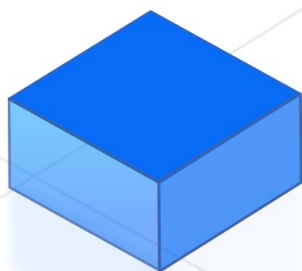
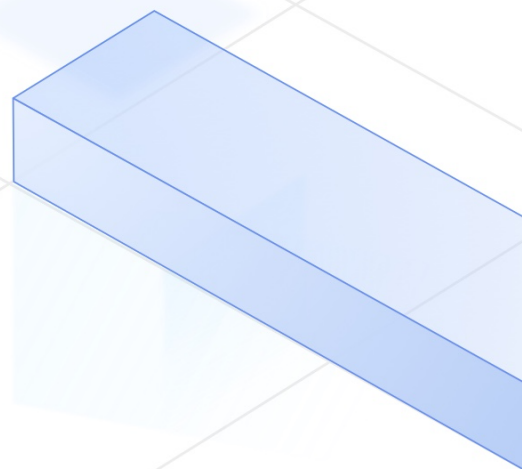
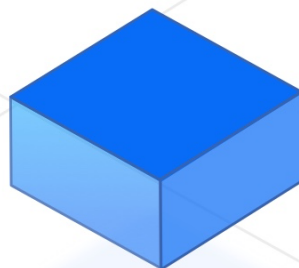
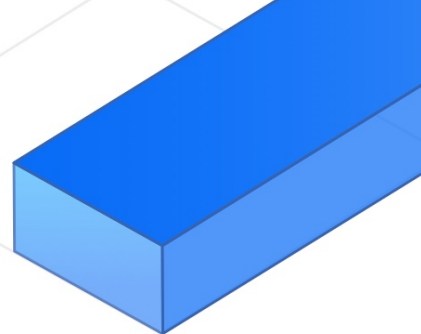


**OCEANBASE**



# OceanBase 云主机环境基础安装与配置

## 实验指导

| 产品版本：V 1

# 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

声明 .....	1
1. 实验概述 .....	1
2. 实验准备 .....	2
2.1. 实验架构 .....	2
2.2. 设备需求 .....	2
3. 实验：OB 集群基础安装 .....	4
3.1. 准备安装包(此文档针对的是 OCP2.5, OB2.2.75 版本) .....	4
3.2. 服务器基本设置 .....	5
3.3. 服务器环境配置 .....	13
3.4. 单点 OCP 的安装 .....	15
3.5. 在 OCP 上部署 OceanBase 集群 .....	18
3.6. 创建租户和开始实验练习 .....	21

# 1. 实验概述

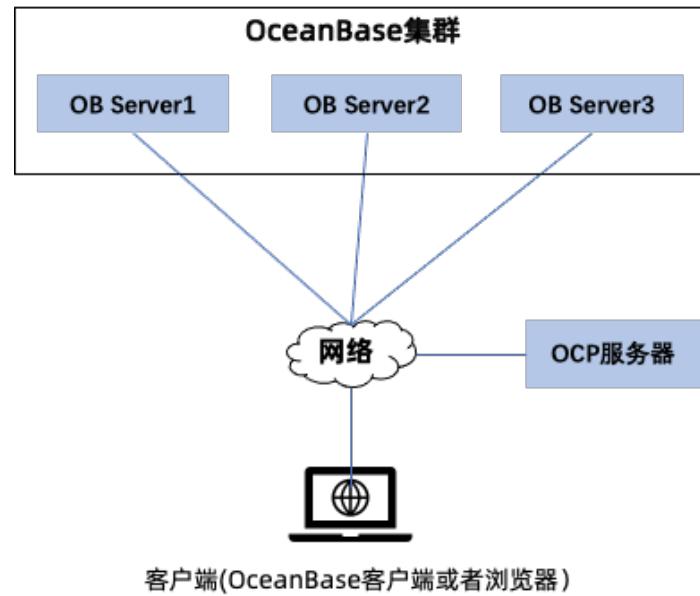
本手册主要帮助讲师和学员搭建一套基本的 OceanBase 实验环境（1 OCP + 3 Observer），可以与 OBCA 课程，也可以用于 OBCP 课程前的环境准备，也可以用于其它 OB 产品技术培训。

