

**OCEANBASE**



# OceanBase 数据库

部署数据库

| 产品版本：V4.0.0

| 文档版本：20230505

# 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

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。

# 通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	<b>危险</b> 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	<b>警告</b> 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	<b>注意</b> 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	<b>说明</b> 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击 <b>设置&gt; 网络&gt; 设置网络类型</b> 。
粗体	表示按键、菜单、页面名称等UI元素。	在 <b>结果确认</b> 页面，单击 <b>确定</b> 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1	部署简介	8
2	部署流程	9
3	准备服务器	10
4	服务器配置	11
4.1	准备操作系统	11
4.2	BIOS 设置	11
4.3	磁盘挂载	13
4.4	网卡设置	14
4.5	软件要求	14
5	准备安装包	15
5.0.0.1	说明	15
6	OceanBase 集群高可用方案简介	16
6.1	同城三机房三副本部署	16
6.1.1	特点	16
6.1.2	部署方案示意图	16
6.2	三地五中心五副本部署	16
6.2.1	特点	16
6.2.2	部署方案示意图	17
7	安装 oat-cli	18
7.1	背景信息	18
7.2	操作步骤	18
8	通过 oat-cli 配置部署环境	19
8.1	添加 admin 用户	19
8.2	修改操作系统内核配置	19
8.2.1	OCP 服务器	19
8.2.2	OBServer 服务器	19
8.3	安装依赖包	19
8.3.1	OCP 服务器安装依赖包	19
8.3.2	OBServer 服务器安装依赖包	19
8.4	OCP 与 OMS 服务器安装 Docker	20
9	配置时钟源	21
9.1	前提条件	21
9.2	操作步骤	21
10	上传安装包	25
10.0.0.1	说明	25
11	配置主机信息	26
11.1	操作步骤	26

12	部署前环境检查	27
12.1	OCP 服务器	27
12.2	OBServer 服务器	27
12.3	备份与恢复服务器	27
12.4	OMS	27
13	部署说明	28
13.1	描述	28
13.2	单节点部署	28
13.3	三节点部署	28
13.4	多 AZ 部署	28
14	生成配置文件	29
14.1	配置环境变量	29
14.1.1	描述	29
14.1.2	适用版本	29
14.1.3	使用 init_obcluster_conf.sh 生成配置文件	29
14.2	单节点部署	30
14.3	三节点部署	31
14.4	多 AZ 模式部署	33
15	开始部署	35
15.1	安装	35
15.2	异常处理	36
16	部署后检查	37
16.1	访问 OCP	37
16.2	检查 MetaDB 集群信息	37
16.3	检查监控状态	38
17	部署单副本 OceanBase 集群	40
17.1	前提条件	40
17.2	操作步骤	40
17.2.1	安装目录和软件	40
17.2.2	初始化 OceanBase 集群	43
18	部署三副本 OceanBase 集群	48
18.1	前提条件	48
18.2	操作步骤	48
18.2.1	安装目录和软件	48
18.2.2	初始化 OceanBase 集群	51

19	部署 OBProxy	57
19.1	操作步骤	57
19.1.1	步骤一：安装 OBProxy RPM 包	57
19.1.2	步骤二：配置目录	57
19.1.3	步骤三：初始化 OBProxy 账户	60
19.1.4	步骤四：启动 OBProxy	60
19.2	后续操作	62
19.2.1	连接 OceanBase 数据库	62
19.2.2	(可选) 调整 OBProxy 参数	63
19.3	相关文档	65
20	查看业务租户可用资源	66
20.1	操作步骤	66
21	创建租户	68
21.1	描述	68
21.2	操作步骤	68
21.2.1	步骤一：创建资源单元	68
21.2.1.1	语法	68
21.2.1.2	参数解释	69
21.2.1.3	示例	69
21.2.2	步骤二：创建资源池	70
21.2.2.1	语法	70
21.2.2.2	参数解释	70
21.2.2.3	示例	70
21.2.3	步骤三：创建租户	71
21.2.3.1	语法	71
21.2.3.2	参数解释	72
21.2.3.3	示例	74
21.2.4	步骤四：设置管理员用户密码	74
22	部署后验证	75
23	部署说明	76
24	修改配置文件	77
24.1	操作步骤	77
25	开始部署	79
25.1	前提条件	79
25.2	操作步骤	79
25.3	后续操作	80
26	部署后检查	82
26.1	背景信息	82
26.2	操作步骤	82
27	部署说明	83
28	修改配置文件	84
28.1	操作步骤	84

29 开始部署 .....	86
29.1 前提条件 .....	86
29.2 操作步骤 .....	86
29.3 后续操作 .....	87
30 部署后检查 .....	89
30.1 访问 ODC .....	89
30.2 通过 ODC 连接 OceanBase 数据库 .....	89

# 1 部署简介

OceanBase 数据库是阿里巴巴和蚂蚁金服 100% 自主研发的金融级分布式关系数据库，其在普通硬件上实现了金融级高可用。

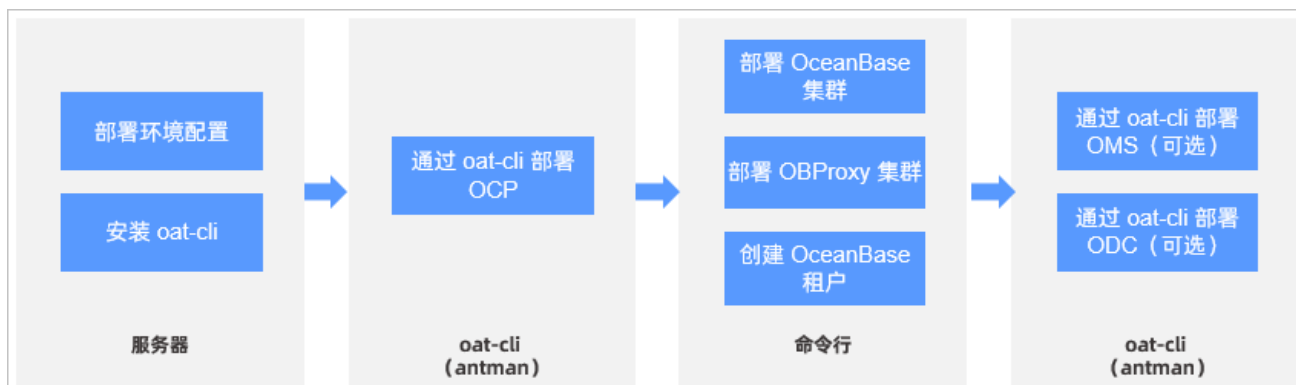
OceanBase 数据库支持部署在 x86\_64 以及 ARM\_64 架构的物理服务器和主流的虚拟机，其操作系统支持主流的 Linux 发行版本。

本手册主要介绍如何部署 OceanBase 云平台（OceanBase Cloud Platform, OCP）、OceanBase 集群、OceanBase 代理服务（OBProxy）等。



## 2 部署流程

OceanBase 数据库集群的命令行部署流程如下图所示：



### 说明

Antman 是一个提供一键部署 OCP, OMS, ODC 等 OceanBase 数据库周边工具平台的 cli 工具（简称：oat-cli）。

## 3 准备服务器

OceanBase 数据库集群高可用和容灾时至少由三个节点组成，每个节点对应一个 observer 进程，不同节点上的多个 observer 进程组成一个集群对外提供服务。

### 说明

- 如果需要 OCP 管控服务提供高可用能力，则需要 3 台管控服务器进行三节点部署，并提供负载均衡软件或者硬件，如 F5、SLB，或者使用 OceanBase 提供的 NLB 软负载组件。
- 在最小化部署 OceanBase 4.0 时，需要 1 台 OBServer 服务器安装 OceanBase 集群和 1 台单独的服务器安装 OCP 用于管理 OceanBase 集群。

服务器应满足的最低配置要求如下表所示：

服务器类型	数量	功能最低配置	性能最低配置
OCP 管控服务器	1台	8C, 32GB 内存, 1.5TB 存储（包含 OAT 与 ODC 所需资源）	32C, 128GB 内存, 1.5 TB SSD 存储, 万兆网卡（包含 OAT 与 ODC 所需资源）
OceanBase 计算服务器	3台	4C, 16GB 内存, 1.2TB 存储  <b>说明</b> 有关 OceanBase 服务器最小资源参数的推荐配置信息，请参见 <a href="#">最小资源参数配置推荐</a> 。	32C, 256GB 内存, 2TB SSD 存储, 万兆网卡
OBProxy 计算服务器	3 台，可复用 OBServer 服务器	4C, 8GB 内存, 200 GB 存储	N/A
OAT 部署服务器	1 台，可复用 OCP 管控服务器	2C, 4GB 内存	N/A
（可选）OMS Docker 部署服务器	1 台	12C, 24GB 内存, 500 GB 存储	32C, 128GB 内存, 2TB 存储
（可选）ODC Docker 部署服务器	1 台，复用 OCP 管控服务器	2C, 8GB 内存	4C, 8GB 内存

## 4 服务器配置

本节主要介绍服务器的硬件设置要求，包括操作系统、BIOS 设置、磁盘挂载、网卡设置、软件要求等。

### 4.1 准备操作系统

支持在下表所示的 Linux 操作系统中安装 OceanBase 数据库。

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（鲲鹏、飞腾）
龙蜥 AnolisOS	8.6 及以上	x86_64（包括海光），ARM_64（鲲鹏、飞腾）
KylinOS	V10	x86_64（包括海光），ARM_64（鲲鹏、飞腾）
统信 UOS	V20	x86_64（包括海光），ARM_64（鲲鹏、飞腾）
中科方德 NFSCChina	4.0 及以上	x86_64（包括海光），ARM_64（鲲鹏、飞腾）
浪潮 Inspur kos	5.8	x86_64（包括海光），ARM_64（鲲鹏、飞腾）
CentOS / Red Hat Enterprise Linux	7.2 及以上  <b>说明</b> 暂不支持 CentOS 8.X。	x86_64（包括海光），ARM_64（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	8.3 及以上	x86_64（包括海光），ARM_64（仅支持鲲鹏、飞腾）

#### 说明

操作系统需要配置网络 and 软件管理器（yum 或 zypper 源）。

### 4.2 BIOS 设置

特殊设置：

- Numa 关闭（INTEL）

**说明**

INTEL 环境建议修改配置文件 `/etc/sysctl.conf` 将参数 `vm.swapness` 设置为 `0`，方法如下：

```
[root@xxx /]
$vi /etc/sysctl.conf
vm.swapness = 0

[root@xxx /]
$sysctl -p #应用 sysctl.conf 配置文件里的值
```

- Numa 开启（海光/AMD/鲲鹏/ARM)

**说明**

ARM 和海光环境建议修改配置文件 `/etc/sysctl.conf` 将参数 `kernel.numa_balancing`、`vm.zone_reclaim_mode` 和 `vm.swapness` 设置为 `0`，方法如下：

```
[root@xxx /]
$vi /etc/sysctl.conf
kernel.numa_balancing = 0
vm.zone_reclaim_mode = 0
vm.swapness = 0

[root@xxx /]
$sysctl -p #应用 sysctl.conf 配置文件里的值
```

