

**OCEANBASE**

# OceanBase 云主机环境基础安装与配置 实验指导 V3.0

## 声明

蚂蚁集团版权所有 © 2020 ， 并保留一切权利。

未经蚂蚁集团事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

## 商标声明

**OCEANBASE** 及其他蚂蚁集团相关的商标均为蚂蚁集团所有。本文档涉及的第三方的注册商标，依法由权利人所有。

## 免责声明

由于产品版本升级、调整或其他原因，本文档内容有可能变更。蚂蚁集团保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在蚂蚁集团授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过蚂蚁集团授权渠道下载、获取最新版的用户文档。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

# 目录

<b>1. 实验准备 .....</b>	<b>1</b>
1.1. 打开实验 .....	1
1.2. 实验环境介绍 .....	3
1.2.1. 实验架构 .....	3
1.2.2. 设备需求 .....	4
1.3. 安装介质介绍 .....	5
1.3.1. OCP 安装介质介绍: .....	5
1.3.2. OB 安装介质介绍: .....	6
<b>2. 单点 OCP 的安装 .....</b>	<b>6</b>
2.1. OCP 服务器的基本配置 .....	6
2.1.1. 云主机服务器最小资源配置 .....	6
2.1.2. OCP 服务器的磁盘最小分配 .....	7
2.1.3. 磁盘分配 .....	7
2.2. OCP 服务器的环境配置 .....	11
2.2.1. OCP 安装自动化部署工具 .....	11
2.2.2. OCP 添加 admin 用户 .....	11
2.2.3. 操作系统内核配置 .....	12
2.2.4. 安装依赖包 .....	12
2.2.5. 配置 NTP 时钟同步 .....	12
2.2.6. ocp 服务器安装 docker .....	13
2.2.7. 部署前环境检查 .....	13
2.3. 单点 OCP 的安装 .....	13
2.3.1. 生成 OCP 的配置文件 .....	13
2.3.2. 开始部署 OCP .....	16
2.3.3. 部署后检查 .....	17
<b>3. OceanBase 服务器的环境配置 .....</b>	<b>19</b>
3.1. OB 服务器的基本配置 .....	19
3.1.1. OB 云主机服务器的最小资源配置 .....	19
3.1.2. OB 云主机磁盘分配建议方案 .....	19
3.1.3. 磁盘分配 .....	20
3.2. OB 服务器的环境配置 .....	22
3.2.1. OB 安装自动化部署工具 .....	22
3.2.2. OB 添加 admin 用户 .....	22
3.2.3. OB 操作系统内核配置 .....	22
3.2.4. 安装依赖包 .....	22

3.2.5.	配置 NTP 时钟同步 .....	22
3.2.6.	部署前环境检查 .....	23
<b>4.</b>	<b>在 OCP 上部署 OceanBase 集群 .....</b>	<b>24</b>
4.1.	在 OCP 上部署 OceanBase .....	24
4.1.1.	添加主机 .....	24
4.1.2.	创建 OB 集群 .....	26
4.2.	在 OCP 上部署 OBProxy .....	28
<b>5.</b>	<b>创建租户 .....</b>	<b>29</b>
5.1.	打开一个 OCP 开关（紧限于实验环境） .....	29
5.2.	使用黑屏登录 OB .....	31
5.3.	使用白屏或黑屏创建租户 .....	31
5.3.1.	使用白屏创建租户 .....	31
5.3.2.	使用黑屏创建租户 .....	32

# 1. 实验准备

## 1.1. 打开实验

- 标准的实验环境已经发布在网上，大家可以在云上直接购买并开始实验，例如：

[https://edu.aliyun.com/lab/courses?pageNumber=1&\\_search=oceanbase](https://edu.aliyun.com/lab/courses?pageNumber=1&_search=oceanbase)

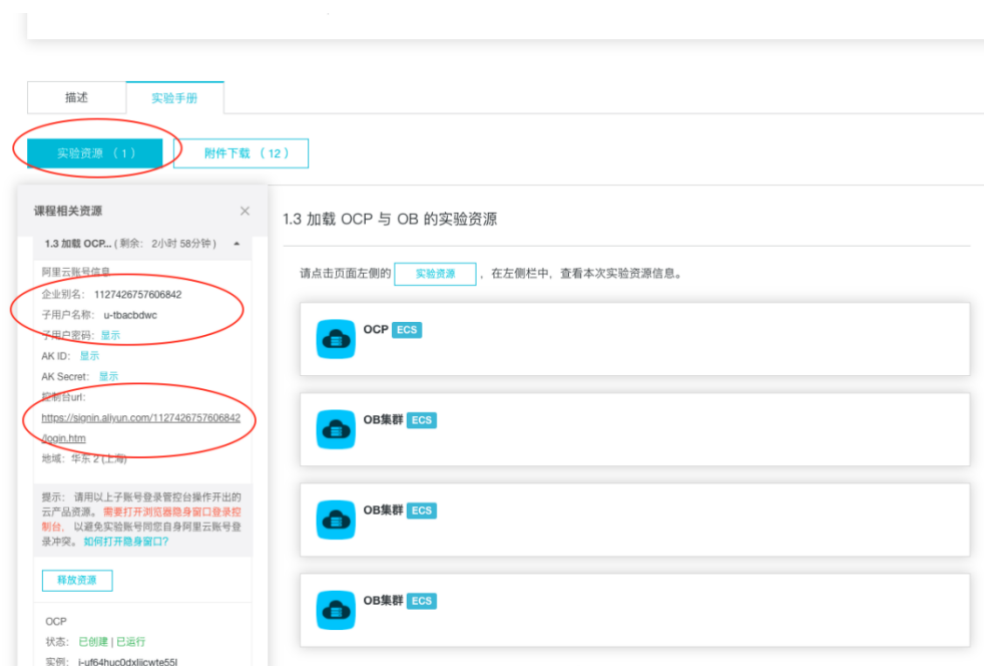


- 左侧是一步步的实验指导，可以在线查看，也可以下载附件在本地查看。 其中的步骤

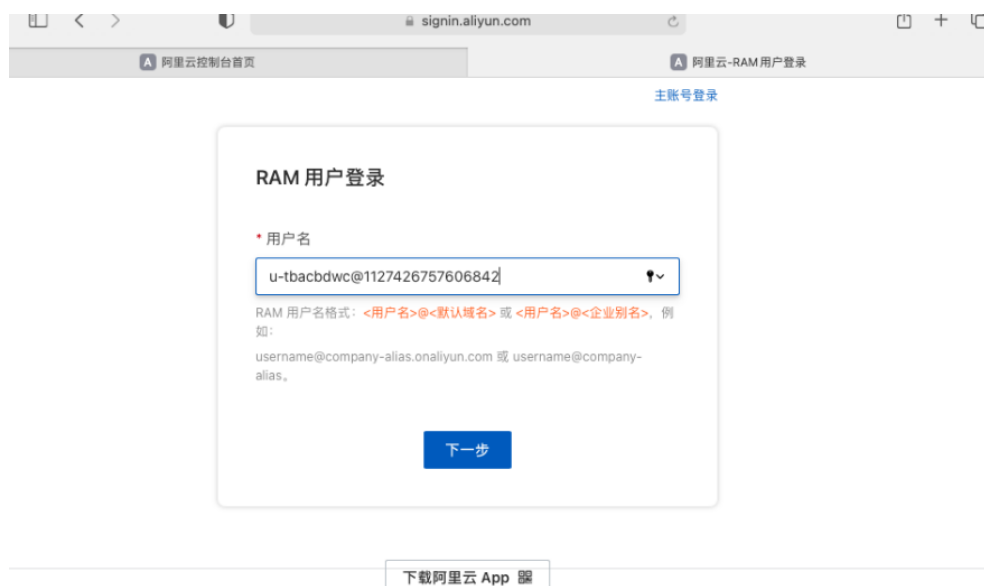
1.3 可以一键启动所有的云主机资源，包括：

- 1 OCP 服务器： 16C，64G，650G 存储
- 3 OB 服务器： 4C，16G，170GB

- 启动之后大家可以在实验资源中看到相关云主机的资源：



- 可以通过 url 打开所有云主机的控制台（建议以另外一个浏览器打开，避免当前账户退出）

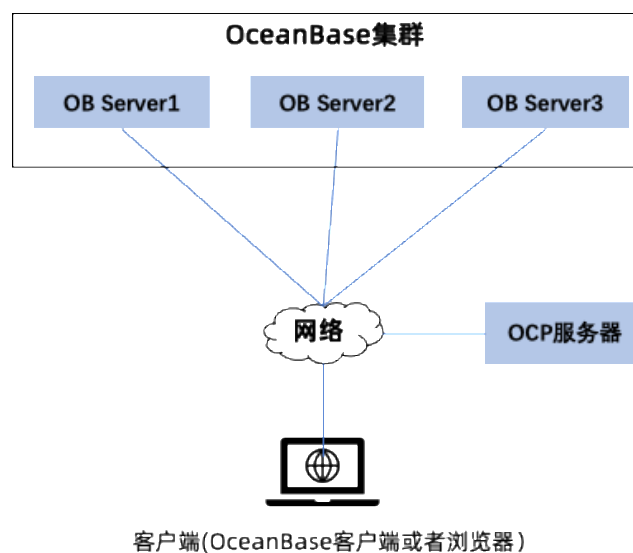


- 黑屏直接登录各个服务器：



## 1.2. 实验环境介绍

### 1.2.1. 实验架构



- 可以采用物理服务器，也可以使用阿里云ECS；
- 操作系统使用主流的Linux操作系统；

## 1.2.2. 设备需求

### ■ 基于物理服务器的安装

对于基于物理服务器的安装，请参照官网文档中心的指导即可。

### ■ 基于云主机的安装

当采用云主机的实验环境模式，如果选取和物理服务器相同的配置当然是可以的。不过我们希望能够尽量节省成本，以下是实验室验证的安装 OB 正式版本的最低配置，可供参考。**该方式只可用于实验练习，不能保证稳定性与性能。**