通过本实验指导，大家能够完成以下的工作：

- 1、 基于云主机环境搭建起一套 OceanBase 的基础实验环境：1 OCP+3 Observer，以及 OBProxy.
- 2、 完成基本的集群配置、租户配置等
- 3、 能够通过黑屏或白屏对 OB 集群进行管理配置。

## 2. 实验准备

### 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；

### 2.2. 设备需求

#### 一、 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。这里简单摘录相关信息：

服务器类型	数量	功能最低配置
OCP 服务器	1 台	32C, 128G, 1.5TB 存储
OceanBase 服务器 (Observer)	3 台	32C, 128G, 1.2TB 存储

操作系统准备：

Linux 操作系统	版本	服务器架构
------------	----	-------

AliOS	7.2 及以上	x86_64（包括海光），ARM_64 （仅支持鲲鹏、飞腾）
CentOS / Red Hat Enterprise Linux	7.2 及以上	x86_64（包括海光），ARM_64 （仅支持鲲鹏、飞腾）
SUSE Enterprise Linux	12SP3 及以上	x86_64（包括海光），ARM_64 （仅支持鲲鹏、飞腾）
Debian / Ubuntu	8.3 及以上	x86_64（包括海光），ARM_64 （仅支持鲲鹏、飞腾）

## 二、 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。不过我们一般都希望能够尽量节省成本，以下是实验室验证的能够安装 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的部分，这些操作只是用于学习环境。

云主机环境因为增加了额外的操作，比实际生产环境还多复杂了一些。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）。

## 3. 实验：OB 集群基础安装

说明：基于物理服务器的安装，大家可以在 OB 官网参考最新的部署文档。这篇安装指导手册，一方面包含正式的安装步骤（大家可以和官网文档对照，最终以官网最新的文档为准），一方面包含在云主机的最低配置下的一些特殊配置修改和说明。

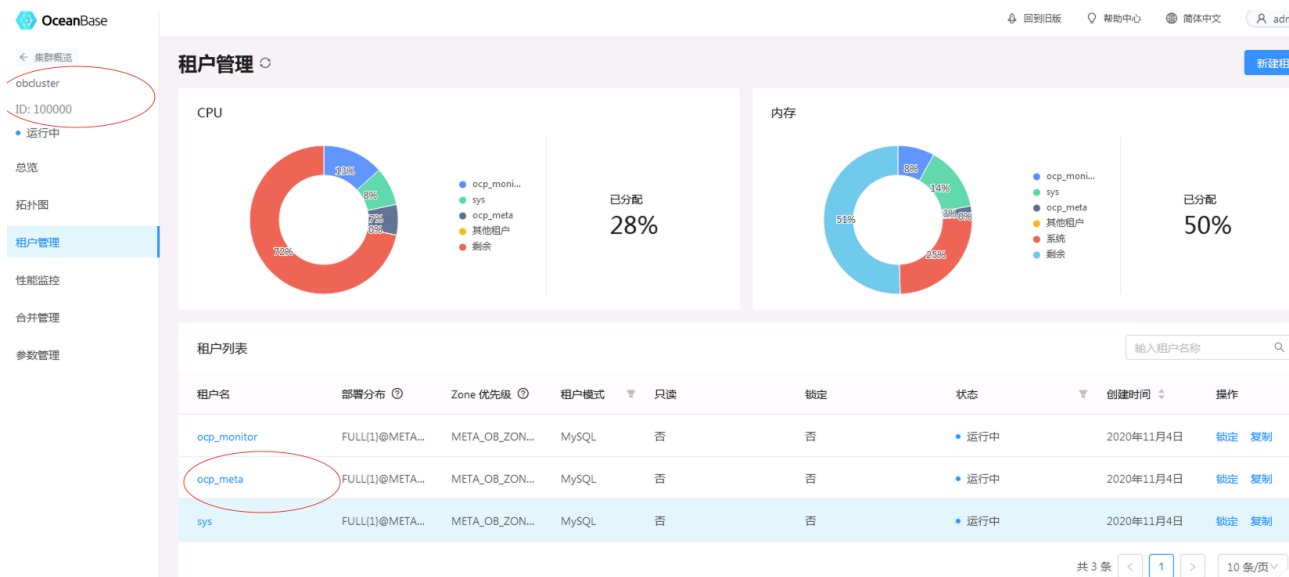
### 3.1. 准备安装包(此文档针对的是 OCP2.5, OB2.2.75 版本)

