Welcome to Ironic's developer documentation!

Introduction

Ironic is an OpenStack project which provisions bare metal (as opposed to virtual) machines. It may be used independently or as part of an OpenStack Cloud, and integrates with the OpenStack Identity (keystone), Compute (nova), Network (neutron), Image (glance), and Object (swift) services.

The Bare Metal service manages hardware through both common (eg. PXE and IPMI) and vendor-specific remote management protocols. It provides the cloud operator with a unified interface to a heterogeneous fleet of servers while also providing the Compute service with an interface that allows physical servers to be managed as though they were virtual machines.

An introduction to ironic's conceptual architecture is available for those new to the project.

Site Notes

This site is primarily intended to provide documentation for developers interested in contributing to or working with ironic. It *also* contains references and guides for administrators which are not yet hosted elsewhere on the OpenStack documentation sites.

This documentation is continually updated and may not represent the state of the project at any specific prior release. To access documentation for a previous release of ironic, append the OpenStack release name to the URL, for example:

> http://docs.openstack.org/developer/ironic/mitaka/

Bare Metal API References

Ironic's REST API has changed since its first release, and continues to evolve to meet the changing needs of the community. Here we provide a conceptual guide as well as more detailed reference documentation.

- API Concept Guide
- API Reference (latest)
- API Version History

Developer's Guide

Getting Started

If you are new to ironic, this section contains information that should help you get started as a developer working on the project or contributing to the project.

- Developer Contribution Guide
- Setting Up Your Development Environment
- Frequently Asked Questions

The following pages describe the architecture of the Bare Metal service and may be helpful to anyone working on or with the service, but are written primarily for developers.

- Ironic System Architecture
- Provisioning State Machine
- Notifications

These pages contain information for PTLs, cross-project liaisons, and core reviewers.

- Releasing Ironic Projects
- Ironic Governance Structure

Writing Drivers

Ironic's community includes many hardware vendors who contribute drivers that enable more advanced functionality when Ironic is used in conjunction with that hardware. To do this, the Ironic developer community is committed to standardizing on a Python Driver API that meets the common needs of all hardware vendors, and evolving this API without breaking backwards compatibility. However, it is sometimes necessary for driver authors to implement functionality - and expose it through the REST API - that can not be done through any existing API.

To facilitate that, we also provide the means for API calls to be "passed through" ironic and directly to the driver. Some guidelines on how to implement this are provided below. Driver authors are strongly encouraged to talk with the developer community about any implementation using this functionality.

- Driver Overview
- Writing "vendor_passthru" methods

Testing Network Integration

In order to test the integration between the Bare Metal and Networking services, support has been added to devstack to mimic an external physical switch. Here we include a recommended configuration for devstack to bring up this environment.

- Configuring Devstack for multitenant network testing

Administrator's Guide

Installation & Operations

If you are a system administrator running Ironic, this section contains information that should help you understand how to deploy, operate, and upgrade the services.

- Installation Guide
- Upgrade Guide
- Release Notes
- Troubleshooting FAQ

Configuration

There are many aspects of the Bare Metal service which are environment specific. The following pages will be helpful in configuring specific aspects of ironic that may or may not be suitable to every situation.

- Guide to Node Cleaning
- Configuring Node Inspection
- Configuring RAID during deployment
- Security considerations for your Bare Metal installation
- Adopting Nodes in an ACTIVE state
- Auditing API Traffic
- Configuring for Multi-tenant Networking
- Configuring node web or serial console
- Emitting software metrics
- Configuration Reference
- Sample configuration file

Dashboard Integration

A plugin for the OpenStack Dashboard (horizon) service is under development. Documentation for that can be found within the ironic-ui project.

- Dashboard (horizon) plugin

Driver References

Every driver author is expected to document the use and configuration of their driver. These pages are linked below.

- Driver Documentation pages
  - Ironic-Python-Agent (agent)
  - IPMITool driver
  - DRAC driver
  - AMT driver
  - SNMP driver
  - iLO driver
  - SeaMicro driver
  - iRMC driver
  - VirtualBox driver
  - Cisco UCS driver
  - Wake-On-Lan driver
  - iBoot driver
  - CIMC driver
  - OneView driver
  - XenServer ssh driver
- Further Considerations for the Agent Drivers
  - Overview
  - Drivers
  - Using proxies for image download in agent drivers
  - Advanced configuration

Command References

Here are references for commands not elsewhere documented.

- ironic-dbsync

Indices and tables

- Index
- Module Index
- Search Page