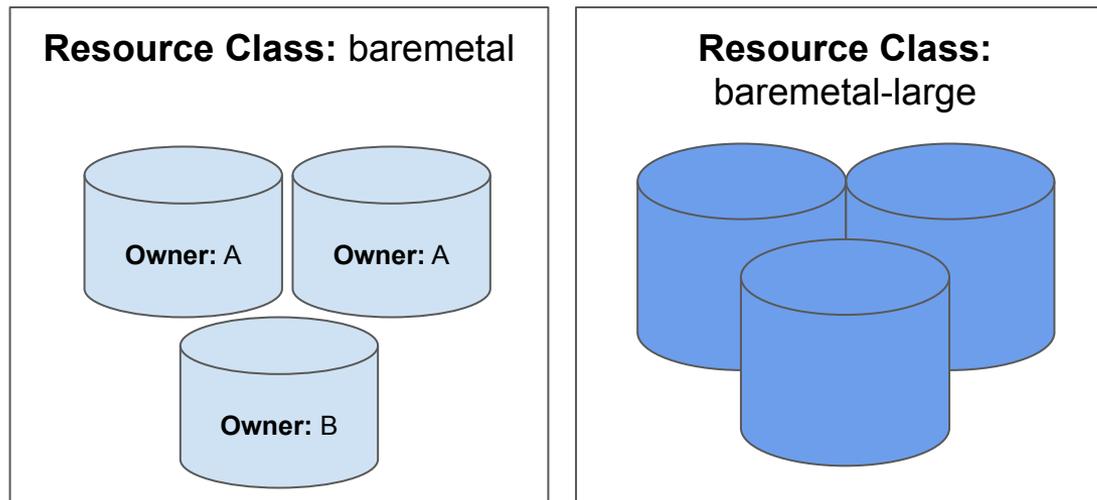
Tweak node deployment through the Ironic API

- It is possible to provision baremetal nodes using standalone Ironic through a complex series of API commands.
- However one of those commands allows a user to update any node attribute - not appropriate for a lessee.
- Necessary to create additional policy rules to govern attributes that may can be updated by a lessee during deployment (such as the image to be used).



Allow Ironic to reserve nodes based on owner/lessee

- Ironic permits node reservation through the Allocations API.
 - `openstack baremetal allocation create --resource-class baremetal`
- This change allows non-admin use of Allocations: a node owner or lessee will only be assigned their own hardware.



Metalsmith

- Client-side Python library used for provisioning Ironic nodes.
- Uses Ironic Allocations and APIs
- Integrated with Neutron and Glance
 - `metalsmith deploy --image centos7 --network provisioning --ssh-public-key ~/.ssh/id_rsa.pub --resource-class baremetal`
- “Just works” with multi-tenant changes

State of Code

- **MERGED:** Allow node owners to administer nodes
- **MERGED:** Add owner to allocations and create relevant policies
- **MERGED:** Restrict ability to change owner on provisioned or allocated node
- **MERGED:** Add allocation owner
- **MERGED:** Allow node owners to administer associated ports
- **PARTIAL APPROVAL:** Add node lessee field
- **PARTIAL APPROVAL:** Expose allocation owner to additional policy checks
- **UNDER REVIEW:** Add separate policies for updating node instance_info and extra
- **UNDER REVIEW:** Use auth values from neutron conf when managing Neutron ports

What's Next?

- Upstream Ironic
 - Create a Deployment API so that a client-side library like Metalsmith is no longer necessary
- ESI
 - Leasing service: Provide a convenient mechanism for owners to make hardware available for others to lease.
 - Networking: Make it easier to attach tenant networks to baremetal nodes.
 - Installation documentation: Explain how to configure OpenStack services to enable ESI functionality.

THANK YOU



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



twitter.com/RedHat