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

	文件	描述
OCP 服务器	t-oceanbase-antman-1.3.6-1919351.aliyos7.x86_64	自动化部署工具
	ocp-all-in-one25-20210105.tar	基于 docker 部署的 OCP 文件
	ob-docker2273_202101.tar	基于 docker 部署的 OB 文件
	obproxy-docker173_20200603.tar	基于 docker 部署的 OBProxy 文件

这里需要说明一下，为何安装 OCP 服务器却需要 oceanbase（observer）和 OBProxy 的安装包。其实 OCP 作为一个网管服务器，也是架构在一个 OB 集群上的（这个用于 OCP 网管的集群既可以是单机 OB，也可以是 3 台 OB）。不过 OCP 服务器中的各个组件（OCP 服务、Oceanbase 服务，OBProxy 服务）都是安装在 Docker 上。OCP 安装好之后，大家打开 OCP 的管理页面，如下，就会发现有一个默认的集群存在，这个集群下有 sys 租户，ocp\_meta 租户，ocp\_monitor 租户等，他们都是 OCP 的集群和租户。

(OCP 有两个 sys 租户，一个是 ID 为 500 的系统租户，白屏不可见，在黑屏中的一些命令中可以看到；另外一个用于 ID 为 1 的管理 sys 租户，在白屏中可以看到，黑屏登录管理的也是这个 sys 租户)



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-2.2.75-20210108170301.el7.x86_64	OceanBase 服务器安装文件
OBproxy	obproxy-1.9.1.2-1919592.el7.x86_64	OBProxy 安装文件

## 3.2. 服务器基本设置

### 3.2.1: 物理服务器设置

大家可以参考官网的指导，这里主要描述的是生产环境中的配置要求。

<https://www.oceanbase.com/docs/apasadb-for-oceanbase/docs/V2.2.50/prepare-server>

### 3.2.2: 云主机服务器基本设置

**3.2.2.1 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
OB docker 总计		13 (可以放大到 16)	60
Obproxy docker		4	2
OCP docker		8	16
总计		25->16 (共享模式)	78->64 (共享模式)

在安装 OCP 前，我们会参照以上的方案配置 OCP 的资源分配文件。可以看到即使在这种搞状况下，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多的压力。

OCP 服务器的磁盘最小分配：

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（取决于所需存储的数据大小）	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		

下面以阿里云的 ECS 申请为例演示 OCP 服务器的申请与基础的配置：

当前代 所有代 16 vCPU 64 GiB

筛选 16 vCPU 选择内存 搜索规格名称, 如: ecs.g5.large I/O 优化实例 是否支持IPv6

规格族	实例规格	vCPU	内存	GPU/FPGA	本地存储	处理器主频	内网带宽	内网收发包	存储IOPS	IPv6	参考价格	处理器型号
<input type="radio"/> 本地 SSD 网络增强型 i2gne	ecs.i2gne.4xlarge	16 vCPU	64 GiB	-	1 * 1788 GiB	2.5 GHz	5 Gbps	150 万 PPS	-	否	¥ 6.454 /时	Intel Xeon(Skylake) Platinum 8163
<input checked="" type="radio"/> 本地SSD型 i2g	ecs.i2g.4xlarge	16 vCPU	64 GiB	-	1 * 1788 GiB	2.5 GHz/2.7 GHz	3 Gbps	150 万 PPS	-	否	¥ 6.146 /时	Intel Xeon(Skylake) Platinum 8163
<input type="radio"/> 本地SSD型 i1	ecs.i1.4xlarge	16 vCPU	64 GiB	-	2 * 416 GiB	2.5 GHz	3 Gbps	50 万 PPS	-	否	¥ 11.77 /时	Intel Xeon E5-2682v4
<input type="radio"/> 本地SSD型 i1	ecs.i1-c5d1.4xlarge	16 vCPU	64 GiB	-	2 * 1456 GiB	2.5 GHz	3 Gbps	40 万 PPS	-	否	¥ 14.44 /时	Intel Xeon E5-2682v4
<input type="radio"/> 大数据网络增强型	ecs.d1ne.4xlarge	16 vCPU	64 GiB	-	8 * 5500 GiB	2.5 GHz	12 Gbps	160 万 PPS	-	是	¥ 13.37 /时	Intel Xeon E5-2682v4

ecs.i2g.4xlarge (16 vCPU 64 GiB, 本地SSD型 i2g)  
本地存储有丢失数据风险, 不适用于应用层没有数据冗余架构的使用场景; 创建后暂不支持变更规格。您可点击查看 本地存储 详细介绍。

