

salt ceph 部署

runsisi@hust.edu.cn

<http://www.cppblog.com/runsisi>

本文简单描述使用 saltstack 部署 ceph 集群 (MON+OSD) 的方法。

1. saltstack 基本概念

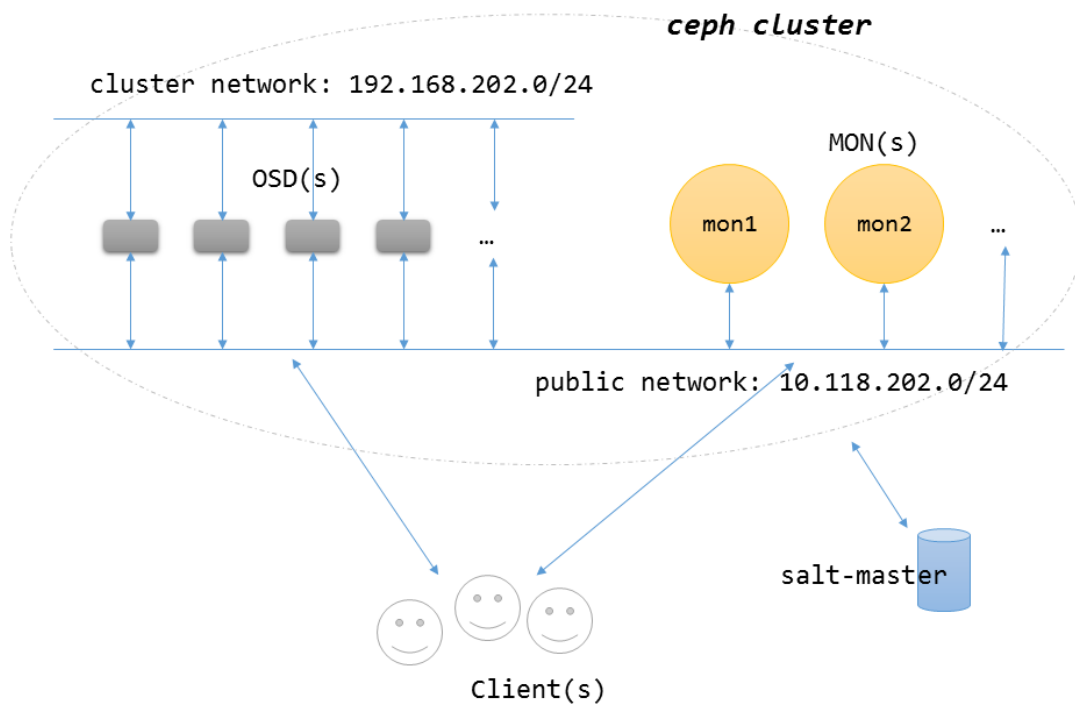
master: saltstack 服务端, 可执行程序为 salt-master;

minion: saltstack 客户端, 可执行程序为 salt-minion;

grains: 客户端节点自身所固有的静态数据, 如节点的基本信息等;

pillar: 在服务端定义的、针对具体客户端的数据。

2. ceph 集群基本假设



- 1) 集群网络拓扑结构如图所示;
- 2) 集群各节点预装 CentOS 7.x x86_64 系统;
- 3) 集群各节点有合法的 FQDN;
- 4) 集群内部无 DNS 解析;
- 5) 集群各节点防火墙关闭;
- 6) 集群各节点可通过 SSH 以 root 用户登录。

3. ceph 集群部署步骤

在正式部署之前还需要做一些准备工作，如准备 master 节点，部署 minion 节点、部署 NTP 等（注意：以下所有操作都在 master 节点进行）。

1) 准备 master 节点

master 节点是一台安装了 salt-master 软件包的 GNU/Linux 机器，对具体发行版本无要求，但要求该节点所安装的 salt-master 软件包的版本必须与 ceph 集群内各节点所安装的 salt-minion 软件包版本一致（本文假设该节点所装系统为 Ubuntu 14.04，saltstack 版本为 2014.7.x），ceph 集群的部署、管理操作都从该节点发起。

