



SUSE Enterprise Storage Deployment

February 2019

Arthur Yang
Sales Engineer

社区部署方案

部署工具

- Ceph-deploy
- 手工
- k8s+helm
- ceph-ansible



```
$ ceph-deploy --help
```

```
usage: ceph-deploy [-h] [-v] [--cluster NAME] COMMAND ...
```

Deploy Ceph

optional arguments:

-h, --help show this help message and exit

-v, --verbose be more verbose

--cluster NAME name of the cluster

commands:

| COMMAND | description |
|---------|---|
| new | Start deploying a new cluster, and write a CLUSTER.conf for it. |
| install | Install Ceph packages on remote hosts. |
| mon | Deploy ceph monitor on remote hosts. |
| osd | Deploy ceph osd on remote hosts. |
| disk | Prepare a data disk on remote host. |

容器化部署现状

- ceph-ansible has basic container support
 - run daemons via docker...
- (most) people really want a container orchestrator (e.g., kubernetes)
 - stateful services (e.g., OSDs) are super annoying
 - Ceph has lots of stateless services (radosgw, ceph-mds, rbd-mirror, ceph-mgr. Also ganesha, samba, ...)
- real value for small, hyperconverged clusters
- container orchestrators as the new distributed OS

SES部署

DeepSea

- Salt files的集合
 - 部署、管理、运维ceph集群
 - 通过一个salt master管理多个ceph集群
 - 自动发现、部署、配置，管理ceph集群
-
- **Stage 0—the provisioning**— this stage is optional as many sites provides their own provisioning of servers. If you do not have your provisioning tool, you should run this stage. During this stage all required updates are applied and your system may be rebooted.
 - **Stage 1—the discovery**— here you detect all hardware in your cluster and collect necessary information for the Ceph configuration. For details about configuration refer to [Section 4.3, Configuration and Customization](#).
 - **Stage 2—the configuration**— you need to prepare configuration data in a particular format.
 - **Stage 3—the deployment**— creates a basic Ceph cluster with OSD and monitors.
 - **Stage 4—the services**— additional features of Ceph like iSCSI, RadosGW and CephFS can be installed in this stage. Each is optional.
 - **Stage 5—the removal stage**. This stage is not mandatory and during the initial setup it is usually not needed. In this stage the roles of minions and also the cluster configuration are removed. Run this stage, when you need to remove a storage node from your cluster, for details refer to [Section 27.9.3, Removing and Reinstalling Salt Cluster Nodes](#).

DeepSea配置文件

Salt has several standard locations and several naming conventions used on your master node:

`/srv/pillar`

The directory stores configuration data for your cluster minions. *Pillar* is an interface for providing global configuration values to all your cluster minions.

`/srv/salt/`

The directory stores Salt state files (also called *sls* files). State files are formatted descriptions of states in which the cluster should be. For more information, refer to the [Salt documentation](#).

`/srv/module/runners`

The directory stores Python scripts known as runners. Runners are executed on the master node.

`/srv/salt/_modules`

The directory stores Python scripts that are called modules. The modules are applied to all minions in your cluster.

