

ceph-ansible

Ansible playbooks for Ceph, the distributed filesystem.

Installation

github

You can install directly from the source on github by following these steps:

- Clone the repository:

```
git clone https://github.com/ceph/ceph-ansible.git
```

- Next, you must decide which branch of `ceph-ansible` you wish to use. There are stable branches to choose from or you could use the master branch:

```
git checkout $branch
```

Ansible on RHEL and CentOS

You can acquire Ansible on RHEL and CentOS by installing from [Extras](#).

On RHEL:

```
subscription-manager repos --enable=rhel-7-server-extras-rpms
```

(CentOS does not use subscription-manager and already has “Extras” enabled by default.)

```
sudo yum install ansible
```

Ansible on Ubuntu

You can acquire Ansible on Ubuntu by using the [Ansible PPA](#).

```
sudo add-apt-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
```

Releases

The following branches should be used depending on your requirements. The `stable-*` branches have been QE tested and sometimes receive backport fixes throughout their lifecycle. The `master` branch should be considered experimental and used with caution.

- `stable-2.1` Support for ceph version `jewel`. This branch supports ansible versions `2.1` and `2.2.1`.
- `stable-2.2` Support for ceph versions `jewel` and `luminous`. This branch supports ansible versions `2.1` and `2.2.2`.
- `stable-3.0` Support for ceph versions `jewel` and `luminous`. This branch supports ansible versions `2.3.1`, `2.3.2` and `2.4.2`.
- `master` Support for ceph versions `jewel`, and `luminous`. This branch supports ansible version `2.4.2`.

Configuration and Usage

This project assumes you have a basic knowledge of how ansible works and have already prepared your hosts for configuration by ansible.

After you've cloned the `ceph-ansible` repository, selected your branch and installed ansible then you'll need to create your inventory file, playbook and configuration for your ceph cluster.

Inventory

The ansible inventory file defines the hosts in your cluster and what roles each host plays in your ceph cluster. The default location for an inventory file is `/etc/ansible/hosts` but this file can be placed anywhere and used with the `-i` flag of `ansible-playbook`. An example inventory file would look like:

```
[mons]
mon1
mon2
mon3

[osds]
osd1
osd2
osd3
```

Note:

For more information on ansible inventories please refer to the ansible documentation: http://docs.ansible.com/ansible/latest/intro_inventory.html

Playbook

You must have a playbook to pass to the `ansible-playbook` command when deploying your cluster. There is a sample playbook at the root of the `ceph-ansible` project called `site.yml.sample`. This playbook should work fine for most usages, but it does include by default every daemon group which might not be appropriate for your cluster setup. Perform the following steps to prepare your playbook:

- Rename the sample playbook: `mv site.yml.sample site.yml`
- Modify the playbook as necessary for the requirements of your cluster

Note:

It's important the playbook you use is placed at the root of the `ceph-ansible` project. This is how ansible will be able to find the roles that `ceph-ansible` provides.

ceph-ansible - choose installation method

Ceph can be installed through several methods.

- [Installation methods](#)

ceph-ansible Configuration

The configuration for your ceph cluster will be set by the use of ansible variables that `ceph-ansible` provides. All of these options and their default values are defined in the `group_vars/` directory at the root of the `ceph-ansible` project. Ansible will use configuration in a `group_vars/` directory that is relative to your inventory file or your playbook. Inside of the

`group_vars/` directory there are many sample ansible configuration files that relate to each of the ceph daemon groups by their filename. For example, the `osds.yml.sample` contains all the default configuration for the OSD daemons. The `all.yml.sample` file is a special `group_vars` file that applies to all hosts in your cluster.

Note:

For more information on setting group or host specific configuration refer to the ansible documentation:

http://docs.ansible.com/ansible/latest/intro_inventory.html#splitting-out-host-and-group-specific-data

At the most basic level you must tell `ceph-ansible` what version of ceph you wish to install, the method of installation, your clusters network settings and how you want your OSDs configured. To begin your configuration rename each file in `group_vars/` you wish to use so that it does not include the `.sample` at the end of the filename, uncomment the options you wish to change and provide your own value.

An example configuration that deploys the upstream `jewel` version of ceph with OSDs that have colocated journals would look like this in `group_vars/all.yml`:

```
ceph_origin: repository
ceph_repository: community
ceph_stable_release: jewel
public_network: "192.168.3.0/24"
cluster_network: "192.168.4.0/24"
monitor_interface: eth1
devices:
  - '/dev/sda'
  - '/dev/sdb'
osd_scenario: colocated
```

The following config options are required to be changed on all installations but there could be other required options depending on your OSD scenario selection or other aspects of your cluster.

- `ceph_origin`
- `ceph_stable_release`
- `public_network`
- `osd_scenario`
- `monitor_interface` or `monitor_address`
- `radosgw_interface` or `radosgw_address`

ceph.conf Configuration

The supported method for defining your `ceph.conf` is to use the `ceph_conf_overrides` variable. This allows you to specify configuration options using an INI format. This variable can be used to override sections already defined in `ceph.conf` (see: `roles/ceph-common/templates/ceph.conf.j2`) or to provide new configuration options. The following sections in `ceph.conf` are supported: `[global]`, `[mon]`, `[osd]`, `[mds]` and `[rgw]`.

An example:

```
ceph_conf_overrides:
  global:
    foo: 1234
    bar: 5678
  osd:
    osd_mkfs_type: ext4
```

Note:

We will no longer accept pull requests that modify the `ceph.conf` template unless it helps the deployment. For simple configuration tweaks please use the `ceph_conf_overrides` variable.

Full documentation for configuring each of the ceph daemon types are in the following sections.

OSD Configuration

OSD configuration is set by selecting an osd scenario and providing the configuration needed for that scenario. Each scenario is different in it's requirements. Selecting your OSD scenario is done by setting the `osd_scenario` configuration option.

- [OSD Scenarios](#)

Contribution

See the following section for guidelines on how to contribute to `ceph-ansible`.

- [Contribution Guidelines](#)

Testing

Documentation for writing functional testing scenarios for `ceph-ansible`.

- [Testing with ceph-ansible](#)
- [Glossary](#)

Demos

Vagrant Demo

Deployment from scratch on bare metal machines: <https://youtu.be/E8-96NamLDo>

Bare metal demo

Deployment from scratch on bare metal machines: https://youtu.be/dv_PEp9qAqg