我们希望基于云主机的安装能够和标准物理服务器的安装保持一致。但因为降低了配置，不可避免的增加了一些修改参数的操作。**请讲师和学员注意，整个文档中标记红色部分的操作只用于学习环境。**

云主机环境因增加了额外操作反而比实际生产环境部署还复杂了一些。当您能够使用云主机环境搭建起来集群，实际生产环境对您来说就更简单了。

服务器类型	数量	功能最低配置
OCP 服务器	1 台	16C, 64G, 650GB 存储
OceanBase 服务器 (OBserver)	3 台	4C, 16G, 170GB 存储

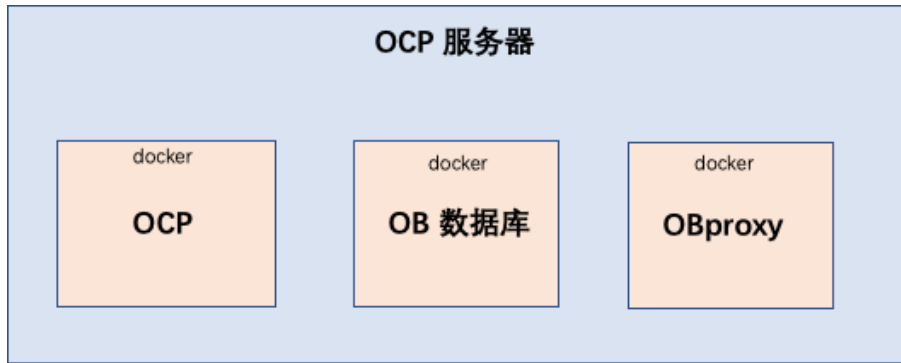
除以上服务器外，每个学员应准备一台笔记本电脑，用以安装 OceanBase 客户端以及浏览器（Chrome、Firefox、Safari 等）。



## 1.3. 安装介质介绍

### 1.3.1. OCP 安装介质介绍：

- 首先介绍一下管理服务器 OCP 的架构，以一个单点的 OCP 服务器为例：



- OCP 服务器其实包含 3 个 docker

一个 docker 安装的是 OCP 的应用，另外两个 docker 安装的是 OCP 管理平台使用的数据库，该数据库是一个单节点的 OB 数据库，以及和 OB 数据库配套的 OBproxy。

（OCP 安装好之后，打开 OCP 的管理页面，大家就会发现有一个默认的 OB 集群存在，这个集群下有 sys 租户，ocp\_meta 租户，ocp\_monitor 租户等，这个集群就是 OCP 使用的 OB 数据库集群和租户）。



- OCP 的安装介质如下

这些介质已经放在了 OCP 服务器的 `/root/t-oceanbase-antman` 目录下。

在这主要简单说明 OCP，OceanBase 服务器（observer）与 OBProxy 的安装包。

	文件	描述
OCP 服务器	t-oceanbase-antman-1.4.1-1936487.aliyos7.x86_64.rpm	自动化部署工具
	ocp330.tar.gz	基于 docker 部署的 OCP 文件
	OB2277_OBP320_x86_20220110.tgz	基于 docker 部署的 OB 文件
	obproxy-3.2.1-20211020153313.e17.x86_64.rpm	Obproxy 安装文件

### 1.3.2. OB 安装介质介绍：

以下是 OB 和 OBProxy 安装文件。OCP、OB 与 OBProxy 的软件版本有一定的匹配关系，以后请参考官网。

	文件	描述
Oceanbase 服务器	oceanbase-3.2.3.0-20220530152606.e17.x86_64	OceanBase 服务器安装文件
OBproxy	obproxy-3.2.1-20211020153313.e17.x86_64.rpm	OBProxy 安装文件

在/root/t-oceanbase-antman 的目录中，还有一些工具软件，如数据库登录的客户端软件 obclient，数据库压测工具 sysbench 等。

## 2. 单点 OCP 的安装

### 2.1. OCP 服务器的基本配置

#### 2.1.1. 云主机服务器最小资源配置

■ OCP 最小资源要求：16C，64G，650G 存储。资源分配详情：

组件		CPU（逻辑 C）	内存（G）
OB docker	sys（500 租户）	/	50（默认） -->需改为 15
	os	2	/
	sys 租户（管理）	5	24

	ocp meta 租户	2	5
	ocp monitor 租户	4	16
	总计	13 (可放大至 16)	60
Obproxy docker		4	2
OCP docker		8	16
总计		25-->16 (共享模式)	78-->64 (共享模式)

在后面正式安装 OCP 的时候，我们会参照以上的方案配置 OCP 的资源分配。可以看到在这种情况下，总的 ocp 资源量是 CPU 25，内存 78G，而我们使用的是 16C64G 的云主机，还是要有一些共享的，这也就要求在实验环境下不能让这三个服务组件承载过大的压力。

### 2.1.2. OCP 服务器的磁盘最小分配

挂载点	大小 (GB)	用途	磁盘格式
/home	100	各组件运行日志盘	建议 ext4
/data/log1	200 (内存大小的 3 倍)	OCP 元数据库 OB 日志盘	建议 ext4
/data/1	100 (存储的数据大小决定)	OCP 元数据库 OB 数据盘	建议 ext4
/docker	200	docker 根目录	建议 ext4
总计	600+其它系统空间 总数 650+G		

### 2.1.3. 磁盘分配

学员从 PC 登录远端 OCP 服务器

```
ssh -l root <ip addr>
```

然后使用 fdisk 查看和创建相关分区，以阿里云创建云主机为例。分别创建 4 个分区

100G	给/home 使用
200G	给/data/log1 使用

100G	给/data/1 使用
200G	给/docker 使用

### 磁盘分区、格式化、挂载

```
fdisk /dev/vdb

命令(输入 m 获取帮助): n
Partition type:
  p   primary (0 primary, 0 extended, 4 free)
  e   extended
Select (default p): p
分区号 (1-4, 默认 1):
起始 扇区 (2048-1258291199, 默认为 2048):
将使用默认值 2048
Last 扇区, +扇区 or +size{K,M,G} (2048-1258291199, 默认为 1258291199):
+100G
分区 1 已设置为 Linux 类型, 大小设为 100 GiB

.....#依次类推, 不重复展示

命令(输入 m 获取帮助): w。#w 保存设置
The partition table has been altered!

Calling ioctl() to re-read partition table.
```

```
[root@xinran-OCP ~]# fdisk -l

磁盘 /dev/vda: 42.9 GB, 42949672960 字节, 83886080 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x000d2717

   设备 Boot      Start         End      Blocks   Id  System
/dev/vda1  *          2048       83884031    41940992   83  Linux

磁盘 /dev/vdb: 1919.9 GB, 1919850381312 字节, 3749707776 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节

[root@xinran-OCP ~]# fdisk /dev/vdb
欢迎使用 fdisk (util-linux 2.23.2)。

更改将停留在内存中, 直到您决定将更改写入磁盘。
使用写入命令前请三思。

Device does not contain a recognized partition table
使用磁盘标识符 0x2bfa4999 创建新的 DOS 磁盘标签。

命令(输入 m 获取帮助): n
Partition type:
  p   primary (0 primary, 0 extended, 4 free)
  e   extended
Select (default p): p
分区号 (1-4, 默认 1):
起始 扇区 (2048-3749707775, 默认为 2048):
将使用默认值 2048
Last 扇区, +扇区 or +size{K,M,G} (2048-3749707775, 默认为 3749707775): +100G
```

创建完成后查看分区情况，并使用 ext4 文件格式格式化

```
mkfs.ext4 /dev/vdb1
mkfs.ext4 /dev/vdb2
mkfs.ext4 /dev/vdb3
mkfs.ext4 /dev/vdb4

fdisk -l      #查看磁盘情况
```

```
[root@xinran-OCP ~]# fdisk -l

磁盘 /dev/vda: 42.9 GB, 42949672960 字节, 83886080 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x000d2717

   设备 Boot      Start         End      Blocks   Id  System
/dev/vda1  *          2048       83884031   41940992   83   Linux

磁盘 /dev/vdb: 1919.9 GB, 1919850381312 字节, 3749707776 个扇区
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理): 512 字节 / 512 字节
I/O 大小(最小/最佳): 512 字节 / 512 字节
磁盘标签类型: dos
磁盘标识符: 0x2bfa4999

   设备 Boot      Start         End      Blocks   Id  System
/dev/vdb1             2048    209717247   104857600   83   Linux
/dev/vdb2          209717248    629147647   209715200   83   Linux
/dev/vdb3          629147648    838862847   104857600   83   Linux
/dev/vdb4          838862848   1258293247   209715200   83   Linux
[root@xinran-OCP ~]# mkfs.ext4 /dev/vdb1
```