**BIOS 需要关闭以下选项：**

- Cstate
- Pstate
- EIST
- Power saving
- Turbo Mode

**BIOS 需要配置以下选项：**

- Automatic Power on After Power Loss: Always on
- Intel Virtualization Technology: 开启
- Hyper-threading: 开启
- Hardware prefetcher: 开启
- VT-d: 开启
- SR-IOV: 开启
- Energy performance: 开启最大 performance

**说明**

不同的服务器修改 BIOS 的方式不同，具体操作请参见服务器操作手册。

## 4.3 磁盘挂载

- OCP 服务器的磁盘挂载点要求如下表所示。

挂载点	大小	用途	磁盘格式
/home	100 GB~300 GB	各组件运行日志盘	建议 ext4
/data/log1	内存大小的 3~4 倍	OCP 元数据库日志盘	建议 ext4
/data/1	取决于所需存储的数据大小	OCP 元数据库数据盘	建议 ext4，大于 16 TB 使用 xfs
/docker	200 GB~500 GB	Docker 根目录	建议 ext4

- OBCServer 服务器的磁盘挂载点要求如下表所示。

挂载点	大小	用途	文件系统格式
/home	100 GB~300 GB	OBCServer 运行日志盘	建议 ext4
/data/log1	内存大小的 3~4 倍	OBCServer 日志盘	建议 ext4
/data/1	取决于所需存储的数据大小	OBCServer 数据盘	建议 ext4，大于 16 TB 使用 xfs

- OMS 服务器的磁盘挂载点要求如下表所示。

挂载点	大小	用途	文件系统格式
/data/oms	400 GB，取决于同步对象的大小、同步作业个数、同步对象 DML 操作的频率和 Store 保留的时间	OMS Store 组件数据盘	建议 ext4

#### 说明

建议根目录不小于 50 GB，不建议单独挂载 /opt 和 /var 目录。如果使用 LVM，建议创建时使用条带化参数。示例如下：

```
lvcreate -n data -L 3000G obvg --stripes=3 --stripesize=128
```

## 4.4 网卡设置

建议配置 2 块万兆网卡。

- bond 模式取名 bond0，mode1 或 mode4，推荐使用 mode4。对于 mode4 模式，交换机需要配置 802.3ad。
- 网卡名建议使用 eth0、eth1。
- 建议使用 network 服务，不建议使用 NetworkManager。

## 4.5 软件要求

在部署 OAT 图形化界面、ODC 与 OMS 的机器需要安装 Docker，版本要求如下表所示。

软件	版本
Docker	Docker CE 17.03 及以上

## 5 准备安装包

部署前，请确认已获取以下安装包，用于后续的安装部署。

各组件安装包推荐使用的镜像如下表所示：

组件	说明	安装包说明
OAT 镜像包	用于安装和管理 OCP、OMS 和 ODC 等。	请联系技术支持获取最新 OAT 镜像包。
OCP docker image	用于部署 OCP。	请联系技术支持获取最新 OCP Docker 镜像。
MetaDB docker image	用于部署 OCP MetaDB、MonitorDB。	请联系技术支持获取最新 MetaDB 镜像。
OceanBase RPM 包	用于部署 OceanBase 集群。	请联系技术支持获取最新 OceanBase 数据库安装包。
OBProxy RPM 包	用于部署代理服务。	请联系技术支持获取最新 OBProxy 安装包。
（可选）OMS docker image	用于部署 OceanBase 迁移服务。	请联系技术支持获取最新 OMS docker 镜像。
（可选）ODC docker image	用户部署 ODC 服务。	请联系技术支持获取最新 ODC docker 镜像。

### 5.0.0.1 说明

在 OceanBase 官方网站左上角 **资源** 下的 [下载中心](#) 中提供各组件部分版本的安装包下载。请联系技术支持获取 **下载中心** 中没有的安装包。

## 6 OceanBase 集群高可用方案简介

OceanBase 数据库采用基于无共享（Shared-Nothing）的多副本架构，让整个系统没有任何单点故障，保证系统的持续可用。OceanBase 支持单机、机房、城市级别的高可用和容灾，可以进行单机房、双机房、两地三中心、三地五中心部署。

### 6.1 同城三机房三副本部署

#### 6.1.1 特点

- 同城 3 个机房组成一个集群（每个机房是一个 Zone），机房间网络延迟一般在 0.5 ~ 2 ms 之间。
- 机房级灾难时，剩余的两份副本依然是多数派，依然可以同步 Redo-Log 日志，保证 RPO=0。
- 无法应对城市级的灾难。

#### 6.1.2 部署方案示意图



### 6.2 三地五中心五副本部署

#### 6.2.3 特点

- 三个城市，组成一个 5 副本的集群。



- 任何一个 IDC 或者城市的故障，依然构成多数派，可以确保 RPO=0。
- 由于 3 份以上副本才能构成多数派，但每个城市最多只有 2 份副本，为降低时延，城市 1 和城市 2 应该离得较近，以降低同步 Redo-Log 的时延。
- 为降低成本，城市 3 可以只部署日志型副本（只有日志）。

## 6.2.4 部署方案示意图



### 说明

同城三机房或者三地五中心的方案对基础设施要求太高。为了利用企业现网的基础设施，OceanBase 提供了同城两机房和两地三中心两种方案。

## 7 安装 oat-cli

oat-cli 是 OceanBase 数据库生态产品的自动化部署工具。部署前，需要在所有服务器上安装 oat-cli 的 RPM 包，用于自动化部署。

### 7.1 背景信息

OceanBase 独立版（Antman）是一个提供一键部署 OCP，OMS，ODC 等 OceanBase 数据库周边工具平台的 cli 工具（简称：oat-cli），并且为 OBServer 提供标准化操作系统配置的能力，支持部署在 x86\_64 以及 aarch64 架构的物理服务器和主流的虚拟机，操作系统支持主流的 Linux 发行版本。

### 7.2 操作步骤

1. 将 oat-cli 的安装包拷贝到任意一台服务器。

#### 说明

请联系技术支持人员获取 Antman 安装包。

2. 使用 root 用户登录该机器，执行以下命令，安装 oat-cli。

有关最新版本 oat-cli 安装包的获取方式，请联系 OceanBase 技术支持。

- x86\_64平台：

```
[root@hostname /]# rpm -ivh t-oceanbase-antman-x.x.x-xxxxxxx.aliyos7.x86_64.rpm
```

- ARM\_64平台：

```
[root@hostname /]# rpm -ivh t-oceanbase-antman-x.x.x-xxxxxxx.aliyos7.aarch64.rpm
```

3. 重复步骤 1 ~ 步骤 2，完成所有服务器上 oat-cli 的安装。

## 8 通过 oat-cli 配置部署环境

您需要使用 oat-cli 修改各服务器的部署环境配置。

### 8.1 添加 admin 用户

为便于后续维护，部署前，您需要为每台服务器添加 admin 用户。

```
[root@hostname /]# cd /root/t-oceanbase-antman/clonescripts  
[root@hostname clonescripts]# ./clone.sh -u
```

### 8.2 修改操作系统内核配置

#### 8.2.1 OCP 服务器

执行以下命令，对每台 OCP 服务器进行设置。

```
[root@hostname /]# cd /root/t-oceanbase-antman/clonescripts  
[root@hostname clonescripts]# ./clone.sh -c -r ocp
```

#### 8.2.2 OBServer 服务器

执行以下命令，对每台 OBServer 服务器进行设置。

```
[root@hostname /]# cd /root/t-oceanbase-antman/clonescripts  
[root@hostname clonescripts]# ./clone.sh -c -r ob
```

### 8.3 安装依赖包

#### 8.3.3 OCP 服务器安装依赖包

在每台 OCP 服务器执行以下命令，安装依赖包。

```
[root@hostname /]# cd /root/t-oceanbase-antman/clonescripts  
[root@hostname clonescripts]# ./clone.sh -m -r ocp
```

#### 8.3.4 OBServer 服务器安装依赖包

在每台 OBServer 服务器执行以下命令，安装依赖包。

```
[root@hostname /]# cd /root/t-oceanbase-antman/clonescripts  
[root@hostname clonescripts]# ./clone.sh -m -r ob
```

## 8.4 OCP 与 OMS 服务器安装 Docker

在 OCP 与 OMS 服务器上执行以下命令，安装 Docker。

```
[root@hostname /]# cd /root/t-oceanbase-antman/clonescripts  
[root@hostname clonescripts]# ./clone.sh -i
```

## 9 配置时钟源

OceanBase 相关产品之间，在时间上都有一些隐含的联系，因此要求 OceanBase 产品的所有服务器的物理时间要保持同步，误差尽量控制在 100 ms 以内，甚至更小。因此集群的多个节点以及 OCP 节点的时钟必须配置时钟同步服务 NTP 或者 chrony，保证所有节点的时钟偏差在 100 ms 以内。本文介绍如何为集群内的多台机器配置 NTP 时钟源。如果您在单机安装 OceanBase 数据库，则无需设置。

### 说明

OceanBase 集群中的服务器时间必须保持一致，否则会导致 OceanBase 集群无法启动，运行时也会出现故障。物理机与时钟服务器的误差在 50 ms 以下可认为时钟是同步状态，OceanBase 集群最大容忍误差不能超过 100 ms。当超过 100 ms 时，会出现无主情况。恢复时钟同步后。重启 OceanBase 集群，可以恢复正常。如果您已配置 NTP 时钟同步，则无需重新配置。

### 9.1 前提条件

配置 NTP 时钟源前，确保您拥有所有机器的 root 用户权限。

### 9.2 操作步骤

按以下步骤配置 NTP 时钟同步。

#### 说明

通常使用 Linux 自带的 NTP 服务可以同步时间。如果当前机器环境有稳定可靠的 NTP 服务器，则选它作为所有服务器的 NTP 源。如果没有，则选固定的一台服务器，把它作为 NTP 源。

1. 在每台机器上通过 YUM 包管理器安装 NTP。

```
[root@xxx /]# yum install ntp ntpdate -y
```

2. 配置 ntp.conf 文件。

```
[root@xxx /]# vi /etc/ntp.conf  
restrict default ignore  
restrict 127.0.0.1  
restrict 192.168.0.0 mask 255.255.0.0
```

```
driftfile /var/lib/ntp/drift
pidfile /var/run/ntpd.pid
#logfile /var/log/ntp.log

# local clock
server 127.127.1.0
fudge 127.127.1.0 stratum 10

server 192.168.XX.XX iburst minpoll 4 maxpoll 6
```

参数解释：

- `restrict` 用于指定 IP 的相关时间同步命令权限。

```
restrict [IP] mask [netmask_IP] [parameter]
```

`parameter` 包括 `ignore`、`nomodify`、`noquery`、`notrap` 和 `notrust` 等。  
`ignore` 默认指拒绝所有类型的 NTP 同步。

- `server` 用于指定上层 NTP 源服务器。

```
server [IP or hostname] [prefer]
```

如果没有上层 NTP 源服务器，可以设置为 127.127.1.0，即本机作为 NTP 源服务器。

### 3. 重启 NTP 同步服务。

```
[root@xxx /]# systemctl restart ntpd
[root@xxx /]# systemctl status ntpd
```

### 4. 检查 NTP 同步状态。

#### 说明

如果 NTP 服务刚启动时，需要等一会再执行 `ntpstat` 命令。

```
[root@xxx /]# ntpstat
[root@xxx /]# ntpq
remote refid st t when poll reach delay offset jitter
```

```
=====
==
LOCAL(0) .LOCL. 10 l 589 64 0 0.000 0.000 0.000
*192.168.XX.XX 10.10.10.1 2 u 18 64 377 1.591 0.249 0.054
```