镜像

公共镜像

自定义镜像

共享镜像

镜像市场

CentOS 7.2 64位 安全加固

存储

系统盘

高效云盘 40 GIB 2120 IOPS 随实例释放

不同云盘性能指标不同, 查看 各云盘性能指标

本地存储

磁盘种类	容量 GiB	IOPS	设备名	块数
NVMe SSD	1788	240000	随机分配	1

当前实例的数据缓存盘是本地盘, 有丢失数据的风险 (比如 NVMe SSD 故障或宿主机宕机时) 如果您的应用不能做到数据可靠性的架构, 我们强烈建议您使用云盘搭建您的实例

为降低因为物理服务器宕机引起业务受损, 建议您将本地盘实例放置在 部署集 内, 提高业务可用性

数据盘 您已选择 0 块盘, 还可以选择 16 块盘

+ 增加一块数据盘

以上创建云主机由管理员/班主任完成。 学员只需要活动分配到的云主机信息： IP 地址与登录账号。

学员从 PC 登录远端 OCP 服务器：

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

然后使用 fdisk 查看和创建相关的分区，以此次阿里云创建的云主机为例。 分别创建了 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

```
[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：

```
[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,

```
# /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。

```
[root@xinran-0CP ~]# mount -a
[root@xinran-0CP ~]#
[root@xinran-0CP ~]#
[root@xinran-0CP ~]# 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
```

### 3.2.2.2 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)
Observer	sys (500 租户)	/	system_memory=5
	os	2	/
	sys 租户 (管理)	5	默认 (2.5) G, 可以不修改, 或者待启动后改为 2G
	业务租户	5	每个租户分配 2G
OB server 总计		cpu_count=32 (逻辑 CPU)	Memory_limit=15G
OBproxy (在实验环境与 OB 部署在一起 P)			Proxy_mem_limit=1G(最大 1G)
云主机		4C(物理)	16G(物理)

*OceanBase* 云主机的磁盘分配建议方案:

载点	大小（GB）	用途	磁盘格式
/home	5		建议 ext4
/data/log1	20（无刚性需求，看保留日志要求）	OCP 元数据库 OB 日志盘	建议 ext4
/data/1	110（取决于存储的数据大小，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	135G+其它系统空间，总数 170G		

注：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。

下面以阿里云的 ECS 申请为例演示 OB 服务器的申请与基础的配置：

当前代 所有代 4 vCPU 16 GiB

筛选 4 vCPU 16 GiB 搜索规格名称，如：ecs.g5.large I/O 优化实例 是否支持IPv6

规格族	实例规格	vCPU	内存	GPU/FPGA	本地存储	平均基准CPU 计算性能	处理器主 频	内网带宽	内网收发包	存储IOPS	IPv6	参考价格	处理器型号
高主频通用型 hfg7	ecs.hfg7.xlarge	4 vCPU	16 GiB	-	-	-	3.3(3.8 GHz)	最高 10 Gbps	100 万 PPS	3 万	是	¥ 1.158 /时	Intel Xeon Platinum (Cooper Lake) 8369
通用平衡增强型 g6e	ecs.g6e.xlarge	4 vCPU	16 GiB	-	-	-	2.5 GHz(3.2 GHz)	最高 10 Gbps	100 万 PPS	4.2 万	是	¥ 1.1 /时	Intel Xeon(Cascade Lake) Platinum 8269CY
高主频通用型 hfg6	ecs.hfg6.xlarge	4 vCPU	16 GiB	-	-	-	3.1 GHz(3.5 GHz)	最高 5 Gbps	50 万 PPS	2 万	是	¥ 1.1 /时	Intel Xeon (Cascade Lake) Platinum 8269
通用型 g6	ecs.g6.xlarge	4 vCPU	16 GiB	-	-	-	2.5 GHz(3.2 GHz)	最高 5 Gbps	50 万 PPS	2.1 万	是	¥ 1.0 /时	Intel Xeon(Cascade Lake) Platinum 8269CY
通用型 g5	ecs.g5.xlarge	4 vCPU	16 GiB	-	-	-	2.6 GHz(3.2 GHz)	最高 10 Gbps	100 万 PPS	2 万	是	¥ 0.98 /时	Intel Xeon(Cascade Lake) Platinum 8269CY

ecs.g6e.xlarge (4 vCPU 16 GiB, 通用平衡增强型 g6e)

当前所选实例规格在 华东 1 可用区 H 的总配额最多，已开通 35 台，最多还可开通 29 台。如需更多配额，您可前往控制台提升>

建议您使用 弹性伸缩 Auto Scaling 产品 来创建/伸缩 ECS 实例。创建伸缩组时扩容策略可选择“成本优化策略”，更加经济高效。去创建>

镜像

公共镜像 自定义镜像 共享镜像 镜像市场

CentOS 7.2 64位 安全加固

存储

系统盘

ESSD云盘 40 GiB 2280 IOPS 性能级别：PL0 (单盘IOPS性能上限1万) 随实例释放

不同云盘性能指标不同，查看 各云盘性能指标>

数据盘 您已选择 1 块盘，还可以选择 15 块盘

每台实例将自动挂载如下相同配置的数据盘

ESSD云盘 130 GiB 8300 IOPS 性能级别：PL1 (单盘IOPS性能上限5万) 数量：1 自动分配设备名 随实例释放

当前区域已购买 ESSD云盘 5040 GB，还可购买的容量额度为：60106 GB

增加一块数据盘

以上创建云主机由管理员/班主任完成。学员只需要活动分配到的云主机信息：IP 地址与登录账号。

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(/etc/fstab)。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
```

