Enabling the configuration drive (configdrive)

« (standalone.html) » (deploy-ramdisk.html) (https://bugs.launchpad.net/ironic/+filebug?field.title=Enabling%20the%20configuration%20drive%20(configdrive)%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/configdrive.rst%0AURL: http://docs.openstack.org/project-install-guide/baremetal/draft/configdrive.html&field.tags=install-guide)

UPDATED: 2016-11-09 20:35

Contents (index.html)

- When used with Compute service
- When used standalone
- Accessing the configuration drive data
- Cloud-init integration

Starting with the Kilo release, the Bare Metal service supports exposing a configuration drive image to the instances.

The configuration drive is used to store instance-specific metadata and is present to the instance as a disk partition labeled `config-2` . The configuration drive has a maximum size of 64MB. One use case for using the configuration drive is to expose a networking configuration when you do not use DHCP to assign IP addresses to instances.

The configuration drive is usually used in conjunction with the Compute service, but the Bare Metal service also offers a standalone way of using it. The following sections will describe both methods.

When used with Compute service¶

To enable the configuration drive for a specific request, pass `--config-drive true` parameter to the `nova boot` command, for example:

```
nova boot --config-drive true --flavor baremetal --image test-image instance-1
```

It's also possible to enable the configuration drive automatically on all instances by configuring the `OpenStack Compute service` to always create a configuration drive by setting the following option in the `/etc/nova/nova.conf` file, for example:

```
[DEFAULT]
...

force_config_drive=True
```

In some cases, you may wish to pass a user customized script when deploying an instance. To do this, pass `--user-data /path/to/file` to the `nova boot` command. More information can be found at Provide user data to instances (http://docs.openstack.org/user-guide/cli_provide_user_data_to_instances.html)

When used standalone¶

When used without the Compute service, the operator needs to create a configuration drive and provide the file or HTTP URL to the Bare Metal service.

For the format of the configuration drive, Bare Metal service expects a `gzipped` and `base64` encoded ISO 9660 [*] file with a `config-2` label. The ironic client (http://docs.openstack.org/developer/python-ironicclient/) can generate a configuration drive in the expected format (http://docs.openstack.org/user-guide/cli_config_drive.html#openstack-metadata-format). Just pass a directory path containing the files that will be injected into it via the `--config-drive` parameter of the `node-set-provision-state` command, for example:

```
ironic node-set-provision-state --config-drive /dir/configdrive_files $node_identifier active
```

Accessing the configuration drive data¶

When the configuration drive is enabled, the Bare Metal service will create a partition on the instance disk and write the configuration drive image onto it. The configuration drive must be mounted before use. This is performed automatically by many tools, such as cloud-init and cloudbase-init. To mount it manually on a Linux distribution that supports accessing devices by labels, simply run the following:

```
mkdir -p /mnt/config
mount /dev/disk/by-label/config-2 /mnt/config
```

If the guest OS doesn't support accessing devices by labels, you can use other tools such as `blkid` to identify which device corresponds to the configuration drive and mount it, for example:

```
CONFIG_DEV=$(blkid -t LABEL="config-2" -odevice)
mkdir -p /mnt/config
mount $CONFIG_DEV /mnt/config
```

[1] A config drive could also be a data block with a VFAT filesystem on it instead of ISO 9660. But it's unlikely that it would be needed since ISO 9660 is widely supported across operating systems.

Cloud-init integration¶

The configuration drive can be especially useful when used with cloud-init (http://cloudinit.readthedocs.org/en/latest/topics/datasources.html#config-drive), but in order to use it we should follow some rules:

- `Cloud-init` data should be organized in the expected format (http://docs.openstack.org/user-guide/cli_config_drive.html#openstack-metadata-format).
- Since the Bare Metal service uses a disk partition as the configuration drive, it will only work with cloud-init version >= 0.7.5 (http://bazaar.launchpad.net/~cloud-init-dev/cloud-init/trunk/view/head:/ChangeLog).
- `Cloud-init` has a collection of data source modules, so when building the image with disk-image-builder (http://docs.openstack.org/developer/diskimage-builder/) we have to define `DIB_CLOUD_INIT_DATASOURCES` environment variable and set the appropriate sources to enable the configuration drive, for example:

  ```
  DIB_CLOUD_INIT_DATASOURCES="ConfigDrive, OpenStack" disk-image-create -o fedora-cloud-image fedora baremetal
  ```

  For more information see how to configure cloud-init data sources (http://docs.openstack.org/developer/diskimage-builder/elements/cloud-init-datasources/README.html).

« (standalone.html) » (deploy-ramdisk.html) (https://bugs.launchpad.net/ironic/+filebug?field.title=Enabling%20the%20configuration%20drive%20(configdrive)%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/configdrive.rst%0AURL: http://docs.openstack.org/project-install-guide/baremetal/draft/configdrive.html&field.tags=install-guide)

UPDATED: 2016-11-09 20:35

OpenStack Documentation

Contents

(index.html)

- Bare Metal service overview (get_started.html)
- Install and configure the Bare Metal service (install.html)
- Integration with other OpenStack services (configure-integration.html)
- Configure the Bare Metal service for cleaning (configure-cleaning.html)
- Configure tenant networks (configure-tenant-networks.html)
- Enrollment (enrollment.html)
- Enabling HTTPS (enabling-https.html)
- Using Bare Metal service as a standalone service (standalone.html)
- Enabling the configuration drive (configdrive) ()
  - When used with Compute service
  - When used standalone
  - Accessing the configuration drive data
  - Cloud-init integration
- Building or downloading a deploy ramdisk image (deploy-ramdisk.html)
- Setup the drivers for the Bare Metal service (setup-drivers.html)
- Advanced features (advanced.html)
- Troubleshooting (troubleshooting.html)
- Next steps (next-steps.html)