Using Bare Metal service as a standalone service

« (enabling-https.html) | » (configdrive.html) | 🐛 (https://bugs.launchpad.net/ironic/+filebug?
field.title=Using%20Bare%20Metal%20service%20as%20a%20standalone%20service%20in%20Installation%20Guide%20for%20Bare%20Metal%20Service&field.comm
----------------------------------%0ARelease:%200.1%20on%202016-11-
09%2020:35%0ASHA:%20e3bedc4eadbafbf11c1e26a216e9d40a1839a838%0ASource:%20http://git.openstack.org/cgit/openstack/ironic/tree/install-
guide/source/standalone.rst%0AURL: http://docs.openstack.org/project-install-guide/baremetal/draft/standalone.html&field.tags=install-guide)

UPDATED: 2016-11-09 20:35

Contents (index.html)

Starting with the Kilo release, it's possible to use Bare Metal service without other OpenStack services.

You should make the following changes to `/etc/ironic/ironic.conf` :

1. To disable usage of Identity service tokens:

```
[DEFAULT]
...
auth_strategy=noauth
```

2. If you want to disable the Networking service, you should have your network pre-configured to serve DHCP and TFTP for machines that you're deploying. To disable it, change the following lines:

```
[dhcp]
...
dhcp_provider=none
```

> **❂ Note**
>
> If you disabled the Networking service and the driver that you use is supported by at most one conductor, PXE boot will still work for your nodes without any manual config editing. This is because you know all the DHCP options that will be used for deployment and can set up your DHCP server appropriately.
>
> If you have multiple conductors per driver, it would be better to use Networking since it will do all the dynamically changing configurations for you.

If you don't use Image service, it's possible to provide images to Bare Metal service via hrefs.

> **❂ Note**
>
> At the moment, only two types of hrefs are acceptable instead of Image service UUIDs: HTTP(S) hrefs (for example, "http://my.server.net/images/img (http://my.server.net/images/img)") and file hrefs (file:///images/img (file:///images/img)).

There are however some limitations for different drivers:

- If you're using one of the drivers that use agent deploy method (namely, `agent_ilo` , `agent_ipmitool` , `agent_pyghmi` , `agent_ssh` or `agent_vbox`) you have to know MD5 checksum for your instance image. To compute it, you can use the following command:

```
md5sum image.qcow2
ed82def8730f394fb85aef8a208635f6  image.qcow2
```

  Apart from that, because of the way the agent deploy method works, image hrefs can use only HTTP(S) protocol.

- If you're using `iscsi_ilo` or `agent_ilo` driver, Object Storage service is required, as these drivers need to store floppy image that is used to pass parameters to deployment iso. For this method also only HTTP(S) hrefs are acceptable, as HP iLO servers cannot attach other types of hrefs as virtual media.

- Other drivers use PXE deploy method and there are no special requirements in this case.

Steps to start a deployment are pretty similar to those when using Compute:

1. To use the ironic CLI (http://docs.openstack.org/developer/python-ironicclient/cli.html), set up these environment variables. Since no authentication strategy is being used, the value can be any string for OS_AUTH_TOKEN. IRONIC_URL is the URL of the ironic-api process. For example:

```
export OS_AUTH_TOKEN=fake-token
export IRONIC_URL=http://localhost:6385/
```

2. Create a node in Bare Metal service. At minimum, you must specify the driver name (for example, "pxe_ipmitool"). You can also specify all the required driver parameters in one command. This will return the node UUID:

```
ironic node-create -d pxe_ipmitool -i ipmi_address=ipmi.server.net \
-i ipmi_username=user -i ipmi_password=pass \
-i deploy_kernel=file:///images/deploy.vmlinuz \
-i deploy_ramdisk=http://my.server.net/images/deploy.ramdisk

+--------------+----------------------------------------------------------------------+
| Property     | Value                                                                |
+--------------+----------------------------------------------------------------------+
| uuid         | be94df40-b80a-4f63-b92b-e9368ee8d14c                                 |
| driver_info  | {u'deploy_ramdisk': u'http://my.server.net/images/deploy.ramdisk',   |
|              | u'deploy_kernel': u'file:///images/deploy.vmlinuz', u'ipmi_address': |
|              | u'ipmi.server.net', u'ipmi_username': u'user', u'ipmi_password':    |
|              | u'******'}                                                           |
| extra        | {}                                                                   |
| driver       | pxe_ipmitool                                                        |
| chassis_uuid |                                                                     |
| properties   | {}                                                                  |
+--------------+----------------------------------------------------------------------+
```

Note that here deploy_kernel and deploy_ramdisk contain links to images instead of Image service UUIDs.

3. As in case of Compute service, you can also provide `capabilities` to node properties, but they will be used only by Bare Metal service (for example, boot mode). Although you don't need to add properties like `memory_mb`, `cpus` etc. as Bare Metal service will require UUID of a node you're going to deploy.