提醒：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）

### 3.3. 服务器环境配置

在每台服务器的/root 目录下创建 t-oceanbase-antman 目录，并将相应的安装包上传到该目录。对于云主机，可以先上传到 OCP 服务器（OCP 服务器的带宽可以设置高一些），然后从 OCP 服务器经过内网拷贝到其它 OB 服务器，这样速度更快并节省费用。

1、在 OCP 与 OceanBase 的服务器上都安装自动化部署工具，例如：

```
# rpm -ivh t-oceanbase-antman-1.3.3-1896174.alios7.x86_64.rpm
```

2、为每台服务器都添加 admin 用户：

```
# cd /root/t-oceanbase-antman/clonescripts
```

```
# ./clone.sh -u
```

默认密码是 admin，建议立刻修改，尤其是云主机容易受到攻击。

3、对每台服务器的操作系统内核进行设置：

```
# cd /root/t-oceanbase-antman/clonescripts
```

```
OCP 服务器: # ./clone.sh -r ocp -c
```

```
OceanBase 服务器: # ./clone.sh -r ob -c
```



#### 4、安装依赖包

OCP 服务器: `#!/clone.sh -r ocp -m`

OceanBase 服务器: `#!/clone.sh -r ob -m`

#### 5、配置 NTP 时钟同步, 保证所有节点的时钟偏差在 100ms 以内

选择一台服务器作为 NTP server (例如 OCP 172.18.6.38) , 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:

将自己设置为 ntp server, 并允许 172.18.0.0 网段的 ntp client 访问 (这里的 NTP client 例如是 OB) :

```
restrict 172.18.0.0 mask 255.255.0.0 nomodify
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

其它所有服务器作为 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:

```
server 172.18.6.38
fudge 172.18.6.38 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。

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)。

#### 6、在 ocp 服务器上安装 docker

OCP 服务器: `#!/clone.sh -i`

#### 7、部署前环境检查

OCP 服务器: `sh precheck.sh -m ocp`

OceanBase 服务器: `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。

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。

## 3.4. 单点 OCP 的安装

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。

### 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）

另外今天使用的 antman 安装脚本有个小 bug，如果密码中有 \$ 符号的话会被识别成变量符号，建议大家密码中不要带 \$。大家在这个阶段可以将 root/admin 用户的密码改为弱密码，安装完成之后即刻改为强密码，因为 OCP 有公网 IP 暴露，随时可能被攻击掉。（研发已经修改，不久新的版本就没有这个问题了）

```
SINGLE_OCP_MODE=TRUE
##### 根据环境必须修改 / MUST CHANGE ACCORDING ENVIRONMENT #####
##### 填写机器IP和root/admin密码 / Edit Machine IP and Password Of root /admin #####
ZONE1_RS_IP=172.18.1.1
OBSERVER01_ROOTPASS=123123
OBSERVER01_ADMINPASS=admin
SSH_PORT=22

##### 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
```