创建相关的目录

```
mkdir /data
mkdir /data/log1
mkdir /data/1
mkdir /docker
```

执行 mount 挂载，修改/etc/fstab 文件

```
vim /etc/fstab
```

```
# /etc/fstab
# Created by anaconda on Thu Aug 17 07:38:21 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=976105f5-f402-456c-aadd-50de49ff88f9 / ext4 defaults 1 1
/dev/vdb1 /home ext4 defaults 1 1
/dev/vdb2 /data/log1 ext4 defaults 1 1
/dev/vdb3 /data/1 ext4 defaults 1 1
/dev/vdb4 /docker ext4 defaults 1 1
```

执行 mount 操作，最后查看挂载点是否成功

```
mount -a
df -h
```

```
[root@xinran-OCP ~]# mount -a
[root@xinran-OCP ~]#
[root@xinran-OCP ~]#
[root@xinran-OCP ~]# df -h
```

文件系统	容量	已用	可用	已用%	挂载点
/dev/vda1	40G	1.7G	36G	5%	/
devtmpfs	32G	0	32G	0%	/dev
tmpfs	32G	0	32G	0%	/dev/shm
tmpfs	32G	316K	32G	1%	/run
tmpfs	32G	0	32G	0%	/sys/fs/cgroup
tmpfs	6.3G	0	6.3G	0%	/run/user/0
/dev/vdb1	99G	61M	94G	1%	/home
/dev/vdb2	197G	61M	187G	1%	/data/log1
/dev/vdb3	99G	61M	94G	1%	/data/1
/dev/vdb4	197G	61M	187G	1%	/docker

### 2.1.3.1. 磁盘分配：lvm 方式

#### OCP 云主机

```
yum -y install lvm2
pvcreate /dev/vdb && vgcreate obcp /dev/vdb

lvcreate -L 100G -a y -n home obcp
lvcreate -L 200G -a y -n datalog obcp
lvcreate -L 100G -a y -n data obcp
lvcreate -L 200G -a y -n docker obcp

mkfs.ext4 /dev/obcp/home
mkfs.ext4 /dev/obcp/datalog
mkfs.ext4 /dev/obcp/data
mkfs.ext4 /dev/obcp/docker
```

```

mkdir -p /data/log1
mkdir -p /data/1
mkdir /docker

echo "/dev/obcp/home /home ext4 defaults 0 0" >>/etc/fstab
echo "/dev/obcp/data1og /data/log1 ext4 defaults 0 0" >>/etc/fstab
echo "/dev/obcp/data /data/1 ext4 defaults 0 0" >>/etc/fstab
echo "/dev/obcp/docker /docker ext4 defaults 0 0" >>/etc/fstab

mount -a
df -Th

```

## 2.2. OCP 服务器的环境配置

进入到/root/t-oceanbase-antman 目录, 相应的安装包已经上传到该目录。

### 2.2.1. OCP 安装自动化部署工具

OCP 与 OceanBase 的服务器上都安装, 例如:

```
rpm -ivh t-oceanbase-antman-1.4.1-1936487.aliios7.x86_64.rpm
```

### 2.2.2. OCP 添加 admin 用户

每台服务器都添加 admin 用户

```

cd /root/t-oceanbase-antman/clonescripts
./clone.sh -u
passwd admin
#请立即修改成复杂密码

```

[[root@obce-ocp-test clonescripts]#  
[[root@obce-ocp-test clonescripts]# passwd admin  
更改用户 admin 的密码。  
新的 密码:  
重新输入新的 密码:  
passwd: 所有的身份验证令牌已经成功更新。

### 2.2.3. 操作系统内核配置

OCP 与 OceanBase 的所有服务器上都进行配置

```
cd /root/t-oceanbase-antman/clonescripts
./clone.sh -r ocp -c
```

### 2.2.4. 安装依赖包

```
./clone.sh -r ocp -m
```

### 2.2.5. 配置 NTP 时钟同步

配置 NTP 时钟同步，保证所有节点的时钟偏差在 100ms 以内。

对于云主机，一般已经配置好了 NTP，但时差可能会过大，而 OB 集群对时钟差要求严格（100ms 以内），所以建议还是在几台服务器之间配置自己的 NTP 同步，避免出现因时间的各种问题（例如 OB 不能启动）。

我们选择 OCP 服务器作为 NTP server（例如 OCP 172.23.204.86），确认本地的时间设置（date 命令）。然后修改/etc/ntp.conf 文件。建议将原 ntp.conf 改为 ntp.conf.bak（命令：mv ntp.conf ntp.conf.bak），然后用 vi 创建新的 ntp.conf：

```
cd /etc
mv ntp.conf ntp.conf.bak #备份原配置文件
vi ntp.conf              #新建一个 ntp 配置文件
```

将自己设置为 ntp server，并允许 172.23.0.0 网段的 ntp client 访问（这里的 NTP client 例如是 OB）：

修改之后记得重启 ntp 服务：service ntpd restart

然后用 ntpstat ntpq -np 等命令查看同步状态。

```
restrict 172.28.0.10 mask 255.255.0.0 nomodify
server 127.127.1.0
fudge 127.127.1.0 stratum 10
systemctl restart ntpd
systemctl status ntpd
systemctl enable ntpd
ntpstat
```



```
ntpq -np
```

## 2.2.6. ocp 服务器安装 docker

```
cd /root/t-oceanbase-antman/clonescripts/  
./clone.sh -i
```

## 2.2.7. 部署前环境检查

```
sh precheck.sh -m ocp
```

```
### SUMMARY OF ISSUES IN PRE-CHECK ###  
check SELinux status: Permissive ... EXPECT Disabled ... FAIL  
TIPS: disable SELinux  
sed -i 's/enforcing/disabled/g' /etc/selinux/config  
/usr/sbin/setenforce 0
```

在检查的最后，会将所有存在的问题一并列出，并且给出解决建议。可以参照建议解决问题，并再次运行环境检查。

**⚠ 注意：**云主机因为使用最小资源配置，自检的时候可能会有资源方面的告警(mem, disk)，可以忽略。

## 2.3. 单点 OCP 的安装

**⚠ 注意：**OCP 有单节点和三节点两种部署模式，在实验环境中，我们使用单节点模式。

### 2.3.1. 生成 OCP 的配置文件

```
cd /root/t-oceanbase-antman  
bash init_obcluster_conf.sh
```

- 填写正确 OCP 服务器的 ip、root 和 admin 用户的密码。单节点的负载均衡模式是 none。（云主机可以填写私网 ip、OCP 和 Observer 之间全部使用私网 IP 通信）

```
## obcluster.conf
##
SINGLE_OCP_MODE=TRUE
##### 根据环境必须修改 / MUST CHANGE ACCORDING ENVIRONMENT #####
##### 填写机器IP和 metaob 容器内的root/admin密码 / Edit Machine IP and Password Of root/admin #####
ZONE1_RS_IP=172.23.204.86
OBSERVER01_ROOTPASS='^~!@/Basee' # metaob容器使用, 非宿主账号, 宿主SSH信息请使用指定环境变量的方式, 详见 install.sh -h
OBSERVER01_ADMINPASS='^~!@/Basee'

##### 填写负载均衡配置 / Edit Configuration of Load Balance #####
# LB_MODE: dns/f5/none, default: dns for 3 ocp, none for 1 ocp
LB_MODE=none
```

- 下面配置 OCP 服务器上各个 Docker 的资源（OCP Docker，OB Docker，OBProxy Docker），默认可能是以下配置：

```
##### 根据服务器CPU、内存设置容器资源编排 / Allocate Container Resources According To Server #####
OB_DOCKER_CPUS=32
OB_DOCKER_MEMORY=128G
OB_SYSTEM_MEMORY=50G # Do not modify OS_SYSTEM_MEMORY unless you know what you are doing!
OCP_DOCKER_CPUS=8
OCP_DOCKER_MEMORY=16G
OBPROXY_DOCKER_CPUS=4
OBPROXY_DOCKER_MEMORY=12G
```

因为我们的云主机总容量是 16C64G，650GB 磁盘，所以采用以下配置：

```
OB_DOCKER_CPUS=16
OB_DOCKER_MEMORY=60G
OB_SYSTEM_MEMORY=50G # Do not modify OS_SYSTEM_MEMORY unless you know
what you are doing!
OCP_DOCKER_CPUS=8
OCP_DOCKER_MEMORY=16G
OBPROXY_DOCKER_CPUS=4
OBPROXY_DOCKER_MEMORY=2G
```