命令 ntpq 结果参数释义：

- remote：表示使用的 NTP 服务器，\* 表示目前选择的 NTP 服务器。LOCAL 表示本机。  
x 表示已不再使用。- 表示已不再使用。+ 良好的优先考虑的。# 良好的但未使用的。
- refid：远程 NTP 服务器使用的更高一级 NTP 服务器。INIT 表示还在获取。
- st：远程 NTP 服务器的 Stratum（级别）。
- when：最后一次同步到现在的时间（默认为秒，h 表示小时，d 表示天）。
- poll：同步的频率，单位：秒。
- delay：从本机到远程 NTP 服务器的往返时间（单位毫秒）。
- offset：本机与远程 NTP 服务器的时间偏移量（单位毫秒）。
- jitter：本机与远程 NTP 服务器的时间偏移量平均偏差（单位毫秒）。

## 5. 最终检查方法。

最终同步延时检查以下面为准：

```
[root@xxx /]# clockdiff 192.168.XX.XX
.....
host=192.168.XX.XX rtt=1(0)ms/1ms delta=0ms/0ms Sat Apr 18 14:41:40 2020
```

## 6. （可选）手动同步方法。

如果时间总是有很大误差，可以使用 ntpdate 命令手动订正时间。这个跟自动同步服务是冲突的，需要先停掉 NTP 服务。

```
[root@xxx /]# systemctl stop ntpd
[root@xxx /]# ntpdate -u 192.168.XX.XX
18 Apr 14:54:20 ntpdate[108001]: adjust time server 192.168.XX.XX offset
-0.000180 sec
```

如果这个方法有效，就配置到系统的 crontab 下。

```
[root@xxx /]# crontab -e
***** /sbin/ntpdate -u 192.168.XX.XX 2>&1 1>>/tmp/ntpupdate.log
```

#### 7. （可选）防火墙规则。

标准化的环境是要关闭防火墙的，如果服务器环境非要开启防火墙，可以参照下面规则操作。

```
[root@xxx /]# iptables --A INPUT --p udp --i eth0 --s 192.168.1.0/16 --dport 123 --
j ACCEPT
[root@xxx /]# systemctl restart ntpd
```



## 10 上传安装包

在使用 oat-cli 命令行工具部署 OCP、OMS 或 ODC 前，您需要向安装 oat-cli 的服务器上传各产品所需的安装包。

通过 `scp` 命令将本地的安装包上传到安装 oat-cli 的服务器的 `/root/t-oceanbase-antman/` 目录下。

有关各安装包的具体版本，请联系技术支持获取您所需的版本。

- 对于 OCP 部署，您需要向安装 oat-cli 的服务器中上传 OCP、OCP MetaDB 的镜像。

其中 `oat_server` 表示部署 oat-cli 的服务器的 IP 地址。

### 10.0.0.1 说明

如果您希望在三节点 OCP 部署中使用 DNS 模式，您还需要准备 DNS 的镜像。

```
[root@hostname /]# scp ./ocp.tar <oat_server>:/root/t-oceanbase-antman/ocp.tar
[root@hostname /]# scp ./metadb.tar <oat_server>:/root/t-oceanbase-antman
/observer.tar
[root@hostname /]# scp ./dns.tar <oat_server>:/root/t-oceanbase-antman/dns.tar
```

- 对于 ODC 部署，您需要向安装 oat-cli 的服务器中上传 ODC 的镜像：

其中 `oat_server` 表示部署 oat-cli 的服务器的 IP 地址。

```
[root@hostname /]# scp ./odc.tar <oat_server>:/root/t-oceanbase-antman/odc.tar
```

- 对于 OMS 部署，您需要向安装 oat-cli 的服务器中上传 OMS 的镜像：

其中 `oat_server` 表示部署 oat-cli 的服务器的 IP 地址。

```
[root@hostname /]# scp ./oms.tar <oat_server>:/root/t-oceanbase-antman/oms.
tar
```

# 11 配置主机信息

本节主要介绍主机名称标准化。

## 11.1 操作步骤

在所有服务器上通过以下步骤配置主机信息：

1. 修改 `etc/hosts`，指定当前服务器的 IP 地址和主机名。

```
[root@xxx /]# vi /etc/hosts
10.10.10.1 observer1
```

### 注意

IP 做了脱敏处理，应根据本机情况填写。

2. 修改 `etc/sysconfig/network`，指定 `HOSTNAME=observer1`。

```
[root@xxx /]# vi /etc/sysconfig/network
HOSTNAME=observer1
```

3. 修改 `/etc/hostname`。

```
[root@xxx /]# vi /etc/hostname
observer1
```

4. 执行以下命令，确保可以正确获取当前主机的 IP 地址。

```
[root@xxx /]# hostname -i
10.10.10.1
```

## 12 部署前环境检查

服务器和操作系统参数设置完成后，为确保部署的正常进行，需要验证 OBCS 和 OCP 服务器是否满足了标准安装条件。

### 12.1 OCP 服务器

在 OCP 所有服务器上，执行以下命令：

```
cd /root/t-oceanbase-antman/clonescripts
sh precheck.sh -m ocp
```

执行后，如果有显示 **FAIL** 的信息，根据错误提示进行修改即可。

### 12.2 OBCS 服务器

在 OBCS 所有服务器上，执行以下命令：

```
cd /root/t-oceanbase-antman/clonescripts
sh precheck.sh -m ob
```

执行后，如果有显示 **FAIL** 的信息，根据错误提示进行修改即可。

### 12.3 备份与恢复服务器

在备份与恢复所有服务器上，执行以下命令：

```
[root@hostname /]# cd /root/t-oceanbase-antman/clonescripts
[root@hostname clonescripts]# sh precheck.sh -m obbackup
```

### 12.4 OMS

在 OMS 所有服务器上，执行以下命令：

```
[root@hostname /]# cd /root/t-oceanbase-antman/clonescripts
[root@hostname clonescripts]# sh precheck.sh -m oms
```

## 13 部署说明

OCP 产品以 Docker 形态运行，对服务器结构和 Linux 操作系统的要求与 OceanBase 数据库服务器一致。

### 13.1 描述

OceanBase 云平台（OceanBase Cloud Platform，简称 OCP）伴随 OceanBase 数据库而生，是一款以 OceanBase 数据库为核心的企业级数据库管理平台。通过 OCP，您可以一键安装、升级、扩容、卸载 OceanBase 数据库集群，创建和管理运维任务，监控集群的运行状态，并查看告警。

OCP 当前支持 OceanBase 数据库的所有主流版本，不仅提供对 OceanBase 集群和租户等组件的全生命周期管理服务，同时也对 OceanBase 数据库相关的资源（主机、网络 and 软件包等）提供管理服务，让您能够更加高效地管理 OceanBase 集群，降低企业的 IT 运维成本。

对于 OCP 的部署，目前支持单节点、三节点和多 AZ 部署模式。

### 13.2 单节点部署

通过单个节点提供全部 OCP 能力。单节点部署的负载均衡模式为 none。

### 13.3 三节点部署

OCP 支持通过三节点部署来实现高可用。

三节点部署可以选择 DNS 负载均衡或者其他外部负载均衡设备，例如 F5 等。

### 13.4 多 AZ 部署

可用区（Available-Zone，Azone）指由承载 OceanBase 业务的一个或多个机房映射出的逻辑区域。

OCP 的多 AZ（Multi Available-Zone，多可用区）模式是指将同一个 OCP 集群分别部署到多个可用区中，并优先以可用区为边界，来限定 OCP、OceanBase 和 OBProxy 之间的访问服务链路。各可用区内承载的业务会优先由其本区域内的 OCP 管理。

## 14 生成配置文件

在部署 OCP 前，您需要使用 `oat-cli` 命令行工具生成配置文件模板，并根据实际信息修改模板。

### 14.1 配置环境变量

#### 14.1.1 描述

通过指定环境变量的方式执行 SSH 连接，支持非 root 用户、私钥方式，用法参见 `install.sh -h` 帮助信息（重要变更，不再打通免密，提高安全性）。

#### 14.1.2 适用版本

`oat-cli` 1.4.0 及以后版本。

#### 14.1.3 使用 `init_obcluster_conf.sh` 生成配置文件

如果存在配置文件，脚本将从环境变量中读取 `SSH_AUTH`、`SSH_USER`、`SSH_PORT`、`SSH_PASSWORD` 和 `SSH_KEY_FILE`。

设置参数语法：

```
export ssh_parameter = parameter_value
```

ssh\_parameter:

`SSH_AUTH`

| `SSH_USER`

| `SSH_PORT`

| `SSH_PASSWORD`

| `SSH_KEY_FILE`

参数解释：

- `SSH_AUTH`：设置私钥方式，默认值 `"password"`。可以设置为 `"password"` 或 `"pubkey"`。

- SSH\_USER: 设置用户, 默认值 root。如果不是 root 用户, 请确保该用户拥有执行 `sudo bash` 权限, 没有密码和 `/etc/sudoers` 的设置。
- SSH\_PORT: 设置端口, 默认值 22。
- SSH\_PASSWORD: 设置密码, 默认为空。
- SSH\_KEY\_FILE: 设置密钥文件, 默认值 `/root/.ssh/id_rsa`。仅在 `SSH_AUTH = pubkey` 时生效。

示例:

```
export SSH_AUTH = password
export SSH_USER = root
export SSH_PORT = 22
export SSH_PASSWORD = '*****'
export SSH_KEY_FILE = /root/.ssh/id_rsa
```

## 14.2 单节点部署

1. 以 root 用户登录用于部署 OCP 的服务器, 执行以下命令。

```
[root@hostname /]# cd /root/t-oceanbase-antman
[root@hostname /root/t-oceanbase-antman]# bash init_obcluster_conf.sh
```

2. 在显示的模式选择中, 输入 1, 表示单节点部署。

系统会生成对应的配置模板。

示例如下:

```
## obcluster.conf
##
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=xxx.xxx.xxx.xxx
```

```
OBSERVER01_ROOTPASS=rootpass
OBSERVER01_ADMINPASS=adminpass
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

#剩余内容
...
```

### 3. 根据注释信息，修改配置模板。

需要修改的信息如下：

- 机器 IP、root 密码和 admin 密码必须填写，单节点部署只需要填写 1 台机器。
- 对于单节点部署场景，负载均衡 `lb_mode` 选择 none 模式，表示不使用负载均衡。
- 根据物理机的 CPU 和内存资源编排 Docker 容器资源。
- 可使用命令 `docker load -i {docker image 包}` 和命令 `docker images` 获取 OCP 各组件 Docker 容器的版本信息，包括 OBServer、OBProxy 和 OCP。
- 如果准备部署 OMS，请填写 OMS 基本配置，包括部署 OMS 机器的 IP 和 OMS Docker 容器的版本信息。
- 如果准备部署 ODC，请填写 ODC Docker 容器的版本信息。