添加 saltstack 源：

```
add-apt-repository ppa:saltstack/salt
```

安装 salt-master，salt-ssh 软件包：

```
apt-get install salt-master=2014.7.0+ds-2trusty1
```

```
apt-get install salt-ssh=2014.7.0+ds-2trusty1
```

确保 salt-master 服务已启动：

```
service salt-master restart
```

2) 部署 minion 节点

saltstack 的 SSH 模式可以在不安装 saltstack 客户端的情况下对目标节点进行管理，此处使用 SSH 模式给集群各节点安装 salt-minion 软件包。

下载部署脚本：

```
git clone https://github.com/runsisi/ceph-formula.git
```

创建 saltstack 所需的文件夹：

```
mkdir -p /srv/{salt,pillar}
```

拷贝部署脚本到/srv/salt 目录：

```
cp -r ceph /srv/salt/
```

拷贝 pillar 数据到/srv/pillar 目录：

```
cp -r examples/pillar/* /srv/pillar/
```

修改部署 minion 节点所需的 pillar 数据：

`/srv/pillar/ceph/bootstrap/repo.sls` 定义了安装 `minion`、`NTP` 等软件包所需要的源配置；

`/srv/pillar/ceph/bootstrap/salt.sls` 定义了 `minion` 软件包版本号以及是否在 `minion` 节点的 `hosts` 文件中添加解析 `master` 节点的配置。

SSH 模式需要定义 `roster` 文件用于定义 `minion` 节点：

```
cp examples/etc/salt/roster /etc/salt/roster
```

按照 `/etc/salt/roster` 中的例子修改 `minion` 节点定义，有多少 `minion` 节点就在该文件中定义多少个条目。

确认配置数据无误后开始部署 `minion` 节点：

```
salt-ssh -i '*' state.sls ceph.bootstrap.minion
```

等待部署完成后，查看客户端是否都已经连接到服务端：

```
salt-key -L
```

接收所有客户端的连接：

```
salt-key -A
```

检查客户端与服务端是否连接正常：

```
salt '*' test.ping
```

注意：必须等待 `minion` 节点全部部署完成才能开始下面的部署操作。

3) 部署 NTP

部署 `NTP` 需要考虑不同的情况：一是有外部的 `NTP` 服务器，`ceph` 集群内的节点都作为 `NTP` 客户端；一是没有外部的 `NTP` 服务器，`ceph` 集群内部有一些节点作为 `NTP` 服务器给集群内部其它节点同步。前一种情况需要修改 `/srv/pillar/ceph/bootstrap/ntp.sls` 的 `srvs` 字段用于定义外部 `NTP` 服务器地址，后一种情况需要修改 `peers` 字段用于定义内部 `NTP` 服务器地址。

针对前一种情况部署 `NTP` 客户端：

```
salt '*' state.sls ceph.bootstrap.ntpc
```

针对后一种情况，给内部 `NTP` 服务器部署 `NTP` 服务端（`xxx`，`yyy` 是节点的 `minion id`，可以通过 `salt '*' test.ping` 查看当前所有的 `minion` 节点）：

```
salt 'xxx' state.sls ceph.bootstrap.ntpd
```

```
salt 'yyy' state.sls ceph.bootstrap.ntpd
```

给集群内部其它节点部署 NTP 客户端:

```
salt -C '* and not L@xxx,yyy' ceph.bootstrap.ntpc
```

部署前的准备工作完成之后就可以开始部署 ceph 了，以下操作同样在 master 节点进行:

1) 修改 pillar 数据

/srv/pillar/ceph/repo.sls 定义了 ceph 包的源配置信息;

/srv/pillar/ceph/pkg.sls 定义了 ceph 包的版本号等信息;

/srv/pillar/ceph/conf.sls 定义了 ceph 集群的配置信息;

/srv/pillar/ceph/osd.sls 定义了每个 OSD 节点的设备信息;

2) 部署 MON 节点

```
salt -L 'aaa,bbb,ccc' state.sls ceph.mon
```

3) 部署 OSD 节点

```
salt -C '* and not L@aaa,bbb,ccc' ceph.osd
```

由于我们不支持 MDS 节点的部署，至此所有部署工作全部完成。