详情见前面章节的描述。

- 另外就是修改安装软件的版本信息，根据你获得的实际软件的情况修改：（以下样例针对的是 前满章节中列出的安装文件对应的信息）

```
##### 填写OCP各组件容器的版本信息 / Edit Docker Image, Repo And Tag of OCP Components #####
# OB docker
OB_DOCKER_IMAGE_PACKAGE=OB2277_OBP320_x86_20220110.tgz
OB_IMAGE_REPO=reg.docker.alibaba-inc.com/antman/ob-docker
OB_IMAGE_TAG=OB2277_OBP320_x86_20220110
# OCP docker
OCP_DOCKER_IMAGE_PACKAGE=ocp322.tar.gz
OCP_IMAGE_REPO=reg.docker.alibaba-inc.com/oceanbase/ocp-all-in-one
OCP_IMAGE_TAG=3.2.2-1936175
# OBPROXY docker, 如果使用了内置proxy的OB镜像, OBPROXY镜像会被忽略
OBPROXY_DOCKER_IMAGE_PACKAGE=obproxy183_x86.tar.gz
OBPROXY_IMAGE_REPO=reg.docker.alibaba-inc.com/antman/obproxy
OBPROXY_IMAGE_TAG=OBP183_20201223
OBPROXY_USE_RS=FALSE # 使用独立proxy镜像时是否使用rs而不是configurl启动
```

image\_package 填写你获得安装软件的名称，例如：

```
# OB docker
OB_DOCKER_IMAGE_PACKAGE=OB2277_OBP320_x86_20220110.tgz
OB_IMAGE_REPO=reg.docker.alibaba-inc.com/antman/ob-docker
OB_IMAGE_TAG=OB2277_OBP320_x86_20220110
# OCP docker
OCP_DOCKER_IMAGE_PACKAGE=ocp330.tar.gz
```

```
OCP_IMAGE_REPO=reg.docker.alibaba-inc.com/oceanbase/ocp-all-in-one
OCP_IMAGE_TAG=3.3.0-20220427184540
```

查看容器版本信息方法

新开 session ssh 访问 OCP 服务器

```
[root@OCP ~]# cd /root/t-oceanbase-antman/
[root@OCP t-oceanbase-antman]#
for img in `ls *gz`;do echo $img ;docker load < $img; done
[root@OCP t-oceanbase-antman]# docker images
[root@OCP t-oceanbase-antman]# ll *gz
```

**△ 注意：**OBproxy 安装会在 ocp 中执行，暂时不做设置

完成配置后进行保存。OCP 的配置文件在 /root/t-oceanbase-antman 目录下，3.X 版本的名称是 obcluster.conf，也可以手工进行修改。

#### ■ 云主机额外配置

另外我们的云主机是用的最小资源（16C64G，650GB），所以我们需要额外的多做一些配置。

**△ 注意：**生产环境标准配置不需要做以下这些

我们需要将 sys(500 租户)默认的 50G 内存，改为 15G，修改 install\_OB\_docker.sh

```
vi install_OB_docker.sh
#小 tips: 输入/直接搜索 memory_limit_percentage, 直接定位需要修改位置
#小 tips: :set nu 显示行号
```



```
105 -e OPTSTR="\cpu_count=$OB_DOCKER_CPU,system_memory=$OB_SYSTEM_MEMORY,memory_limit=$observer_memory_limit,___min_full_resource_pool_memory=1073741824,ob_enable_prepared_statement=false,memory_limit_percentage=90,system_memory=15G" \
106 --cpu-period 100000 \
107 --cpu-quota $OB_DOCKER_CPU$'000000' \
108 --cpuset-cpus "0" --cgroup $OB_DOCKER_CPU - 1" \ install_OB_docker.sh
109 --memory $OB_DOCKER_MEMORY \
110 -v /home/admin/oceanbase:/home/admin/oceanbase \
111 -v $PHYSICAL_LOG_DIR:/data/log1 \
112 -v $PHYSICAL_DATA_DIR:/data/1 \
113 -v $PHYSICAL_BACKUP_DIR:/obbackup \ 修改install.sh
114 --restart on-failure:5 \
115 $docker_image_name"
116
117 docker_run_cmd="docker run -d -it --cap-add SYS_RESOURCE --name $docker_name --net=host \
118 -e OB_CLUSTER_NAME=$obcluster_name \
119 -e DEV_NAME=$dev_name \
120 -e ROOTSERVICE_LIST="$rootservice_list" \
121 -e DATAFILE_DISK_PERCENTAGE=$DATAFILE_DISK_PERCENTAGE \
122 -e CLUSTER_ID=$cluster_id \
123 -e ZONE_NAME=$zone \
124 -e OBPROXY_PORT=$OBPROXY_PORT \
125 -e MYSQL_PORT=$MYSQL_PORT \
126 -e RPC_PORT=$RPC_PORT \
127 -e OCP_VIP=$OCP_VIP \
128 -e OCP_VPORT=$OCP_VPORT \
129 -e app.password_root="$password_root" \
130 -e app.password_admin="$password_admin" \
131 -e OBPROXY_OPTSTR="$proxy_optstr" \ cd /root/t-oceanbase-antman
132 -e OPTSTR="\cpu_count=$OB_DOCKER_CPU,system_memory=$OB_SYSTEM_MEMORY,memory_limit=$observer_memory_limit,___min_full_resource_pool_memory=1073741824,ob_enable_prepared_statement=false,memory_limit_percentage=90,system_memory=15G" \
```

105 行和 132 行（请注意要修改的是两行）：

```
105 -e
OPTSTR=\"cpu_count=$OB_DOCKER_CPUS,system_memory=$OB_SYSTEM_MEMORY,memory
_limit=$observer_memory_limit,__min_full_resource_pool_memory=1073741824,
_ob_enable_prepared_statement=false,memory_limit_percentage=90\" \
```

修改成

```
105 -e
OPTSTR=\"cpu_count=$OB_DOCKER_CPUS,system_memory=$OB_SYSTEM_MEMORY,memory
_limit=$observer_memory_limit,__min_full_resource_pool_memory=1073741824,
_ob_enable_prepared_statement=false,memory_limit_percentage=90,system_memory=15G\" \
```

另外新版本的 install.sh 文件会检查 system\_memory，我们需要注释掉检查的这一行，修改 install.sh，注释掉 1373 行

```
1373 function step3()
1374 {
1375     # system_memory is 50G default
1376     #[ -z "$OB_SYSTEM_MEMORY" ] && OB_SYSTEM_MEMORY=50G
1377     #avail_tenant_memory_num=$((($OB_DOCKER_MEMORY - $OB_SYSTEM_MEMORY))G)
1378     #[ $avail_tenant_memory_num -lt 37 ] && { antman_log "docker ob avail tenant memory is ($OB_DOCKER_MEMORY - $OB_SYSTEM_MEMORY)G, docker memory limit($OB_DOCKER_MEMORY) too small."
1379     "ERROR"; exit 1; }
```

```
function step3()
{
    # system_memory is 50G default
    #[ -z "$OB_SYSTEM_MEMORY" ] && OB_SYSTEM_MEMORY=50G
    #avail_tenant_memory_num=$((($OB_DOCKER_MEMORY -
    ${OB_SYSTEM_MEMORY}G))
    #[ $avail_tenant_memory_num -lt 37 ] && { antman_log "docker ob
    avail tenant memory is ($OB_DOCKER_MEMORY - ${OB_SYSTEM_MEMORY}G)G,
    docker memory limit($OB_DOCKER_MEMORY) too small." "ERROR"; exit
    1; }\
```

### 2.3.2. 开始部署 OCP

设置 SSH 参数（3.x 版本新增）

```
export SSH_AUTH=password
export SSH_USER=root
export SSH_PORT=22
export SSH_PASSWORD='真实密码' #此处为之前设置的真实密码
export SSH_KEY_FILE=/root/.ssh/id_rsa
```

可以先查看一下 OCP 部署的命令帮助

```
cd /root/t-oceanbase-antman
./install.sh -h
```

可以看到 OCP 的安装需要 8 steps。下面执行 OCP 安装

```
./install.sh -i 1-8
```

```
[2020-12-17 11:27:56.349246] INFO [172.18.1.1: post_check_ob done]
[2020-12-17 11:27:56.352504] INFO [172.18.1.1: post_check_ocp_service start]
[2020-12-17 11:27:56.354859] INFO [Curl 172.18.1.1 -> http://172.18.1.1:8080/se
rvices?Action=GetObProxyConfig&User_ID=admin&UID=alibaba]
[2020-12-17 11:27:56.693398] INFO [172.18.1.1: post_check_ocp_service done]
[2020-12-17 11:27:56.703217] INFO [step8: post check done]
```