- 如果是物理机的部署，可以参考以下的 docker 资源分配（以下是 OCP 相关配置，相见“3.1 主备安装包”的介绍）：  
observer docker: 32C 128G,  
ocp docker: 8C 16G  
OBProxy docker: 4C12G

物理机资源为推荐最低配置，可做自定义调整，OBServer 不低于 24C100G，OCP 不低于 4C8G，OBProxy 不低于 2C10G。

```
##### 根据服务器CPU、内存设置容器资源编排 / Allocate Container Resource
s According To Server #####
OB_docker_cpus=32
OB_docker_memory=128G
OCP_docker_cpus=8
OCP_docker_memory=16G
OBProxy_docker_cpus=4
OBProxy_docker_memory=12G
```

➤ 如果是云主机，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*

*OCP\_docker\_cpus=8, OCP\_docker\_memory=16G*

*OBProxy\_docker\_cpus=4, OBProxy\_docker\_memory=2G*

详情见前面 3.2.2 章节的描述。

OCP 的配置文件在 /root/t-oceanbase-antman 目录下，2.X 版本的名称是 obcluster.conf。另外有一点要注意的是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 3.1 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
docker_image_package=ob-docker2273_202101.tar.gz
OB_image_REPO=reg.docker.alibaba-inc.com/antman/ob-docker
OB_image_TAG=OB2273_x86_20201214
# OCP docker
ocp_docker_image_package=ocp-all-in-one25-20210105.tar.gz
OCP_image_REPO=reg.docker.alibaba-inc.com/oceanbase/ocp-all-in-one
OCP_image_TAG=2.5.0-1918031
# OBPROXY docker
obproxy_docker_image_package=obproxy-docker173_20200603.tar.gz
obproxy_image_REPO=reg.docker.alibaba-inc.com/antman/obproxy
obproxy_image_TAG=OBP173_20200603_1923
```

*image\_package* 填写你获得安装软件的名称，例如：

*ob-docker2273\_202101.tar.gz*

*obproxy-docker173\_20200603.tar.gz*

*ocp-all-in-one25-20210105.tar.gz*

*REPO* 和 *TAG* 可以填写如下的信息

*reg.docker.alibaba-inc.com/oceanbase/ocp-all-in-one:2.5.0-1918031*

*reg.docker.alibaba-inc.com/antman/ob-docker:OB2273\_x86\_20201214*

*reg.docker.alibaba-inc.com/antman/obproxy:OBP173\_20200603\_1923*

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。（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`

76 行和 100 行 (请注意要修改的是两行):