## 14.3 三节点部署

### 1. 以 root 用户登录 OCP 服务器，执行以下命令。

```
bash init_obcluster_conf.sh
```

### 2. 在显示的模式选择中，输入 3，表示三节点部署。

系统会生成对应的配置模板。

示例如下：

```
##### 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=xxx.xxx.xxx.xxx
ZONE2_RS_IP=xxx.xxx.xxx.xxx
ZONE3_RS_IP=xxx.xxx.xxx.xxx
OBSERVER01_ROOTPASS=rootpass
OBSERVER02_ROOTPASS=rootpass
OBSERVER03_ROOTPASS=rootpass
OBSERVER01_ADMINPASS=adminpass
OBSERVER02_ADMINPASS=adminpass
OBSERVER03_ADMINPASS=adminpass

##### 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=dns

# 剩余内容
...
```

### 3. 根据注释信息，修改配置模板。

需要修改的信息如下：

- 机器 IP、root 密码和 admin 密码必须填写，三节点部署需要填写 3 台机器。
  - 如果负载均衡模式选择 dns，默认会有一个推荐域名 `oceanbase.com`，该域名仅在部署的 OCP 的节点可以自由访问，其他节点访问需要配置域名解析。  
本文档以负载均衡 `lb_mode` 选择 DNS 为例提供操作指导。
  - 如果负载均衡模式选择 F5，则需要配置准备好的 `obproxy vip`、`vport` 和 `ocp vip`、`vport`，需要预先申请。



- 请根据物理机的 CPU 和内存资源编排 Docker 容器资源。  
默认规格：OBServer 32C/128G；OCP 8C/16G；OBProxy 4C/12G。  
若物理机资源为推荐最低配置，可做自定义调整，OBServer不低于 24C/100G，OCP 不低于 4C/8G，OBProxy不低于 2C/10G。
- 可使用命令 `docker load -i {docker image 包}` 和命令 `docker images` 获取 OCP 各组件 Docker 容器的版本信息，包括 OBServer、OBProxy 和 OCP。
- 如果准备部署 OMS，请填写 OMS 基本配置，包括部署 OMS 机器的 IP 和 OMS docker 容器的版本信息。
- 如果准备部署 ODC，请填写 ODC docker 容器的版本信息。

## 14.4 多 AZ 模式部署

1. 检查 `/root/t-oceanbase-antman/` 目录下是否有 `obcluster.conf` 文件。
  - 有，则直接跳到步骤 4。
  - 无，则顺序向下执行。
2. 以 root 用户登录用于部署 OCP 的服务器，执行以下命令。

```
[root@hostname /]# cd /root/t-oceanbase-antman/  
[root@hostname clonescripts]# bash init_obcluster_conf.sh
```

3. 在显示的模式选择中，输入 3，表示多节点部署。

系统会生成对应的配置模板。

示例如下：

```
##### OCP 多 AZ 配置 #####  
OCP_MAZ_ENABLED=TRUE # 如果开启多AZ，且没有全局vip时，上面配置的 OCP_VIP  
和 OBPROXY_VIP 可改成主az的 vip 或者 任意ocp节点的ip  
OCP_MAZ_AZ_NAME_LIST=(az1 az1 az2) # 对应 ocp1, ocp2, ocp3配置的 az_name  
OCP_MAZ_OCP_VIP_LIST=(192.168.0.1 192.168.0.2 192.168.0.3) # 对应 ocp1,  
ocp2, ocp3配置的 site-url 的ip  
OCP_MAZ_OCP_VPORT_LIST=(8080 8080 8080) # 对应 ocp1, ocp2, ocp3配置的  
site-url 的port  
OCP_MAZ_METAPROXY_VIP_LIST=(192.168.0.1 192.168.0.2 192.168.0.3) # 对应  
ocp1, ocp2, ocp3使用的metadb 连接ip
```

```
OCP_MAZ_METAPROXY_VPORT_LIST=(2883 2883 2883) # 对应 ocp1, ocp2, ocp3使用的metadb 连接port
```

#### 4. 部署多 AZ 模式的 OCP 时，需配置"OCP 多 AZ 配置"段落的内容。

需要修改的信息如下：

##### 说明

模板中默认多 AZ 是 3 个，如实际规划部署的 AZ 不为 3，可自行增减参数项中值的个数。如实际规划部署 2 个可用区，则可配置 OCP\_MAZ\_AZ\_NAME\_LIST=(az1 az2)。

- OCP\_MAZ\_ENABLED：是否启用多 AZ 模式，默认为 FALSE。启用时配置为 TRUE。
- OCP\_MAZ\_AZ\_NAME\_LIST：分别指定三个 OCP 节点的 AZ 名称，例如 "(az1 az2 az3)"。不同 OCP 节点的 AZ 名称可以相同，例如"(az1 az1 az2)"。
- OCP\_MAZ\_OCP\_VIP\_LIST：分别指定三个 OCP 节点的服务地址，即 SiteUrl 的主机部分，例如"(10.1.1.1 10.1.1.2 10.1.1.3)"。
- OCP\_MAZ\_OCP\_VPORT\_LIST：分别指定三个 OCP 节点的服务端口，即 SiteUrl 的端口部分，例如"(80 80 80)"。
- OCP\_MAZ\_METAPROXY\_VIP\_LIST：分别指定三个 OCP 节点访问 Metadb 的 IP 地址。
- OCP\_MAZ\_METAPROXY\_VPORT\_LIST：分别指定三个 OCP 节点访问 MetaDB 的端口。

# 15 开始部署

在生成配置完成后，您可以通过 oat-cli 部署 OCP。

## 说明

对于三节点部署，在一台服务器上执行安装命令后，oat-cli 会根据上一步生成配置文件中提供的 IP 地址与连接信息，自动将 OCP 安装到三台服务器上。

## 15.1 安装

1. 在 OCP 服务器节点上，执行以下命令，获取脚本帮助。

```
cd /root/t-oceanbase-antman
./install.sh -h
Usage: ./install.sh [OPTIONS]

Options:
-h, --help Print help and exit
-d, --debug Print debug information
-V, --version Print version
-i, --install-steps string For example 1,3-5,7-
-c, --clear-steps string For example 1,3-5,7-
-f, --config-file string Read in a config file
-l, --load-balance dns | haproxy

Steps:
1. ssh authorization
2. install load balancer (default: dns)
3. install ob server
4. init ocp metadb
5. install temp OCP
6. install obproxy
```

7. install OCP
8. POSTCHECK
9. install OMS (optional)
10. install ODC (optional)

根据返回结果，确认 OCP 的安装步骤为 1~8。

2. 执行以下命令，安装 OCP。

```
# n-m 执行第 n 步到第 m 步，-c 清除步骤，-s 安装步骤 -f 指定使用文件
bash install.sh -i 1-8
```

### 说明

在一台服务器上执行安装命令后，安装脚本会根据配置文件中提供的 IP 地址与连接信息，自动将 OCP 安装到三台服务器上。

## 15.2 异常处理

如果某一步部署失败了，只需要清除当前失败的步骤，根据日志的报错内容修复后，从当前步骤继续安装。

例如，第 5 步安装失败了，查看日志，由于配置文件配置错误，只需要修改正确后，执行以下命令：

```
bash install.sh -c 5 -i 5-
```

如果日志中返回如下信息，则表示 OCP 的部署即已完成。

```
[2020-02-11 22:53:20.978315] INFO [xxx.xxx.xxx.xxx post_check_ocp_service start]
[2020-02-11 22:53:20.980821] INFO [Curl xxx.xxx.xxx.xxx -> http://xxx.xxx.xxx.xxx:
8080/services?Action=GetObProxyConfig&User_ID=admin&UID=alibaba]
[2020-02-11 22:53:21.194190] INFO [xxx.xxx.xxx.xxx: post_check_ocp_service done]
[2020-02-11 22:53:21.201536] INFO [step8: post check done]
```

## 16 部署后检查

OCP 部署完成后，您可以登录 OCP 进行部署后检查。

不同 OCP 版本的操作界面可能不同，请以实际界面为准。本节以 OCP V3.3.2 版本为例提供操作指导。

### 16.1 访问 OCP

1. 在浏览器中输入 OCP 的访问地址，按回车键。

访问地址为：`http://< OCP 服务器的 IP 地址 >:8080`

2. 在弹出的登录页面，使用默认账号 `admin` 账号登录。

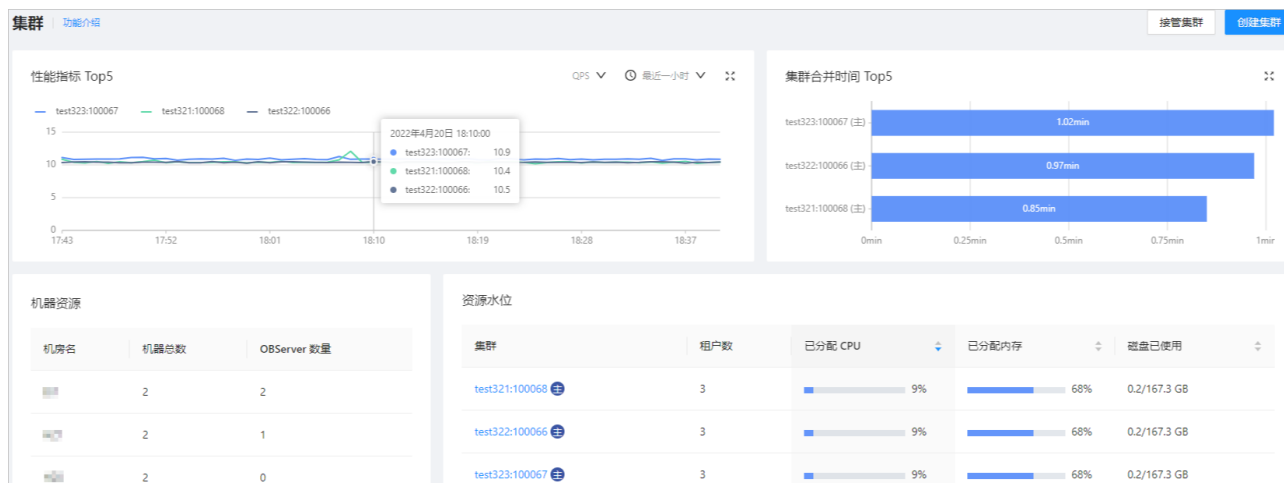


#### 说明

- 如果需要通过域名访问，则需要配置本机的域名解析服务。
- 请联系 OceanBase 售后支持获取 admin 账号的默认密码。为保证账号安全，首次登录后请及时修改密码。