看到以上信息，表示 OCP 部署已完成。如果中间失败，可以根据提示解决问题，然后再次从失败的 step 再次执行。

如果要回退或卸载 OCP，使用

```
./install.sh -c 1-x
```

### Step 8 报错处理：

```
ssh-keygen -R 172.28.0.43                (OCP ip)
./install.sh -c 8
./install.sh -I 8
```

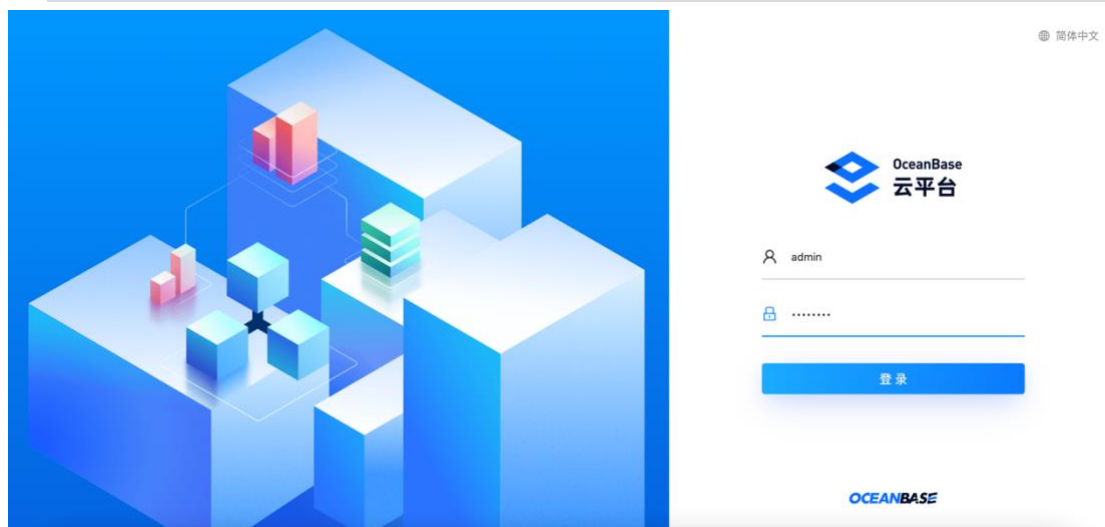
```
[2022-09-11 19:31:25.758789] INFO [OCP multi AZ disabled, pass.]
[2022-09-11 19:31:25.762948] INFO [step7: installation of ocp done]
[2022-09-11 19:31:25.767209] INFO [step8: post-checking service, logfile: /root/t-oceanbase-antman/logs/post_check_service.log]
[2022-09-11 19:31:25.805220] INFO [172.28.0.43: post_check_connection start]
[2022-09-11 19:31:25.808682] INFO [Ping 172.28.0.43 -> 172.28.0.43]
[2022-09-11 19:31:27.975521] INFO [172.28.0.43: post_check_connection done]
[2022-09-11 19:31:27.979882] INFO [172.28.0.43: post_check_port start]
[2022-09-11 19:31:27.982583] INFO [Connect 172.28.0.43 -> 172.28.0.43:8080]
[2022-09-11 19:31:30.521613] ERROR [ANTMAN-210: Connect 172.28.0.43 -> 172.28.0.43:8080 failed!]
[2022-09-11 19:31:30.526718] ERROR [ANTMAN-314: ERROR occurred in post_check_service, install.sh exit]
root@ocp t-oceanbase-antman]#
root@ocp t-oceanbase-antman]# ssh-keygen -R 172.28.0.43
# Host 172.28.0.43 found: line 1
/root/.ssh/known_hosts updated.
Original contents retained as /root/.ssh/known_hosts.old
root@ocp t-oceanbase-antman]# ./install.sh -c 8
run install.sh with DEBUG=FALSE, INSTALL_STEPS= CLEAR_STEPS=8 CONFIG_FILE=/root/t-oceanbase-antman/obcluster.conf
[2022-09-11 19:55:47.224634] INFO [check conf file /root/t-oceanbase-antman/obcluster.conf format ...]
[2022-09-11 19:55:47.229572] INFO [conf file is upper case format.]
[2022-09-11 19:55:49.282974] INFO [start antman API service]
[2022-09-11 19:55:49.337316] INFO [SSH_AUTH=password SSH_USER=root SSH_PORT=22 SSH_PASSWORD=AN*** SSH_KEY_FILE=/root/.ssh/id_rsa]
LB_MODE=none
[2022-09-11 19:55:49.650397] INFO [clear_step8: no need to clear for step network post-check]
root@ocp t-oceanbase-antman]#
root@ocp t-oceanbase-antman]# ./install.sh -I 8
run install.sh with DEBUG=FALSE, INSTALL_STEPS=8 CLEAR_STEPS= CONFIG_FILE=/root/t-oceanbase-antman/obcluster.conf
[2022-09-11 19:55:53.605856] INFO [check conf file /root/t-oceanbase-antman/obcluster.conf format ...]
[2022-09-11 19:55:53.610906] INFO [conf file is upper case format.]
[2022-09-11 19:55:55.664681] INFO [start antman API service]
[2022-09-11 19:55:55.717887] INFO [SSH_AUTH=password SSH_USER=root SSH_PORT=22 SSH_PASSWORD=AN*** SSH_KEY_FILE=/root/.ssh/id_rsa]
LB_MODE=none
[2022-09-11 19:55:56.044036] INFO [step8: post-checking service, logfile: /root/t-oceanbase-antman/logs/post_check_service.log]
[2022-09-11 19:55:56.080715] INFO [172.28.0.43: post_check_connection start]
[2022-09-11 19:55:56.083992] INFO [Ping 172.28.0.43 -> 172.28.0.43]
```

### 2.3.3. 部署后检查

OCF 访问地址: `http://<ocpip>:8080` # 云主机注意安全策略放开 8080 端口

默认用户名: admin

默认密码: aaAA11\_\_ # ocp 3.x 版本默认密码, 注意最后为两个下划线



打开 ocp 的 web 管理页面后, 就有一个集群和 3 个租户存在。



## 3. OceanBase 服务器的环境配置

### 3.1. OB 服务器的基本配置

#### 3.1.1. OB 云主机服务器的最小资源配置

OB 云主机服务器的最小资源配置：4C，16G，170GB 存储

组件		CPU（逻辑 C）	内存（G）
OB server	sys（500 租户）	/	5 (system_memory)
	os	2	/
	sys 租户（管理）	5	2.5（默认） 可在启动后改为 2G
	业务租户	5	2 (每租户)
OB server 总计		32（逻辑 cpu） (cpu_count)	15 (memory_limit)
OB proxy (在实验环境与 OB 部署在一起)		/	1 (proxy_mem_limit)
云主机		4C (物理)	16G (物理)

#### 3.1.2. OB 云主机磁盘分配建议方案

挂载点	大小（GB）	用途	磁盘格式
/home	5	各组件运行日志盘	建议 ext4
/data/log1	20 (看保留日志要求)	OCP 元数据库 OB 日志盘	建议 ext4
/data/1	110 (存储的数据大小决定) 现阶段在安装过程中有一个检查 要求/data/1 必须大于 100G，并 且必须是一个独立的挂载点	OCP 元数据库 OB 数据盘	建议 ext4
总计	135+其它系统空间 总数 170		

**⚠ 注意：**创建云主机的时候一般要选择安全策略，请打开 8080，2883 端口等

### 3.1.3. 磁盘分配

学员登录 OB 服务器。（OB 服务器可以没有公网 IP，学员从 PC 登录远端 OCP 服务器，然后再从 OCP 服务器登录 OB 服务器。）

```
ssh -l root <ip addr>
```

然后使用 fdisk 查看和创建相关的分区，以此次阿里云创建的云主机为例。为 OB 创建 2 个分区：

110G	给/data/1 使用
20G	给/data/log1 使用

在测试环境/home 需要空间很小，默认使用系统盘的根分区即可。

```
[root@xinran-ob001 ~]# fdisk /dev/vdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0xddb9d58.

Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p):
Using default response p
Partition number (1-4, default 1):
First sector (2048-272629759, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-272629759, default 272629759): +110G
Partition 1 of type Linux and of size 110 GiB is set

Command (m for help): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p):
Using default response p
Partition number (2-4, default 2):
First sector (230688768-272629759, default 230688768):
Using default value 230688768
Last sector, +sectors or +size{K,M,G} (230688768-272629759, default 272629759):
Using default value 272629759
Partition 2 of type Linux and of size 20 GiB is set

Command (m for help): w
The partition table has been altered!
```

同样格式化分区（mkfs.ext4），创建相关的目录（mkdir），并进行 mount 操作。

具体操作请参考 2.1.3 章节



磁盘分配操作最后检查结果：

```
[root@xinran-ob001 ~]# mount -a
[root@xinran-ob001 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1        40G   1.7G   36G   5% /
devtmpfs         7.7G     0   7.7G   0% /dev
tmpfs            7.7G     0   7.7G   0% /dev/shm
tmpfs            7.7G  332K   7.7G   1% /run
tmpfs            7.7G     0   7.7G   0% /sys/fs/cgroup
tmpfs            1.6G     0   1.6G   0% /run/user/0
/dev/vdb1        109G   61M  103G   1% /data/1
/dev/vdb2        20G   45M   19G   1% /data/log1
```

⚠ 注意：这里/data/1一定要大于 100G，因为后续在 OCP 上创建集群的过程中，有一个检查的脚本将 100G 的参数写死了，/data/1 低于 100G 将安装失败。（未来可能会修改）

### 3.1.3.1. 磁盘分配：lvm 方式

observer 主机

```
yum -y install lvm2
pvcreate /dev/vdb && vgcreate obcp /dev/vdb

lvcreate -L 20G -a y -n datalog obcp
mkfs.ext4 /dev/obcp/datalog
mkdir -p /data/log1
echo "/dev/obcp/datalog /data/log1 ext4 defaults 0 0" >>/etc/fstab

lvcreate -L 110G -a y -n data obcp
mkfs.ext4 /dev/obcp/data
mkdir -p /data/1
echo "/dev/obcp/data /data/1 ext4 defaults 0 0" >>/etc/fstab

mount -a
df -Th
```

## 3.2. OB 服务器的环境配置

进入到 `/root/t-oceanbase-antman` 目录, 相应的安装包已经上传到该目录。

### 3.2.1. OB 安装自动化部署工具

OB 的服务器上都安装自动化部署工具, 例如:

```
rpm -ivh t-oceanbase-antman-1.4.1-1936487.aliios7.x86_64.rpm
```