```
-e
OPTSTR="\cpu_count=$OB_docker_cpus,memory_limit=$OB_docker_memory,__min_full_resource_pool_
memory=1073741824,_ob_enable_prepared_statement=false,memory_limit_percentage=90\" \
改成
-e
OPTSTR="\cpu_count=$OB_docker_cpus,memory_limit=$OB_docker_memory,__min_full_resource_pool_
memory=1073741824,_ob_enable_prepared_statement=false,memory_limit_percentage=90,system_me
mory=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 1330 行
# system_memory is 50G default
#[ "${OB_docker_memory%G}" -lt 64 ] && { antman_log "docker ob default system_memory is 50G,
docker memory limit(${OB_docker_memory}) too small." "ERROR"; exit 1; }
```

## 2、开始部署 OCP

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 <-----执行 1-8 步
```

```
[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`)

## 3、部署后检查

OCP 的访问地址: [http://<OCP\\_ip>:8080](http://<OCP_ip>:8080) <----云主机的话注意安全策略放开 8080 端口

默认用户名 admin, 密码 root。

打开 ocp 的 web 管理页面后, 就有一个集群和 3 个租户存在。(见上面 3.1 章节的描述)

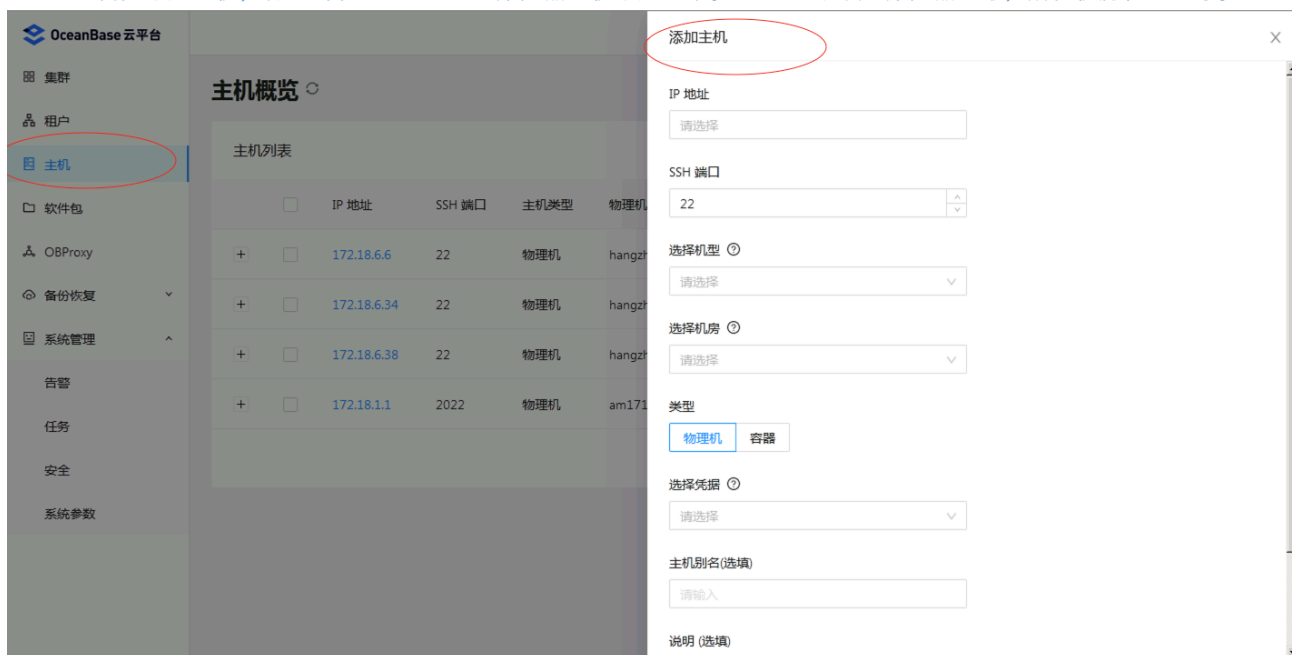


### 3.5. 在 OCP 上部署 OceanBase 集群

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。

### 3.5.1: 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。

### 3.5.2: 创建 OB 集群

创建 OB 集群，按照要求的填写好相应信息。



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）：

参数修改

memory_limit	15G
system_memory	5G
cpu_count	32
__min_full_resource_pool_memory	2147483648
cache_wash_threshold	2GB
net_thread_count	8
workers_per_cpu_quota	4
stack_size	512K

+ 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”。



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）

部署模式

机房	机型 (选项)	主机
请选择	请选择	自动分配
0	请选择	0
请选择	请选择	自动分配
0	请选择	0

+ 添加 OBProxy

参数设置 🔴

启动参数修改

请选择参数 参数值

+ 添加启动参数

其他参数修改

proxy\_mem\_limited(类型: CAPACITY) 1G

+ 添加其他参数

### 3.6. 创建租户和开始实验练习

OB 集群创建之后，下一步可以创建租户。

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 OCP2.5 版本在创建租户时进行自定义 Unit 规格的时候，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。（未来会考虑为学习使用增加一个开关）

所以我们在测试和学习环境，现阶段使用黑屏登录 sys 管理租户后创建普通租户。

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
```

登录黑屏的方式（先登录 sys 租户）：

1、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
```

2、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

OCP 服务器上设置代理命令：



```
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
```

3、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)

**云主机 (4C16G) 的请注意以下事项：**

1、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。

2、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

Zone 名称	副本类型	Unit 规格	Unit 数量	操作
zone1	全功能型副本	CPU (核): 2.5 ~ 5 内存 (GB): 2 ~ 3	1	编辑 删除
zone2	全功能型副本	CPU (核): 2.5 ~ 5 内存 (GB): 2 ~ 3	1	编辑 删除
zone3	全功能型副本	CPU (核): 2.5 ~ 5 内存 (GB): 2 ~ 3	1	编辑 删除

也可以在黑屏中修改:

```
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。
```

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

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;
```

创建租户:

```
create tenant obcp_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='%';
```

在黑屏创建好租户后，在 OCP 白屏也会看得到，后续可以在白屏上操作该租户。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。

补充：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;
```