### 16.2 检查 MetaDB 集群信息

1. 在左侧导航栏上，单击 **集群**。
2. 在 **集群** 页，查看性能指标，机器资源以及集群列表等是否展示正常。

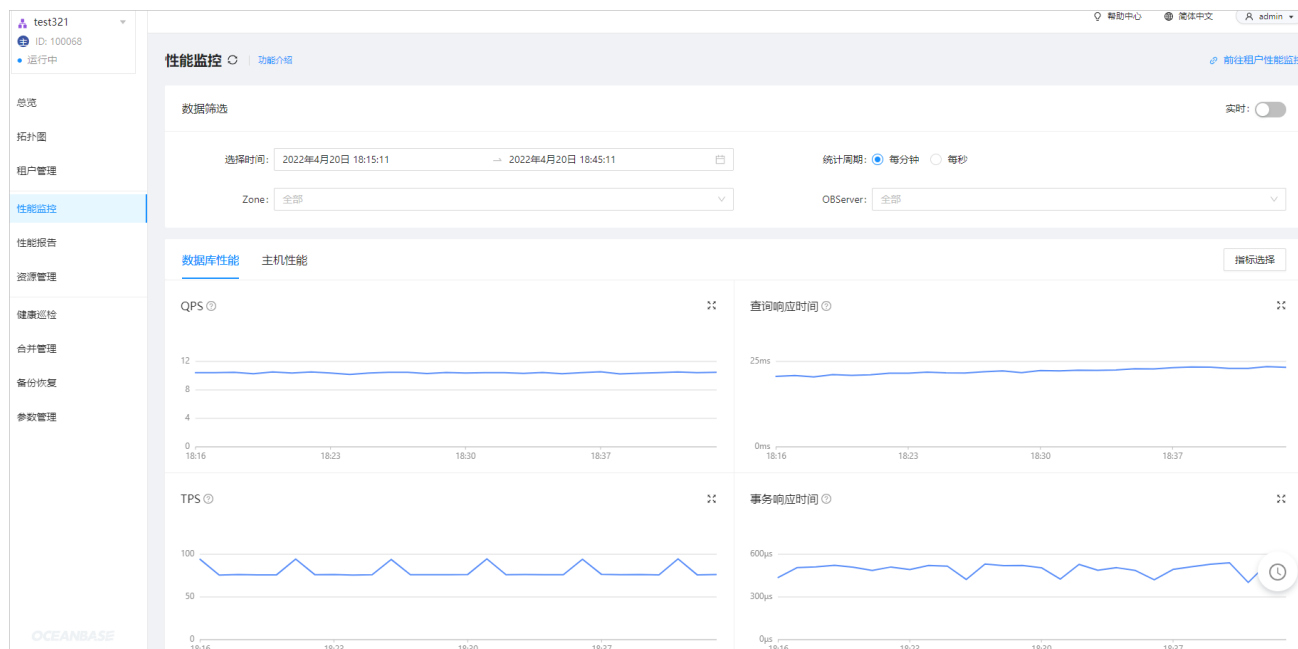


3. 在 **集群** 页的 **集群列表** 中，找到 MetaDB 的集群并单击集群名，进入集群 **总览** 页，查看 MetaDB 数据库的基本信息、资源水位、Zone 列表、OBServer 列表是否展示均正确。



## 16.3 检查监控状态

在 **集群** 所在的左侧导航栏中，单击 **性能监控**，进入集群的 **性能监控** 页面，查看数据库性能以及主机性能曲线是否显示正常。



## 17 部署单副本 OceanBase 集群

OceanBase 也支持单副本部署，并且单副本 OceanBase 集群还可以扩容（增加节点），所以也称为集群。OceanBase 是单进程软件，只需要安装 `oceanbase-xxx.rpm` 包和 `obproxy-xxx.rpm` 包。本节主要介绍通过使用 RPM 包安装单副本 OceanBase 集群。

### 17.1 前提条件

在安装 OceanBase 数据库之前，需要确认以下信息：

- OBServer 服务器已完成相关配置。详细信息，请参见 [部署前准备](#) 中 [服务器配置](#) 章节。
- 部署环境已完成相关配置。详细信息，请参见 [命令行部署](#) 中 [部署环境配置](#) 下的章节。

### 17.2 操作步骤

#### 17.2.1 安装目录和软件

1. 安装 OceanBase 数据库 RPM 包。

其中 `$rpm_dir` 表示存放 RPM 包的目录，`$rpm_name` 表示 RPM 包的名称。

```
[root@xxx /]# cd $rpm_dir
[root@xxx $rpm_dir]# rpm -ivh $rpm_name
```

##### 说明

OceanBase 数据库软件会安装在目录 `/home/admin/oceanbase` 下。

示例如下：

```
[root@xxx admin]# rpm -ivh oceanbase-*-20220418212020.el7.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
1:oceanbase-*-20220418212020.#####
[100%]
```

2. 清理 OceanBase 目录（第一次部署不用）。

当要清理之前的 OceanBase 环境，或者 OceanBase 安装部署过程中出现问题导致环境变乱或者产生文件影响下一次安装时，选择直接清理旧的 OceanBase 目录。其中 `$cluster_name` 为集群名。



```
[root@xxx admin]# su - admin
-bash-4.2$ kill -9 `pidof observer`
-bash-4.2$ rm -rf /data/1/$cluster_name
-bash-4.2$ rm -rf /data/log1/$cluster_name
-bash-4.2$ rm -rf /home/admin/oceanbase/store/$cluster_name /home/admin
/oceanbase/log/* /home/admin/oceanbase/etc/*config*
-bash-4.2$ ps -ef|grep observer
```

示例如下：

```
[root@xxx admin]# su - admin
-bash-4.2$ kill -9 `pidof observer`
-bash-4.2$ rm -rf /data/1/obdemo
-bash-4.2$ rm -rf /data/log1/obdemo
-bash-4.2$ rm -rf /home/admin/oceanbase/store/obdemo /home/admin
/oceanbase/log/* /home/admin/oceanbase/etc/*config*
-bash-4.2$ ps -ef|grep observer
```

### 3. 初始化 OceanBase 目录。

OceanBase 的数据目录通常建议在独立的磁盘上，然后通过软链接方式链接到软件 Home 目录下面。其中 `$cluster_name` 为集群名。

```
[root@xxx admin]# su - admin
-bash-4.2$ mkdir -p /data/1/$cluster_name/{etc3,sort_dir,sstable,slog}
-bash-4.2$ mkdir -p /data/log1/$cluster_name/{clog,etc2,ilog,oob_clog}
-bash-4.2$ mkdir -p /home/admin/oceanbase/store/$cluster_name
-bash-4.2$ for t in {etc3,sort_dir,sstable,slog};do ln -s /data/1/$cluster_name/$t
/home/admin/oceanbase/store/$cluster_name/$t; done
-bash-4.2$ for t in {clog,etc2,ilog,oob_clog};do ln -s /data/log1/$cluster_name
/$t /home/admin/oceanbase/store/$cluster_name/$t; done
```

示例如下：

```
[root@xxx admin]# su - admin
-bash-4.2$ mkdir -p /data/1/obdemo/{etc3,sort_dir,sstable,slog}
-bash-4.2$ mkdir -p /data/log1/obdemo/{clog,etc2,ilog,oob_clog}
-bash-4.2$ mkdir -p /home/admin/oceanbase/store/obdemo
-bash-4.2$ for t in {etc3,sort_dir,sstable,slog};do ln -s /data/1/obdemo/$t /home
/admin/oceanbase/store/obdemo/$t; done
-bash-4.2$ for t in {clog,etc2,ilog,oob_clog};do ln -s /data/log1/obdemo/$t
/home/admin/oceanbase/store/obdemo/$t; done
```

### 说明

`obdemo` 是以集群名称创建的目录，可自定义，在启动进程时会用到。

检查结果：

```
-bash-4.2$ cd /home/admin/oceanbase
-bash-4.2$ tree store/
store/
|-- obdemo
|-- clog -> /data/log1/obdemo/clog
|-- etc2 -> /data/log1/obdemo/etc2
|-- etc3 -> /data/1/obdemo/etc3
|-- ilog -> /data/log1/obdemo/ilog
|-- oob_clog -> /data/log1/obdemo/oob_clog
|-- slog -> /data/1/obdemo/slog
|-- sort_dir -> /data/1/obdemo/sort_dir
`-- sstable -> /data/1/obdemo/sstable

9 directories, 0 files
```

#### 4. 测试 I/O 能力（可选）。

测试数据盘 I/O 能力，生成性能报告文件放在 `/home/admin/oceanbase/etc` 目录下。`observer` 启动时会读取这个文件进而自动设置内部一些跟 I/O 有关的参数。

示例如下：

```
-bash-4.2$ time /home/admin/oceanbase/bin/ob_admin io_bench -c /home
/admin/oceanbase/etc -d /data/1/obdemo user:root succ to open,
filename=ob_admin.log, fd=3, wf_fd=2
...
...
real 0m0.331s
user 0m0.100s
sys 0m0.262s
```

#### 5. 安装 OceanBase 客户端。

OceanBase 实例兼容 Oracle 或 MySQL，如果是 Oracle 租户，Java 程序连接需要使用 OceanBase 提供的 Java 驱动文件（oceanbase-client-\*.jar）。如果要在命令行下访问 Oracle 租户，还需要安装客户端 OBClient。

OBClient 是 OceanBase 命令行客户端，可以访问 OceanBase 的 MySQL 租户和 ORACLE 租户。

示例如下：

```
-bash-4.2$ rpm -ivh obclient-1.2.6-20210510164331.el7.alios7.x86_64.rpm

##核查是否安装成功##
-bash-4.2$ which obclient
/usr/bin/obclient
```

## 17.2.2 初始化 OceanBase 集群

### 说明

示例 IP 做了脱敏处理，这不是安装需求。在部署时应根据自己机器真实 IP 填写。

#### 1. 启动 observer 进程。

示例如下：

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
```

```
/observer -i eth0 -P 2882 -p 2881 -z zone1 -d /home/admin/oceanbase/store  
/obdemo -r '10.10.10.1:2882:2881' -c 10001 -n obdemo -o "system_memory=30G,  
datafile_size=500G,config_additional_dir=/data/1/obdemo/etc3;/data/log1  
/obdemo/etc2"  
devname: eth0  
rpc port: 2882  
mysql port: 2881  
zone: zone1  
data_dir: /home/admin/oceanbase/store/obdemo  
rs list: 10.10.10.1:2882:2881  
cluster id: 10001  
appname: obdemo  
optstr: system_memory=30G,datafile_size=500G,config_additional_dir=/data/1  
/obdemo/etc3;/data/log1/obdemo/etc2
```

参数解释：

参数	说明
-i	指定网卡名，可通过 <code>ifconfig</code> 命令查看。
-p	指定服务端口号，一般指定为 2881。
-P	指定 RPC 端口号，一般指定为 2882。
-n	指定集群名称。可自定义，不同集群名称不要重复即可。
-z	指定启动的 observer 进程所属的 Zone。
-d	指定集群主目录，除集群名字 <code>obdemo</code> 外，其他不要变动。
-c	指定集群 ID。为一组数字，可以自定义，不同集群不要重复即可。
-l	指定日志级别。
-r	指定 RS 列表，格式是 <code>ip:2882:2881</code> ，分号分割，表示 RootService 信息。三台机器参数一致。
-o	指定集群启动参数，需要根据实际情况设置。 <code>system_memory</code> 指定 OceanBase 内部保留内存，默认是 30G，机器内存比较少的时候把这个调小，影响就是可能在性能测试时有内存不足问题。 <code>datafile_size</code> 指定 OceanBase 数据文件 <code>sstable</code> 的大小（一次性初始化），根据 <code>/data/1/</code> 可用空间评估，建议不少于 100G，同时又保留一些剩余空间。如果 <code>/data/1</code> 跟 <code>/data/log1</code> 本身就是一个文件系统（共用一个盘），那么务必保证留给 <code>/data/log1</code> 的空间大小是内存的 2-4 倍。 <code>config_additional_dir</code> 指定参数文件的冗余目录。

可通过 `netstat -ntlp` 命令查看 observer 进程是否启动成功。如果监听到了 2881 与 2882 端口，则说明进程启动成功。

示例如下：