### 3.2.2. OB 添加 admin 用户

为每台服务器都添加 admin 用户 :

```
cd /root/t-oceanbase-antman/clonescripts
./clone.sh -u
passwd admin
#请立即修改成复杂密码
```

### 3.2.3. OB 操作系统内核配置

对每台服务器的操作系统内核进行设置 :

```
./clone.sh -r ob -c
```

### 3.2.4. 安装依赖包

```
./clone.sh -r ob -m
```

### 3.2.5. 配置 NTP 时钟同步

配置 NTP 时钟同步, 保证所有节点的时钟偏差在 100ms 以内

对于云主机, 一般已经配置好了 NTP, 但时差可能会过大, 而 OB 集群对时钟差要求严格(100ms 以内), 所以建议还是在几台服务器之间配置自己的 NTP 同步, 避免出现因时间的各种问题(例如 OB 不能启动)。

在前面的步骤中，我们已经将 OCP 服务器设置成为 NTP Server。现在需要将其它 OB 服务器作为 NTP client，同样建议将原 ntp.conf 改为 ntp.conf.bak （命令：mv ntp.conf ntp.conf.bak），然后用 vi 创建新的 ntp.conf：

```
cd /etc
mv ntp.conf ntp.conf.bak
vim ntp.conf    #修改信息如下图所示
server 172.28.0.10
fudge 172.28.0.10 stratum 10
```

```
server 172.23.204.86
fudge 172.23.204.86 stratum 10
```

修改之后记得重启 ntp 服务：service ntpd restart

然后用 ntpstat, ntpq -np 等命令查看同步状态。

```
service ntpd restart
ntpstat ntpq -np    #查看同步状态，重启后需要等待一段时间，显示 synchronised

ntpq -plgrep -E "\*|\=|remote"    #查看 offset 时间应该小于 50ms
```

```
[root@obce-test-observer01 etc]# ntpstat ntpq -np
synchronised to NTP server (172.23.204.86) at stratum 12
time correct to within 51 ms
polling server every 1024 s
[root@obce-test-observer01 etc]#
```

```
[root@obce-test-observer01 etc]# ntpq -plgrep -E "\*|\=|remote"
remote      refid      st t when poll reach  delay  offset  jitter
-----
*172.23.204.86 LOCAL(0)    11 u  700 1024  377   0.187   0.032   0.017
[root@obce-test-observer01 etc]#
```

### 3.2.6. 修改目录归属

```
check /data/1: exist and mounted as individual disk ... PASS
check /data/1 owner: root ... EXPECT admin ... FAIL
TIPS: modify owner of /data/1
chown -R admin:admin /data/1
check /data/1 disk usage, total: 109G, used: 61M, use%: 1% < 91% ... PASS
check /data/log1: exist and mounted as individual disk ... PASS
check /data/log1 owner: root ... EXPECT admin ... FAIL
TIPS: modify owner of /data/log1
chown -R admin:admin /data/log1
```

OBserver 的服务器上都修改目录归属

```
chown -R admin:admin /data/1
chown -R admin:admin /data/log1
```

### 3.2.7. 部署前环境检查

```
sh precheck.sh -m ob
```

```
### SUMMARY OF ISSUES IN PRE-CHECK ###  
check SELinux status: Permissive ... EXPECT Disabled ... FAIL  
TIPS: disable SELinux  
    sed -i 's/enforcing/disabled/g' /etc/selinux/config  
    /usr/sbin/setenforce 0
```

在检查的最后，会将所有存在的问题一并列出，并且给出解决建议。可以参照建议解决问题，并再次运行环境检查。

*⚠注意：云主机因为使用最小资源配置，自检的时候可能会有资源方面的告警（mem, disk），可以忽略*

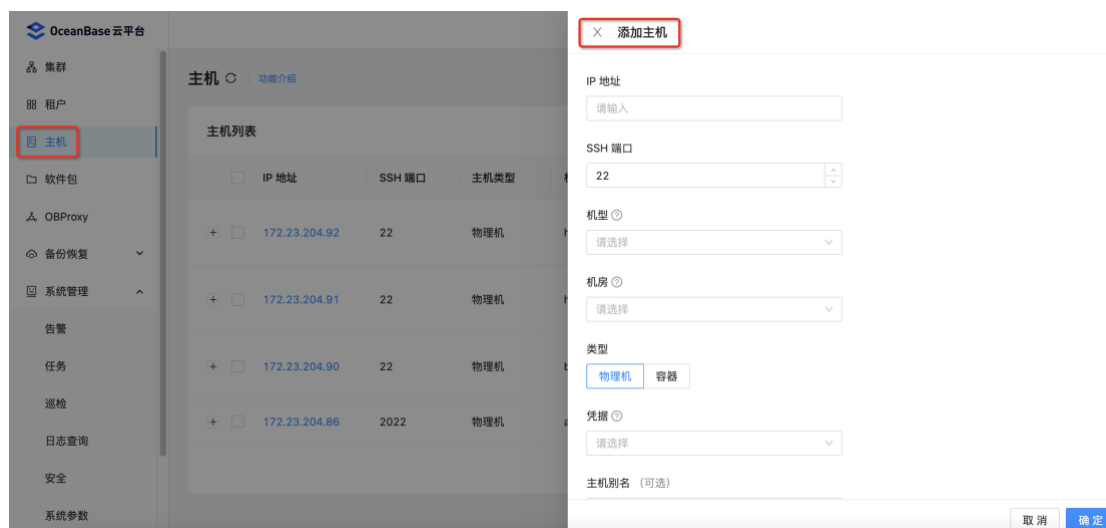
## 4. 在 OCP 上部署 OceanBase 集群

### 4.1. 在 OCP 上部署 OceanBase

下面可以通过 OCP 完成“添加主机”，“创建 OB 集群”，“部署 OBProxy”三个步骤。

#### 4.1.1. 添加主机

首先添加主机，将 3 台 OceanBase 服务器主机添加进来。这里可以设置服务器型号，所在机房和区域等。



这里要填写机型、机房/区域等信息，如果在生产环境，应该根据是实际情况填写。例如生产环境有 100 多台服务器，有浪潮、联想、阿里、华为不同品牌，那就创建不同的机型。生产环境是三地五中心（北京：西单、上地，杭州：西湖区、余杭区，深圳：前海），就按照实际情况创建机房（idc）/区域（region）等。

在实验练习环境，你可以根据自己的想象构建一个虚拟业务环境，用以模拟生产的环境。例如 3 台服务器，1 台型号是阿里的，部署在杭州西湖区，2 台型号是浪潮，部署在杭州余杭区。

选择凭据：这里其实就是写主机的登录用户信息，大家选择创建一个凭据，然后选择 root 用户，输入你的几台 OB 服务器 root 用户的密码。

### 4.1.2. 创建 OB 集群

创建 OB 集群，按照要求的填写好相应信息。

在 OCP 上部署 OB 集群的时候，需要上传 OB 的软件。如果 OB 软件在 OCP 服务器上，从 OCP 服务器下载 OB 软件到 PC，再从 PC 上传到 OCP web 界面，会耗费较长的时间。可以使用以下命令，将 OCP 服务器上的 OB 软件直接加载：

```
curl -i --user <用户名:密码> -X POST
'http://127.0.0.1:8080/api/v2/software-packages' --header 'Content-Type:
multipart/form-data' --form 'file=@/目录/文件名'
```

其中的参数：

```
--user <用户名:密码> : OCP web 界面的登录用户和密码，默认是 admin: aaAA11__
file=@/目录/文件名 : 存在在 OCP 服务器上的安装软件 (OB、OBProxy 等)
```

```
curl -i --user admin:aaAA11__ -X POST
'http://172.28.0.10:8080/api/v2/software-packages' --header 'Content-
Type:multipart/form-data' --form 'file=@/root/t-oceanbase-
antman/oceanbase-3.2.2-20220301220547.el7.x86_64.rpm'
```

```
curl -i --user admin:aaAA11__ -X POST
'http://172.28.0.10:8080/api/v2/software-packages' --header 'Content-
Type:multipart/form-data' --form 'file=@/root/t-oceanbase-
antman/obproxy-3.2.1-20211020153313.el7.x86_64.rpm'
```

创建集群的时候设置 primary zone 的优先级，这里的设定是给 sys 租户的。新版本只允许 primaryzone 有 1 台服务器。

在云主机（4C16G）的上面部署 OB 集群，需要限制资源的使用， 打开下面的“高级设置”，并且填写以下的参数（生产环境一般不需要）：

参数名	参数值	取值类型	取值范围	操作
memory_limit	15G	CAPACITY	-	删除
system_memory	5G	CAPACITY	[0M, ]	删除
cpu_count	32	INT	[0, ]	删除
__min_full_resource_pool_memory	2147483648	INT	[268435456, ]	删除
cache_wash_threshold	2GB	CAPACITY	[0B, ]	删除
net_thread_count	8	INT	[1, 100]	删除
workers_per_cpu_quota	4	INT	[2, 20]	删除
stack_size	512K	CAPACITY	[512K, 20M]	删除
+ 添加启动参数				

提交后可以到任务中查看部署进展，如果在某一步遇到问题，可以点击“查看日志”

OceanBase 云平台

任务 / Create primary OB cluster

任务 ID: 11824 任务状态: 运行中 当前进度: 17/49 开始执行时间: 2022年3月31日 19:57:08 任务累计耗时: 1分钟31秒 任务发起人: admin