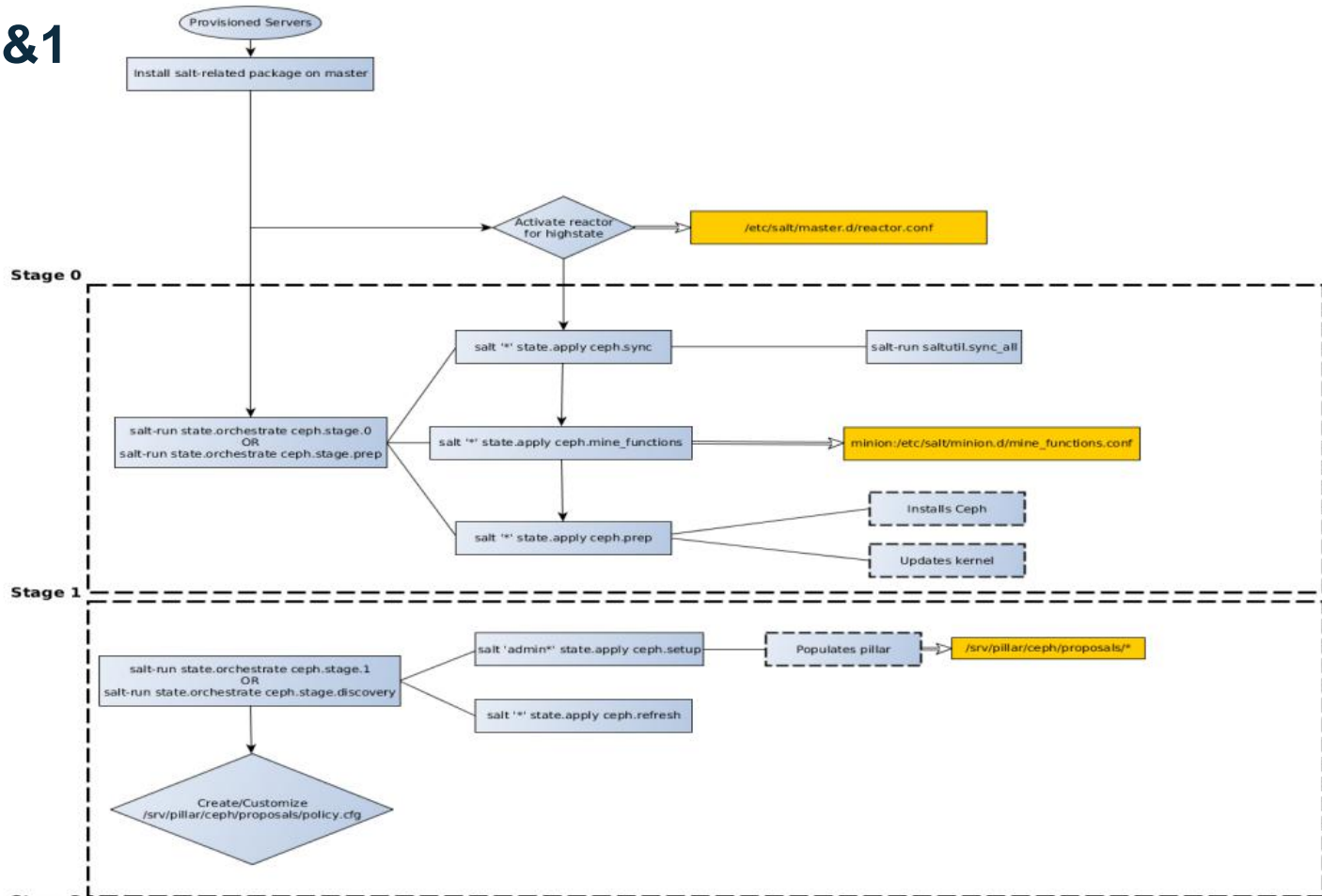
`/srv/pillar/ceph`

The directory is used by DeepSea. Collected configuration data are stored here.

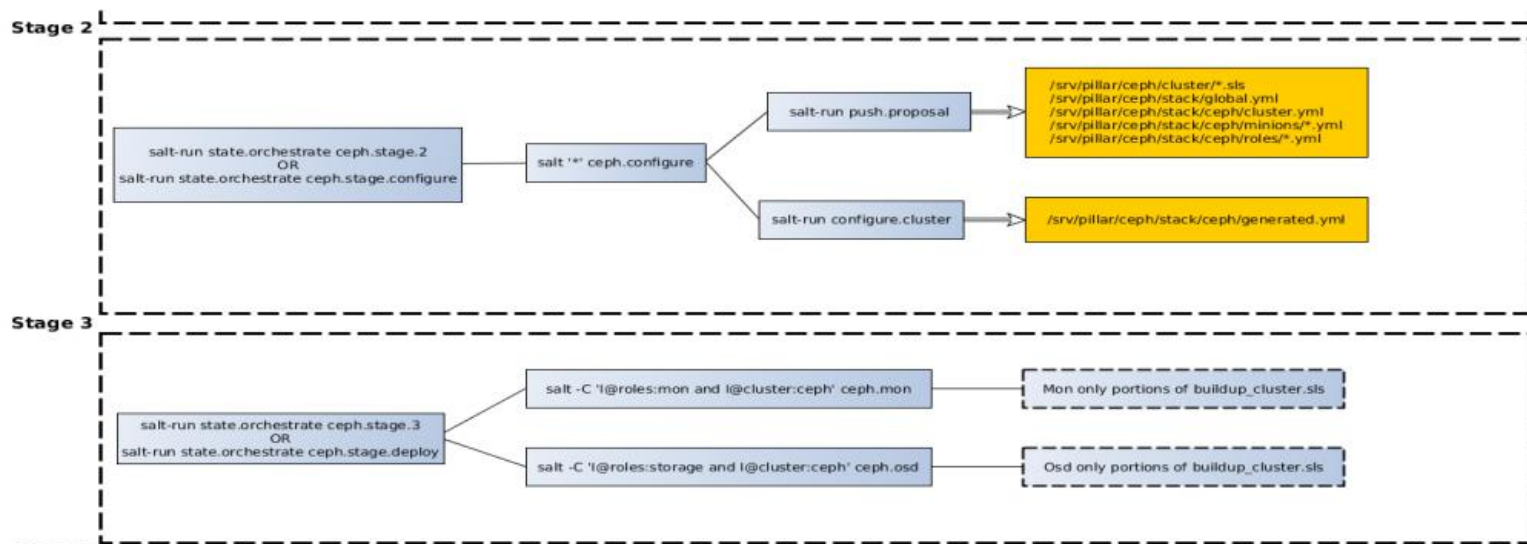
`/srv/salt/ceph`

A directory used by DeepSea. It stores sls files that can be in different formats, but each subdirectory contains sls files. Each subdirectory contains only one type of sls file. For example, `/srv/salt/ceph/stage` contains orchestration files that are executed by `salt-run state.orchestrate`.

Stage.0&1

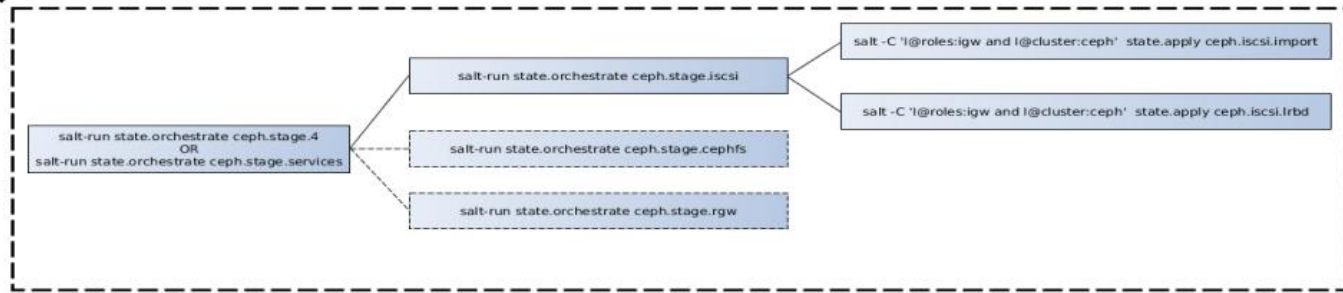


Stage.2&3



Stage.4

Stage 4



SES部署实操

Questions



谢谢!