```
-bash-4.2$ netstat -ntlp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 0.0.0.0:62888 0.0.0.0:* LISTEN -
tcp 0 0 0.0.0.0:62889 0.0.0.0:* LISTEN -
tcp 0 0 127.0.0.1:9001 0.0.0.0:* LISTEN -
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN -
tcp 0 0 0.0.0.0:63000 0.0.0.0:* LISTEN -
tcp 0 0 0.0.0.0:2881 0.0.0.0:* LISTEN 79697/observer
tcp 0 0 0.0.0.0:2882 0.0.0.0:* LISTEN 79697/observer
```

## 2. 集群 bootstrap 操作。

通过 obclient 命令连接已启动的 observer 进程，密码为空。

```
[root@xxx admin]# obclient -h127.1 -uroot@sys -P2881 -p
Enter password:

obclient> SET SESSION ob_query_timeout=1000000000;
Query OK, 0 rows affected

obclient> ALTER SYSTEM BOOTSTRAP ZONE 'zone1' SERVER '10.10.10.1:2882';
Query OK, 0 rows affected
```

### 注意

如果这一步失败报错了，其原因很可能是 observer 进程启动参数有不对、observer 相关目录权限不对、日志目录空间不足一定比例（跟数据目录合用了大目录，空间被数据目录占用了）节点内存资源不足等等。请先排查这些问题点后，然后清理 OceanBase 目录从头开始（请参见 **清理 OceanBase 目录（第一次不用）**）。

## 3. 验证集群初始化成功。

进行 bootstrap 后，执行 `SHOW DATABASES;` 命令，能看到数据库列表里有 oceanbase 即可。

```
obclient> SHOW DATABASES;
```

```
+-----+  
| Database |  
+-----+  
| oceanbase |  
| information_schema |  
| mysql |  
| SYS |  
| LBACSYS |  
| ORAAUDITOR |  
| test |  
+-----+  
7 rows in set
```

#### 4. 修改密码。

sys 租户的 root 用户密码默认为空，初始化成功后请修改密码。

```
obclient> ALTER USER root IDENTIFIED BY '*****';
```

```
Query OK, 0 rows affected
```

# 18 部署三副本 OceanBase 集群

OceanBase 集群三副本，至少要有三台主机，分为三个 Zone。OceanBase 是单进程软件，只需要安装 oceanbase-xxx.rpm 包和 obproxy-xxx.rpm 包。本节主要介绍通过使用 RPM 包安装三副本 OceanBase 集群。

## 18.1 前提条件

在安装 OceanBase 数据库之前，需要确认以下信息：

- OBServer 服务器已完成相关配置。详细信息，请参见 [部署前准备](#) 中 [服务器配置](#) 章节。
- 部署环境已完成相关配置。详细信息，请参见 [命令行部署](#) 中 [部署环境配置](#) 下的章节。

## 18.2 操作步骤

### 18.2.1 安装目录和软件

1. 安装 OceanBase 数据库 RPM 包。

在各主机上分别安装 OceanBase 数据库 RPM 包。其中 `$rpm_dir` 表示存放 RPM 包的目录，`$rpm_name` 表示 RPM 包的名称。

```
[root@xxx /]# cd $rpm_dir
[root@xxx $rpm_dir]# rpm -ivh $rpm_name
```

#### 说明

OceanBase 数据库软件会安装在目录 `/home/admin/oceanbase` 下。

2. 清理旧 OceanBase 目录（第一次部署不用）。

当要清理之前的 OceanBase 环境，或者 OceanBase 安装部署过程中出现问题导致环境变乱或者产生文件影响下一次安装时，选择直接清理旧的 OceanBase 目录。各 OBServer 服务器都要操作一遍。

其中 `$cluster_name` 为集群名。

```
[root@xxx admin]# su - admin
-bash-4.2$ kill -9 `pidof observer`
-bash-4.2$ rm -rf /data/1/$cluster_name
-bash-4.2$ rm -rf /data/log1/$cluster_name
```



```
-bash-4.2$ rm -rf /home/admin/oceanbase/store/$cluster_name /home/admin
/oceanbase/log/* /home/admin/oceanbase/etc/*config*
-bash-4.2$ ps -ef|grep observer
```

示例如下：

```
[root@hostname admin]# su - admin
-bash-4.2$ kill -9 `pidof observer`
-bash-4.2$ rm -rf /data/1/obdemo
-bash-4.2$ rm -rf /data/log1/obdemo
-bash-4.2$ rm -rf /home/admin/oceanbase/store/obdemo /home/admin
/oceanbase/log/* /home/admin/oceanbase/etc/*config*
-bash-4.2$ ps -ef|grep observer
```

### 3. 初始化 OceanBase 目录。

OceanBase 的数据目录通常建议在独立的磁盘上，然后通过软链接方式链接到软件 Home 目录下面。各 OBSERVER 服务器都要操作一遍。

其中 `$cluster_name` 为集群名。

```
[root@xxx admin]# su - admin
-bash-4.2$ mkdir -p /data/1/$cluster_name/{etc3,sort_dir,sstable,slog}
-bash-4.2$ mkdir -p /data/log1/$cluster_name/{clog,etc2,ilog,oob_clog}
-bash-4.2$ mkdir -p /home/admin/oceanbase/store/$cluster_name
-bash-4.2$ for t in {etc3,sort_dir,sstable,slog};do ln -s /data/1/$cluster_name/$t
/home/admin/oceanbase/store/$cluster_name/$t; done
-bash-4.2$ for t in {clog,etc2,ilog,oob_clog};do ln -s /data/log1/$cluster_name
/$t /home/admin/oceanbase/store/$cluster_name/$t; done
```

示例如下：

```
[root@xxx admin]# su - admin
-bash-4.2$ mkdir -p /data/1/obdemo/{etc3,sort_dir,sstable,slog}
-bash-4.2$ mkdir -p /data/log1/obdemo/{clog,etc2,ilog,oob_clog}
```

```
-bash-4.2$ mkdir -p /home/admin/oceanbase/store/obdemo
-bash-4.2$ for t in {etc3,sort_dir,sstable,slog};do ln -s /data/1/obdemo/$t /home
/admin/oceanbase/store/obdemo/$t; done
-bash-4.2$ for t in {clog,etc2,ilog,oob_clog};do ln -s /data/log1/obdemo/$t
/home/admin/oceanbase/store/obdemo/$t; done
```

### 说明

`obdemo` 是以集群名称创建的目录，可自定义，在启动进程时会用到。

检查结果：

```
-bash-4.2$ cd /home/admin/oceanbase
-bash-4.2$ tree store/
store/
|-- obdemo
|-- clog -> /data/log1/obdemo/clog
|-- etc2 -> /data/log1/obdemo/etc2
|-- etc3 -> /data/1/obdemo/etc3
|-- ilog -> /data/log1/obdemo/ilog
|-- oob_clog -> /data/log1/obdemo/oob_clog
|-- slog -> /data/1/obdemo/slog
|-- sort_dir -> /data/1/obdemo/sort_dir
`-- sstable -> /data/1/obdemo/sstable

9 directories, 0 files
```

#### 4. 测试 I/O 能力（可选）。

测试数据盘 I/O 能力，生成性能报告文件放在 `/home/admin/oceanbase/etc` 目录下。`observer` 启动时会读取这个文件进而自动设置内部一些跟 I/O 有关的参数。

示例如下：

```
-bash-4.2$ time /home/admin/oceanbase/bin/ob_admin io_bench -c /home
/admin/oceanbase/etc -d /data/1/obdemo user:root succ to open,
filename=ob_admin.log, fd=3, wf_fd=2
...
...
real 0m0.331s
user 0m0.100s
sys 0m0.262s
```

#### 5. 安装 OceanBase 客户端。

OceanBase 实例兼容 Oracle 或 MySQL，如果是 Oracle 租户，Java 程序连接需要使用 OceanBase 提供的 Java 驱动文件（oceanbase-client-\*.jar）。如果要在命令行下访问 Oracle 租户，还需要安装客户端 OBClient。

OBClient 是 OceanBase 命令行客户端，可以访问 OceanBase 的 MySQL 租户和 ORACLE 租户。

示例如下：

```
-bash-4.2$ rpm -ivh obclient-1.2.6-20210510164331.el7.alios7.x86_64.rpm

-- 核查是否安装成功
-bash-4.2$ which obclient
/usr/bin/obclient
```

## 18.2.2 初始化 OceanBase 集群

### 说明

示例 IP 做了脱敏处理，这不是安装需求。在部署时应根据自己机器真实 IP 填写。

#### 1. 启动节点 observer 进程。

在每个节点的 `admin` 用户下，启动 observer 进程。

### 注意

三副本下，每个节点启动参数并不完全相同。启动 `observer` 时只需指定 `RootService` 所在的3台（或者多台）机器，不需要在创建集群时指定所有机器；集群创建完成后可以添加新的机器。

参数里指定数据文件的大小、内存的大小，以方便个别环境资源不足想精确控制 `observer` 对资源的占用。具体说明如下：

参数	说明
<code>-i</code>	指定网卡名，可通过 <code>ifconfig</code> 命令查看。
<code>-p</code>	指定服务端口号，一般指定为 2881。
<code>-P</code>	指定 RPC 端口号，一般指定为 2882。
<code>-n</code>	指定集群名称。可自定义，不同集群名称不要重复即可。
<code>-z</code>	指定启动的 <code>observer</code> 进程所属的 Zone。可自定义。
<code>-d</code>	指定集群主目录，除集群名字 <code>obdemo</code> 外，其他不要变动。
<code>-c</code>	指定集群 ID。为一组数字，可自定义，不同集群不要重复即可。
<code>-l</code>	指定日志级别。
<code>-r</code>	指定 RS 列表，格式是 <code>ip:2882:2881</code> ，分号分割，表示 <code>RootService</code> 信息。三台机器参数一致。
<code>-o</code>	指定集群启动参数，需要根据实际情况设置。 <code>system_memory</code> 指定 OceanBase 内部保留内存，默认是 30G，机器内存比较少的情况下把这个调小，影响就是可能在性能测试时有内存不足问题。 <code>datafile_size</code> 指定 OceanBase 数据文件 <code>sstable</code> 的大小（一次性初始化），根据 <code>/data/1/</code> 可用空间评估，建议不少于 100G，同时又保留一些剩余空间。如果 <code>/data/1</code> 跟 <code>/data/log1</code> 本身就是一个文件系统（共用一个盘），那么务必保证留给 <code>/data/log1</code> 的空间大小是内存的 2-4 倍。 <code>config_additional_dir</code> 指定参数文件的冗余目录。

示例如下：

### zone1:

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
/observer -i eth0 -P 2882 -p 2881 -z zone1 -d /home/admin/oceanbase/store
/obdemo -r '10.10.10.1:2882:2881;10.10.10.2:2882:2881;10.10.10.3:2882:2881' -
c 10001 -n obdemo -o "system_memory=30G,datafile_size=500G,
config_additional_dir=/data/1/obdemo/etc3;/data/log1/obdemo/etc2"
devname: eth0
rpc port: 2882
mysql port: 2881
zone: zone1
data_dir: /home/admin/oceanbase/store/obdemo
rs list: 10.10.10.1:2882:2881;10.10.10.2:2882:2881;10.10.10.3:2882:2881
cluster id: 10001
appname: obdemo
optstr: system_memory=30G,datafile_size=500G,config_additional_dir=/data/1
/obdemo/etc3;/data/log1/obdemo/etc2
```