Prepare install cluster ID: 12039 Prepare install cluster 开始: 19:57:08 耗时: 5秒

Pre check for install ob ID: 12049 Pre check for install... 开始: 19:57:14 耗时: 0.05秒

Pre check for install ob ID: 12058 Pre check for install... 开始: 19:57:14 耗时: 0.05秒

Pre check for install ob ID: 12073 Pre check for install... 开始: 19:57:14 耗时: 0.05秒

Wait node ID: 12053 Wait node 开始: 19:57:15 耗时: 0.01秒

Install dependencies ID: 12040 Install dependencies 开始: 19:57:16 耗时: 0.02秒

Install dependencies ID: 12056 Install dependencies 开始: 19:57:16 耗时: 0.01秒

Install OB rpm ID: 12034

EXECUTE 19:57:18

```
[pool-i-subtask-executor-thread-25,0x22888433c84db3,847484ebf7]
c.a.o.s.task.util.AgentAsyncTaskHelper :
try to request task result(EXECUTE), result:true,{"changed":true,
"packageInfo":{"architecture":"x86_64","buildNumber":
"2022031911151","fullPackageName":
"oceanbase-3.2.2-2022031911151.el7.x86_64","name":"oceanbase","os":
"el7","version":"3.2.2"}},null}
128 2022-03-31 19:58:12.772 INFO 59 ---
[pool-i-subtask-executor-thread-25,0x22888433c84db3,847484ebf7]
c.a.o.s.c.host.HostOperationService : rpm install done,
package info=PackageInfo{fullPackageName=oceanbase-3.2.2-
2022031911151.el7.x86_64, name=oceanbase, version=3.2.2,
buildNumber=2022031911151, os=el7, architecture=x86_64}
130 2022-03-31 19:58:12.774 INFO 59 ---
[pool-i-subtask-executor-thread-25,0x22888433c84db3,847484ebf7]
c.a.o.c.m.t.model.SubtaskInstanceEntity : Set state for subtask:
12034, current state: RUNNING, new state: SUCCESSFUL
```

OceanBase 云平台

任务 / Create primary OB cluster

任务 ID: 11824 任务状态: 运行中 当前进度: 17/49 开始执行时间: 2022年3月31日 19:57:08 任务累计耗时: 1分钟31秒 任务发起人: admin

Prepare install cluster ID: 12039 Prepare install cluster 开始: 19:57:08 耗时: 5秒

Pre check for install ob ID: 12049 Pre check for install... 开始: 19:57:14 耗时: 0.05秒

Pre check for install ob ID: 12058 Pre check for install... 开始: 19:57:14 耗时: 0.05秒

Pre check for install ob ID: 12073 Pre check for install... 开始: 19:57:14 耗时: 0.05秒

Wait node ID: 12053 Wait node 开始: 19:57:15 耗时: 0.01秒

Install dependencies ID: 12040 Install dependencies 开始: 19:57:16 耗时: 0.02秒

Install dependencies ID: 12056 Install dependencies 开始: 19:57:16 耗时: 0.01秒

Install OB rpm ID: 12034

EXECUTE 19:57:18

```
[pool-i-subtask-executor-thread-25,0x22888433c84db3,847484ebf7]
c.a.o.s.task.util.AgentAsyncTaskHelper :
try to request task result(EXECUTE), result:true,{"changed":true,
"packageInfo":{"architecture":"x86_64","buildNumber":
"2022031911151","fullPackageName":
"oceanbase-3.2.2-2022031911151.el7.x86_64","name":"oceanbase","os":
"el7","version":"3.2.2"}},null}
128 2022-03-31 19:58:12.772 INFO 59 ---
[pool-i-subtask-executor-thread-25,0x22888433c84db3,847484ebf7]
c.a.o.s.c.host.HostOperationService : rpm install done,
package info=PackageInfo{fullPackageName=oceanbase-3.2.2-
2022031911151.el7.x86_64, name=oceanbase, version=3.2.2,
buildNumber=2022031911151, os=el7, architecture=x86_64}
130 2022-03-31 19:58:12.774 INFO 59 ---
[pool-i-subtask-executor-thread-25,0x22888433c84db3,847484ebf7]
c.a.o.c.m.t.model.SubtaskInstanceEntity : Set state for subtask:
12034, current state: RUNNING, new state: SUCCESSFUL
```

**小 Tips:**

如果 OB 部署出现一些基础问题，例如说的磁盘挂载点没有配置好。可以采用以下的步骤：

- 1、 OCP 上放弃创建集群的任务
- 2、 OCP 上删除主机
- 3、 黑屏登录 OB 服务器，调整基础的配置，然后删除 /data/1, /data/log1, /home/admin 下面残存的安装遗留文件 (rm -rf )
- 4、 重启一下 OB 服务器。（切莫忘记 NTP 时钟同步，重启完成后记得 service ntpd start, 以及 ntpstat, ntpq -np 等命令检查时钟）
- 5、 从新在 OCP 白屏上添加主机、创建集群等

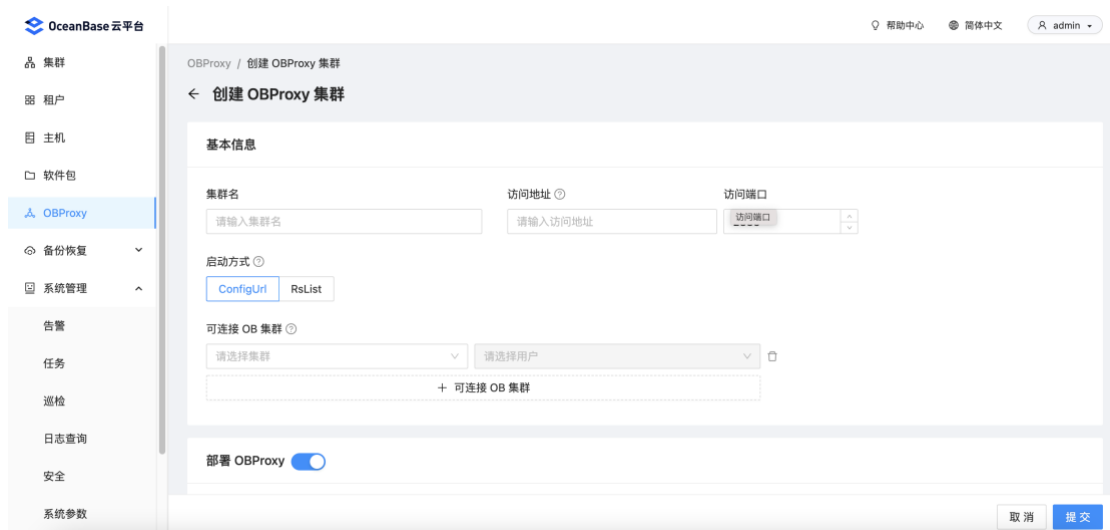
## 4.2. 在 OCP 上部署 OBProxy

### ■ 部署 OBproxy:

在生产环境，OBProxy 可能需要独立部署。在实验练习环境，我们将 OceanBase 和 OBProxy 部署在一起。可以在 3 台 OB 上都部署一个 OBProxy。

### ■ OCP 页面添加 OBProxy 集群:

访问地址参数：在生产环境，多台 obproxy 前可以有一个 F5 均衡，F5 均衡对外提供一个 ip 地址，这里的访问地址参数就是 F5 对外的 IP。我们实验环境没有 F5，那就选择一个 obproxy 的地址填写即可。



如果是云主机（4C16G）部署，点击下面的参数设置，增加一个参数，限制 OBProxy 最大可使用的内存。只是最大值，并不是实际分配或者使用值）





## 5. 创建租户

### 5.1. 打开一个 OCP 开关（仅限于实验环境）

实验环境中需要打开一个 OCP 的开关。因为规格过小的租户产生了很多不必要的问题，所以 OB 与 OCP 上都对租户的规格做了限制。OB 的 2.X 版本之后，默认只能创建 5G 以上的租户。在我们实验环境创建集群的时候，我们将这个限制参数调整成了 2G，也就是可以创建 2G 的租户。

最新的 OCP 版本在创建租户时进行自定义 Unit 规格，默认将小于 5G 内存的 unit 屏蔽了，因为生产环境中创建过小的资源用户产生了很多不必要的问题。在 OCP 2.5.1 之后增加了一个开关，打开该开关后在 OCP 上小于 5G 的 unit 才会显示出来。

**⚠ 注意：该操作仅限于演示或实验环境**

首先需要获得 OCP 所使用的 metadb 数据库的登录密码：

```
cd /root/t-oceanbase-antman/tools
./getpass.sh #获得meta 密码 xxxxxx
```

登录到 OCP 的 metadb 数据库，并打开开关（配置项全称：

ocp.operation.ob.tenant.allow-small-unit）：

```
mysql -h172.23.204.86 -P2883 -uroot@ocp_meta#obcluster -p'刚才通过
getpass 获得的密码'
show databases;
```

```
use ocp
select * from config_properties where `key` like '%small%'\G; #因为 key
是一个关键词，要在前后加着重号（数字键 1 旁边的反引号 `）
```

```
***** 1. row *****
```

```
id: 207
key: ocp.operation.ob.tenant.allow-small-unit
value: false
application: ocp
profile: default
label: master
fatal: 1
```

description: OB 的版本大于等于 2.0.0 时，创建租户是否允许使用小 Unit (小 Unit 指的是资源规格小于 5G 的 Unit)，默认为 false，在特殊情况下，比如 demo 演示等非生产环境，可以将其改为 true

```
mysql -h172.28.0.10 -P2883 -uroot@ocp_meta#obcluster -pkAO4vH@b5M
```