4. Then create a port to inform Bare Metal service of the network interface cards which are part of the node by creating a port with each NIC's MAC address. In this case, they're used for naming of PXE configs for a node:

```
ironic port-create -n $NODE_UUID -a $MAC_ADDRESS
```

5. As there is no Compute service flavor and instance image is not provided with nova boot command, you also need to specify some fields in `instance_info`. For PXE deployment, they are `image_source`, `kernel`, `ramdisk`, `root_gb`:

```
ironic node-update $NODE_UUID add instance_info/image_source=$IMG \
instance_info/kernel=$KERNEL instance_info/ramdisk=$RAMDISK \
instance_info/root_gb=10
```

Here $IMG, $KERNEL, $RAMDISK can also be HTTP(S) or file hrefs. For agent drivers, you don't need to specify kernel and ramdisk, but MD5 checksum of instance image is required:

```
ironic node-update $NODE_UUID add instance_info/image_checksum=$MD5HASH
```

6. Validate that all parameters are correct:

```
ironic node-validate $NODE_UUID

+------------+--------+----------------------------------------------------------------+
| Interface  | Result | Reason                                                         |
+------------+--------+----------------------------------------------------------------+
| console    | False  | Missing 'ipmi_terminal_port' parameter in node's driver_info.  |
| deploy     | True   |                                                                |
| management | True   |                                                                |
| power      | True   |                                                                |
+------------+--------+----------------------------------------------------------------+
```

7. Now you can start the deployment, run:

```
ironic node-set-provision-state $NODE_UUID active
```

You can manage provisioning by issuing this command. Valid provision states are `active`, `rebuild` and `deleted`.

For iLO drivers, fields that should be provided are:

- `ilo_deploy_iso` under `driver_info`;
- `ilo_boot_iso`, `image_source`, `root_gb` under `instance_info`.

> ⊘ **Note**

> Before Liberty release Ironic was not able to track non-Glance images' content changes. Starting with Liberty, it is possible to do so using image modification date. For example, for HTTP image, if 'Last-Modified' header value from response to a HEAD request to "http://my.server.net/images/deploy.ramdisk (http://my.server.net/images/deploy.ramdisk)" is greater than cached image modification time, Ironic will re-download the content. For "file:// (file://)" images, the file system modification time is used.

Other references¶

- Enabling local boot without Compute (advanced.html#local-boot-without-compute)

« (enabling-https.html) » (configdrive.html) 🐞 (https://bugs.launchpad.net/ironic/+filebug?field.title=Using%20Bare%20Metal%20service%20as%20a%20standalone%20service%20in%20Installation%20Guide%20for%20Bare%20Metal%20Service&field.com
----------------------------------%0ARelease:%200.1%20on%202016-11-09%2020:35%0ASHA:%20e3bedc4eadbafbf11c1e26a216e9d40a1839a838%0ASource:%20http://git.openstack.org/cgit/openstack/ironic/tree/install-guide/source/standalone.rst%0AURL: http://docs.openstack.org/project-install-guide/baremetal/draft/standalone.html&field.tags=install-guide)

UPDATED: 2016-11-09 20:35

🐞 FOUND AN ERROR? REPORT A BUG (HTTPS://BUGS.LAUNCHPAD.NET/IRONIC/+FILEBUG?
FIELD.TITLE=USING%20BARE%20METAL%20SERVICE%20AS%20A%20STANDALONE%20SERVICE%20IN%20INSTALLATION%20GUIDE%20FOR%20BARE%20METAL%20SERVICE&FIELD.COMMENT=%0A%0A-----
------------------------------%0ARELEASE:%200.1%20ON%202016-11-
09%2020:35%0ASHA:%20E3BEDC4EADBAFBF11C1E26A216E9D40A1839A838%0ASOURCE:%20HTTP://GIT.OPENSTACK.ORG/CGIT/OPENSTACK/IRONIC/TREE/INSTALL-
GUIDE/SOURCE/STANDALONE.RST%0AURL: HTTP://DOCS.OPENSTACK.ORG/PROJECT-INSTALL-GUIDE/BAREMETAL/DRAFT/STANDALONE.HTML&FIELD.TAGS=INSTALL-GUIDE)

❓ QUESTIONS? (HTTP://ASK.OPENSTACK.ORG)

⊕

OpenStack Documentation ▾

Contents

(index.html)

OpenStack

- Projects (http://openstack.org/projects/)
- OpenStack Security (http://openstack.org/projects/openstack-security/)
- Common Questions (http://openstack.org/projects/openstack-faq/)
- Blog (http://openstack.org/blog/)
- News (http://openstack.org/news/)

Community

- User Groups (http://openstack.org/community/)
- Events (http://openstack.org/community/events/)
- Jobs (http://openstack.org/community/jobs/)
- Companies (http://openstack.org/foundation/companies/)
- Contribute (http://docs.openstack.org/infra/manual/developers.html)