### zone2:

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
/observer -i eth0 -P 2882 -p 2881 -z zone2 -d /home/admin/oceanbase/store
/obdemo -r '10.10.10.1:2882:2881;10.10.10.2:2882:2881;10.10.10.3:2882:2881' -
c 10001 -n obdemo -o "system_memory=30G,datafile_size=500G,
config_additional_dir=/data/1/obdemo/etc3;/data/log1/obdemo/etc2"
devname: eth0
rpc port: 2882
mysql port: 2881
```

```
zone: zone2
data_dir: /home/admin/oceanbase/store/obdemo
rs list: 10.10.10.1:2882:2881;10.10.10.2:2882:2881;10.10.10.3:2882:2881
cluster id: 10001
appname: obdemo
optstr: system_memory=30G,datafile_size=500G,config_additional_dir=/data/1
/obdemo/etc3;/data/log1/obdemo/etc2
```

### zone3:

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
/observer -i eth0 -P 2882 -p 2881 -z zone3 -d /home/admin/oceanbase/store
/obdemo -r '10.10.10.1:2882:2881;10.10.10.2:2882:2881;10.10.10.3:2882:2881' -
c 10001 -n obdemo -o "system_memory=30G,datafile_size=500G,
config_additional_dir=/data/1/obdemo/etc3;/data/log1/obdemo/etc2"
devname: eth0
rpc port: 2882
mysql port: 2881
zone: zone3
data_dir: /home/admin/oceanbase/store/obdemo
rs list: 10.10.10.1:2882:2881;10.10.10.2:2882:2881;10.10.10.3:2882:2881
cluster id: 10001
appname: obdemo
optstr: system_memory=30G,datafile_size=500G,config_additional_dir=/data/1
/obdemo/etc3;/data/log1/obdemo/etc2
```

可通过 `netstat -ntlp` 命令查看 `observer` 进程是否启动成功。如果监听到了 2881 与 2882 端口，则说明进程启动成功。

示例如下：

```
-bash-4.2$ netstat -ntlp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 0.0.0.0:2881 0.0.0.0:* LISTEN 45576/observer
tcp 0 0 0.0.0.0:2882 0.0.0.0:* LISTEN 45576/observer
tcp 0 0 127.0.0.1:9001 0.0.0.0:* LISTEN -
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN -
```

## 2. 集群 bootstrap 操作。

通过 obclient 命令连接任意一台节点，密码为空。

```
[root@xxx admin]# obclient -h127.1 -uroot -P2881 -p
Enter password:

obclient> SET SESSION ob_query_timeout=1000000000;
Query OK, 0 rows affected

obclient> ALTER SYSTEM BOOTSTRAP ZONE 'zone1' SERVER '10.10.10.1:2882',
ZONE 'zone2' SERVER '10.10.10.2:2882',ZONE 'zone3' SERVER '10.10.10.3:2882';
Query OK, 0 rows affected
```

### 注意

如果这一步失败报错了，其原因很可能就是三节点 observer 进程启动参数有不对、observer 相关目录权限不对、日志目录空间不足一定比例（跟数据目录合用了大目录，空间被数据目录占用了）、三节点时间不同步、节点内存资源不足等。请先排查这些问题点后，然后清理 OceanBase 目录从头开始（请参见 [清理 OceanBase 目录（第一次不用）](#)）。

## 3. 验证集群初始化成功。

进行 bootstrap 后，执行 `SHOW DATABASES;` 命令，能看到数据库列表里有 `oceanbase` 即可。

```
obclient> SHOW DATABASES;
```

```
+-----+
```

```
| Database |
```

```
+-----+
```

```
| oceanbase |
```

```
| information_schema |
```

```
| mysql |
```

```
| SYS |
```

```
| LBACSYS |
```

```
| ORAAUDITOR |
```

```
| test |
```

```
+-----+
```

```
7 rows in set
```

#### 4. 修改密码。

sys 租户的 root 用户密码默认为空，初始化成功后请修改密码。

```
obclient> ALTER USER root IDENTIFIED BY '*****';
```

```
Query OK, 0 rows affected
```



## 19 部署 OBProxy

OBProxy 可以将用户发起的数据访问请求转发到 OBCServer 上，转发到后端 OBCServer 的请求返回数据时，依然要通过 OBProxy 回发给客户端。

本节主要介绍通过使用 RPM 包安装 OBProxy。

### 注意

- OBProxy 在任何一个机器上部署后，通过向外暴露一个 `ip:port` 来提供 OceanBase 的代理服务。用户可以像访问 MySQL 数据库一样通过该 `ip:port` 即可访问 OceanBase 数据库，一般推荐部署在 OBCServer 节点上。
- 建议一台机器上只部署一个 OBProxy 服务，并且使用约定的 2883 端口。当同一台机器上部署多个 OBProxy 服务时，需要指定不同端口和不同的配置文件路径予以区分，端口可以自定义为 3306 或 1521、或者其他端口。

### 19.1 操作步骤

#### 19.1.1 步骤一：安装 OBProxy RPM 包

其中 `$rpm_dir` 表示存放 RPM 包的目录，`$rpm_name` 表示 RPM 包的名称。

```
[root@xxx /]# cd $rpm_dir
[root@xxx $rpm_dir]# rpm -ivh $rpm_name
```

### 说明

OBProxy 安装目录在 `/opt/taobao/install` 下。

示例如下：

```
[root@xxx admin]# rpm -ivh obproxy-4.0.0-20230109154442.el7.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
1:obproxy-4.0.0-20230109154442.el7 #####
[100%]
```

#### 19.1.2 步骤二：配置目录

## 1. （可选）创建到 obproxy 的软链接。

### 说明

创建到 obproxy 的软链接可以隐藏掉版本信息，方便后期操作。

obproxy 安装目录在 `/opt/taobao/install` 下。obproxy 进程 home 目录是 `/opt/taobao/install/obproxy`，这是一个软链接，实际指向相应的 obproxy 版本软件目录。

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /opt/taobao/install
-bash-4.2$ ls
ajdk-8.3.6-b129 obproxy-4.0.0
#创建到 obproxy 的软链接
-bash-4.2$ sudo ln -s obproxy-4.0.0 obproxy
[sudo] password for admin:
-bash-4.2$ ll
total 8
drwxr-xr-x 9 root root 4096 Sep 26 15:24 ajdk-8.3.6-b129
lrwxrwxrwx 1 root root 13 Jan 29 16:51 obproxy -> obproxy-4.0.0
drwxr-xr-x 5 admin admin 4096 Jan 29 16:42 obproxy-4.0.0

--把 obproxy 的所有者和所属组都改为 admin。
[root@xxx install]# chown -R admin:admin obproxy
[root@xxx install]# ll
total 8
drwxr-xr-x 9 root root 4096 Sep 26 15:24 ajdk-8.3.6-b129
lrwxrwxrwx 1 admin admin 13 Jan 29 16:51 obproxy -> obproxy-4.0.0
drwxr-xr-x 5 admin admin 4096 Jan 29 16:42 obproxy-4.0.0
```

## 2. 建立 obproxy 进程运行日志目录。

obproxy 进程的运行日志目录通过软链接指向 `/home/admin/logs/obproxy/log`。

```
[root@xxx admin]# su - admin
#创建日志目录
-bash-4.2$ mkdir -p /home/admin/logs/obproxy/log
#查看 obproxy 进程目录情况
-bash-4.2$ tree /opt/taobao/install/obproxy
/opt/taobao/install/obproxy
|-- bin
| |-- obp_xflush.py
| |-- obproxy
| |-- obproxyd.sh
| `-- unzip.py
|-- lib
| |-- libgpr.so.7
| |-- libgrpc++.so.1
| |-- libgrpc+_cronet.so.1
| |-- libgrpc.so.7
| |-- libgrpc_cronet.so.7
| |-- libobproxy_so.so
| |-- libprotobuf.so.18
| `-- libstdc++.so.6
|-- log -> /home/admin/logs/obproxy/log
|-- minidump -> /home/admin/logs/obproxy/minidump
|-- start_obproxy.sh
`-- tools
    |-- dump_syms
    |-- log4cplus.conf
    |-- minidump.sh
    |-- minidump_stackwalk
    `-- obproxy.sym
```

3 directories, 20 files

### 19.1.3 步骤三：初始化 OBProxy 账户

OBProxy 需要跟后端 OBServer 节点保持通信。所以需要提前在 OceanBase 集群的 sys 租户下为 OBProxy 创建连接用户（proxyro）和密码。后续在启动 OBProxy 时可通过设置 proxy 启动参数的方式来告知 proxy。

#### 说明

proxyro 用户是 OBProxy 访问 OceanBase 集群的用户。一个 OceanBase 集群对应一个 proxyro 账号。

示例如下：

创建 proxyro 用户并授予 SELECT 权限。

```
$obclient -h10.10.10.1 -P2881 -uroot@sys -p
```

Enter password:

```
obclient [(none)]> CREATE USER proxyro IDENTIFIED BY '123456';
```

Query OK, 0 rows affected

```
obclient [(none)]> GRANT SELECT ON *.* TO proxyro;
```

Query OK, 0 rows affected

```
obclient [(none)]> SHOW GRANTS FOR proxyro;
```

```
+-----+
```

```
| Grants for proxyro@% |
```

```
+-----+
```

```
| GRANT SELECT ON *.* TO 'proxyro' |
```

```
+-----+
```

1 row in set

### 19.1.4 步骤四：启动 OBProxy

## 注意

启动 OBProxy 时，请在 admin 用户下并在 OBProxy 软件的 home 目录。其他用户或者其他目录下启动都可能带来问题。

1. OBProxy 启动时需要知道 OceanBase 集群在哪里，这是通过参数 rootservice\_list 指定。

启动代码如下：