通过 update 命令修改参数

```
update config_properties set value='true' where `key` like '%small%';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MySQL [ocp]> select * from config_properties where `key` like '%small%'\G;
***** 1. row *****
id: 207
key: ocp.operation.ob.tenant.allow-small-unit
value: true
application: ocp
profile: default
label: master
fatal: 1
description: OB 的版本大于等于 2.0.0 时，创建租户是否允许使用小 Unit (小 Unit 指的是资源规格小于 5G 的 Unit)，默认为 false，在特殊情况下，比如 demo 演示等非生产环境，可以将其改为 true
create_time: 2022-03-30 10:43:41
update_time: 2022-04-02 11:13:32
description_en: Whether allowed to create tenant use small units when the OB version is greater than or equal to 2.0.0 (small units refer to units with resource specifications less than 5G) is set to false. In special circumstances, it can be used in non-production environments such as demonstrations, you can change it to "true"
is_oz_level: 0
1 row in set (0.00 sec)
```

OceanBase 云平台

帮助中心 简体中文 admin

集群

租户

主机

软件包

OBProxy

备份恢复

系统管理

告警

任务

巡检

日志查询

安全

系统参数

Unit 列表

请搜索规格名

规格名	最小 CPU 核数	最大 CPU 核数	最小内存 (GB)	最大内存 (GB)	规格类型	操作
B	0.25	0.25	1	1	系统默认	
LogOnlyNormal	1	1	2	2	系统默认	
LogOnlySystem	5	5	35	35	系统默认	
S0	0.5	0.5	2	2	系统默认	
S1	1.5	1.5	6	6	系统默认	
S2	3	3	12	12	系统默认	
S3	6	6	20	20	系统默认	
S4	12	12	40	40	系统默认	

共 8 条 1 10 条/页

## 5.2. 使用黑屏登录 OB

黑屏登录命令格式：

```
mysql -h<ip>. -u<用户名>@<租户名>#<集群名> -P2883 -p<密码> -c
obclient -h<ip>. -u<用户名>@<租户名>#<集群名> -P2883 -p<密码> -c
```

登录黑屏的几种方式（先登录 sys 租户）：

可以在 OCP 服务器上安装 obclient, mysql 客户端等

OCP (mysql、obclient 客户端) -----> OBproxy

```
mysql -h<obproxy ip> -uroot@sys#obcp_test -P2883 -p <passwd> -c
```

在 PC 上使用 mysql 客户端登录，配置 OCP 服务器作为代理

PC (MYSQL 客户端) ----> OCP (代理) ----> OBproxy

先在 OCP 服务器上设置代理：

```
ncat --sh-exec "ncat <obproxy ip> 2883" -l 2883 --keep-open &
```

然后从远端个人 PC 上使用 mysql 登录：

```
mysql -h<ocp ip> -uroot@sys#obcp_test -P2883 -p <passwd> -c
```

如果 OB 云主机也有公网 IP，可以不必设置代理直接 mysql 登录。（云主机要放开 2883 端口）

## 5.3. 使用白屏或黑屏创建租户

### 5.3.1. 使用白屏创建租户

云主机（4C16G）的请注意以下事项：

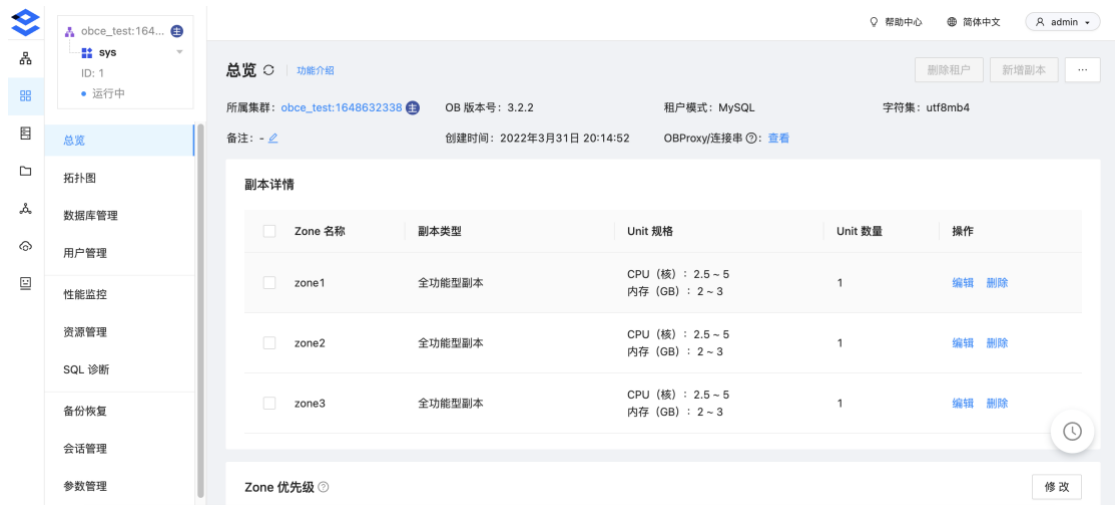
- 创建普通租户的时候请注意限制大小，每个租户 CPU 可以设置为 5，内存不要超过 2G。

- 云主机是 4C16G
- 我们创建集群的时候设定 observer 的内存(memory\_limit)是 15 G.
- 系统租户内存 (ID500 租户 system\_memory) 我们设定的是 5G (15-5=10)
- 管理 sys (ID1 租户) 租户默认是内存是 2-3, 一般是用了 2.5G (15-5-2.5=7.5)
- 剩下来给普通租户的已经是 7.5G 了。可以创建 3 个 2G 租户。

```
ncat --sh-exec "ncat <obproxy ip> 2883" -l 2883 --keep-open &
```

- sys 管理租户 (ID1 租户) 占用内存 2.5G, 可以再降低到 2G, 再释放 0.5, 这样的话可以再多创建一个 2G 的普通租户。(可选)

- 白屏中看管理 sys 租户的资源, 可以在白屏直接修改



- 黑屏中修改

```
mysql -h172.28.0.11 -P2883 -uroot@sys#oceanbase:1661902699 -pOBBCP@bj@2022 -Doceanbase
```

```
select * from __all_unit_config; #查看所有的 unit 规格配置, 其中 sys_unit_config 是系统 sys 管理租户使用的 unit
```

```
MySQL [oceanbase]> select * from __all_unit_config;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|gmt_create|gmt_modified|unit_config_id|name|max_cpu|min_cpu|max_memory|min_memory|max_lops|min_lops|max_disk_size|max_session_num|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|2022-03-31 20:11:02.454499|2022-03-31 20:11:02.454499|1|sys_unit_config|5|2.5|3221225472|268435456|10000|5000|99180609536|9223372036854775807|
|2022-04-01 19:28:53.834093|2022-04-01 19:28:53.834093|1001|mini|5|5|2147483648|2147483648|10000|1000|10737418240|1000000|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

```
alter resource unit sys_unit_config
max_memory='2G',min_memory='2G',max_cpu=5;
```

### 5.3.2. 使用黑屏创建租户

- 创建 unit(5C2G):

```
create resource unit mini max_cpu=5, min_cpu=5, max_memory='2G',
min_memory='2G', max_iops=10000, min_iops=1000, max_session_num=1000000,
max_disk_size='10G';
```

- 创建 resource pool:

```
create resource pool mini_pool_t1 unit=mini ,unit_num=1;
create resource pool mini_pool_t2 unit=mini ,unit_num=1;
```

■ 创建 MySQL 租户:

```
create tenant obce_t1 charset='utf8mb4', replica_num=1,
zone_list=('zone1,zone2,zone3'), primary_zone='zone1,zone2,zone3',
resource_pool_list=('mini_pool_t1') set ob_tcp_invited_nodes='%';
```

设置密码

```
obclient -h172.28.0.71 -P2883 -uroot@obce_t1#obcp
set password=password('OBCEP@bj@2022')
```

■ 创建 Oracle 租户:

```
create tenant obce_t2 charset='utf8mb4', replica_num=1,
zone_list=('zone1,zone2,zone3'), primary_zone='zone1,zone2,zone3',
resource_pool_list=('mini_pool_t2') set ob_tcp_invited_nodes='%',
ob_compatibility_mode='oracle';
```

设置密码

```
obclient -h172.28.0.71 -P2883 -usys@obce_t2#obcp
alter user sys identified by oracle;
```

在黑屏创建好租户后，在 OCP 白屏也会看得到，不过同步需要一点点时间。

OceanBase 的基础安装和配置基本就完成了，下面大家可以开始其它的实验练习了。

**⚠注意：**云主机实验环境，在 `sys` 租户下执行这两个命令。因为我们实验环境留给 `log` 的空间比较小，避免被 `log` 日志塞满。以下两条命令是启动日志文件循环，并且只保留 10 个日志文件。

```
alter system set enable_syslog_recycle=true;
alter system set max_syslog_file_count=10;
```