```
cd /opt/taobao/install/obproxy && bin/obproxy -r "10.10.10.1:2881;10.10.10.2:2881;10.10.10.3:2881" -p 2883 -o "observer_sys_password=$sha1_value,enable_strict_kernel_release=false,enable_cluster_checkout=false,enable_metadb_used=false" -c tenant_name
```

## 说明

- \$sha1\_value 应根据实际设置密码替换。设置 proxy 启动参数方式，设置的密码是 sha1 后的值，而不是原始值。例如：proxyro 用户设置的密码是 123456，则需要设置 observer\_sys\_password 的值是 7c4a8d09ca3762af61e59520943dc26494f8941b。
- 示例 IP 做了脱敏处理，这不是安装需求。在启动时应根据自己机器真实 IP 填写。

参数解释：

参数	说明
-r	指定 OBCServer 服务器 IP 和端口。
-p	指定服务端口号，一般指定为 2883。
-c	指定集群名称。
-o	指定硬件或者内核参数配置。

示例如下：

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /opt/taobao/install/obproxy && bin/obproxy -r "10.10.10.1:2881" -p 2883 -o
```

```
"observer_sys_password=7c4a8d09ca3762af61e59520943dc26494f8941b,
enable_strict_kernel_release=false,enable_cluster_checkout=false,
enable_metadb_used=false" -c test3241
bin/obproxy -r 10.10.10.1:2881 -p 2883 -o
observer_sys_password=7c4a8d09ca3762af61e59520943dc26494f8941b,
enable_strict_kernel_release=false,enable_cluster_checkout=false,
enable_metadb_used=false -c test3241
rs list: 10.10.10.1:2881
listen port: 2883
optstr: observer_sys_password=7c4a8d09ca3762af61e59520943dc26494f8941b,
enable_strict_kernel_release=false,enable_cluster_checkout=false,
enable_metadb_used=false
cluster_name: test3241
```

2. 启动之后，可以检查进程是否存在。

```
ps -ef|grep obproxy
```

示例如下：

```
[root@xxx admin]# ps -ef|grep obproxy
admin 79111 0 6 17:32 ? 00:00:58 bin/obproxy -r 10.10.10.1:2881 -p 2883 -o
observer_sys_password=7c4a8d09ca3762af61e59520943dc26494f8941b,
enable_strict_kernel_release=false,enable_cluster_checkout=false,
enable_metadb_used=false -c test3241
```

## 19.2 后续操作

### 19.2.5 连接 OceanBase 数据库

通过 OBProxy 连接时，用户名的格式需要包含用户名、租户名和集群名。格式为：**用户名@租户名#集群名** 或 **集群名:租户名:用户名**。

示例如下：

- **用户名@租户名#集群名** 连接 OceanBase 数据库。

```
[admin@xxx /home/admin]
$obclient -h10.10.10.1 -P2883 -uroot@sys#test3241 -p
Enter password:
Welcome to the OceanBase. Commands end with ; or \g.
Your OceanBase connection id is 1
Server version: OceanBase 3.2.4.1 (r101000052023010822-
346aa35c32e99d1b82d713f75f0072c45bdf7aab) (Built Jan 8 2023 22:52:43)

Copyright (c) 2000, 2018, OceanBase and/or its affiliates. All rights reserved.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

obclient [(none)]>
```

- **集群名:租户名:用户名** 连接 OceanBase 数据库。

```
[admin@xxx /home/admin]
$obclient -h10.10.10.1 -P2883 -utest3241:sys:root -p
Enter password:
Welcome to the OceanBase. Commands end with ; or \g.
Your OceanBase connection id is 4
Server version: OceanBase 3.2.4.1 (r101000052023010822-
346aa35c32e99d1b82d713f75f0072c45bdf7aab) (Built Jan 8 2023 22:52:43)

Copyright (c) 2000, 2018, OceanBase and/or its affiliates. All rights reserved.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

obclient [(none)]>
```

## 19.2.6 （可选）调整 OBProxy 参数

可以根据自己服务器的实际情况，通过调整 OBProxy 参数的数值来设置运行日志量或 CPU 消耗等信息。

示例如下：

下面是 OBProxy 的一些参数配置用于减少运行日志量或降低 CPU 消耗。

```
obclient [(none)]> ALTER PROXYCONFIG SET
slow_proxy_process_time_threshold='1000ms';
Query OK, 0 rows affected
```

```
obclient [(none)]> ALTER PROXYCONFIG SET xflush_log_level=ERROR;
Query OK, 0 rows affected
```

```
obclient [(none)]> ALTER PROXYCONFIG SET syslog_level=WARN;
Query OK, 0 rows affected
```

```
obclient [(none)]> ALTER PROXYCONFIG SET enable_compression_protocol=false;
Query OK, 0 rows affected
```

```
obclient [(none)]> SHOW PROXYCONFIG LIKE '%compress%'\G
***** 1. row *****
name: enable_compression_protocol
value: False
info: if enabled, proxy will use compression protocol with server
need_reboot: false
visible_level: USER
***** 2. row *****
name: enable_syslog_file_compress
value: False
info: Whether to enable archive log compression
need_reboot: false
```



```
visible_level: SYS
```

```
2 rows in set
```

## 19.3 相关文档

- 有关 OBProxy 部署的详细信息，请参见 [部署方式](#)。
- 有关调整 OBProxy 参数的详细信息，请参见 [配置参数说明](#)。
- 更多通过 OBProxy 连接 OceanBase 数据库的详细信息，请参见 [连接管理](#)。

## 20 查看业务租户可用资源

部署成功后，OceanBase 数据库默认会创建一个名为 `sys` 的租户，该租户用于管理 OceanBase 数据库，不建议在 `sys` 租户中存储业务数据。因此，在使用 OceanBase 数据库前，您需要创建业务租户。本节主要介绍查看业务租户可用资源的方法。

### 20.1 操作步骤

1. 登录 `sys` 租户。
2. 可以通过以下视图查看当前集群资源信息。

视图名	描述
DBA_OB_UNIT_CONFIGS	展示所有 Unit 规格信息。
GV\$OB_SERVERS	展示 server 资源等信息。
GV\$OB_UNITS	展示每台 server 的 Unit 信息。

3. 查看服务器的剩余资源。

```
obclient> SELECT SVR_IP,ZONE,(CPU_CAPACITY-CPU_ASSIGNED) cpu_free,round
((MEM_CAPACITY-MEM_ASSIGNED)/1024/1024/1024) mem_free_gb FROM
GV$OB_SERVERS;
```

```
+-----+-----+-----+-----+
| SVR_IP | ZONE | cpu_free | mem_free_gb |
+-----+-----+-----+-----+
| 10.10.10.1 | zone1 | 57.5 | 42 |
| 10.10.10.2 | zone2 | 57.5 | 42 |
| 10.10.10.3 | zone3 | 57.5 | 42 |
+-----+-----+-----+-----+
3 row in set
```

字段释义：

字段	释义
SVR_IP	observer 的 IP。
ZONE	zone 名称。
cpu_free	剩余可用 CPU。
mem_free_gb	剩余可用内存。

## 21 创建租户

本节主要介绍通过 SQL 命令创建 OceanBase 数据库租户的方法。OceanBase 数据库仅支持 sys 租户创建资源单元、资源池和业务租户。

### 21.1 描述

在创建新租户时，如果当前的资源池均被使用（被其他租户使用），需要创建新的资源池。在创建资源池前，需要先确定资源单元配置和资源使用范围。

### 21.2 操作步骤

#### 21.2.1 步骤一：创建资源单元

资源单元仅仅是规格定义，不实际分配资源。

##### 21.2.1.1 语法

```
CREATE RESOURCE UNIT unit_name
MAX_CPU [=] cpu_num,
[MIN_CPU [=] cpu_num,]
MEMORY_SIZE [=] mem_size,
[LOG_DISK_SIZE = disk_size,]
[MAX_IOPS [=] iops_num,MIN_IOPS [=] iops_num,IOPS_WEIGHT [=] weight_num];
```

## 21.2.1.2 参数解释

参数	描述
unit_name	指定资源单元名称。
MAX_CPU	指定 CPU 的最多数量。最小值为 1C。
MIN_CPU	指定 CPU 的最少数量，为可选项。默认等于 MAX_CPU。最小值为 1C。
MEMORY_SIZE	指定内存容量，最小值为 1G。
LOG_DISK_SIZE	指定租户的 Unit 日志盘大小，为可选项。默认值为 3 倍的内存规格值。最小值为 2G。
MAX_IOPS	指定 IOPS 的最多数量，为可选项。最小值为 1024。
MIN_IOPS	指定 IOPS 的最少数量，为可选项。最小值为 1024。
IOPS_WEIGHT	指定 IOPS 的权重值，为可选项。

### 说明

IOPS 规格参数默认不需要指定，系统根据 CPU 规格自动计算。计算规则如下：

- MAX\_IOPS 和 MIN\_IOPS 最小值 1024，并且要求 MAX\_IOPS 的值大于等于 MIN\_IOPS。
- 如果均未指定 MIN\_IOPS 和 MAX\_IOPS，则根据 MIN\_CPU 规格自动计算，1 个 core 对应 1 万 IOPS 值。
  - $MAX\_IOPS = MIN\_IOPS = MIN\_CPU * 10000$ 。
  - 如果未指定 IOPS\_WEIGHT，则  $IOPS\_WEIGHT = MIN\_CPU$ 。
  - 如果指定了 IOPS\_WEIGHT，则以指定为准。
- 如果只指定了 MAX\_IOPS，MIN\_IOPS 取 MAX\_IOPS 值；反之亦然。
  - 在 IOPS\_WEIGHT 未指定的情况下，默认值为 0。

## 21.2.1.3 示例

创建资源单元 unit001 配置为 2 个 CPU，6G 内存。

```
obclient> CREATE RESOURCE UNIT unit001
MAX_CPU 2, MEMORY_SIZE '6G';
Query OK, 0 rows affected
```

或者

```
obclient> CREATE RESOURCE UNIT unit001
MAX_CPU 2, MEMORY_SIZE 6442450944;
Query OK, 0 rows affected
```

## 21.2.2 步骤二：创建资源池

资源池会实际创建 unit，按规格定义分配资源。

### 21.2.2.4 语法

```
CREATE RESOURCE POOL pool_name
UNIT [=] unit_name,
UNIT_NUM [=] unit_num,
ZONE_LIST [=] ('zone' [, 'zone' ...]);
```

### 21.2.2.5 参数解释

参数	描述
pool_name	指定要创建的资源池的名称。
UNIT [=] unit_name	指定资源规格的名称。
UNIT_NUM [=] unit_num	指定要创建的单个 Zone 下的 Unit 个数。每个单元会根据当前集群负载，自动在每个 Zone 中选择一个 Server 负载，但同一个资源池的多个 Unit 不能分配到同一个 Server，即一个资源池包含的 Unit 个数不能超过单 Zone 内 Server 的个数。
ZONE_LIST [=] ('zone' [, 'zone' ...])	指定要创建的资源池所属的 Zone。一般与 zone 个数保持一致。

### 21.2.2.6 示例

创建资源池 pool001。

```
obclient> CREATE RESOURCE POOL pool001
UNIT='unit001',UNIT_NUM=1,ZONE_LIST=('zone1','zone2','zone3');
Query OK, 0 rows affected
```

## 21.2.3 步骤三：创建租户

将资源池分配给对应的租户。

### 21.2.3.7 语法

```
CREATE TENANT [IF NOT EXISTS] tenant_name
[tenant_characteristic_list] [opt_set_sys_var];

tenant_characteristic_list:
tenant_characteristic [, tenant_characteristic...]

tenant_characteristic:
COMMENT 'string'
| {CHARACTER SET | CHARSET} [=] charset_name
| COLLATE [=] collation_name
| PRIMARY_ZONE [=] zone
| DEFAULT TABLEGROUP [=] {NULL | tablegroup}
| RESOURCE_POOL_LIST [=](pool_name [, pool_name...])
| LOCALITY [=] 'locality description'

opt_set_sys_var:
{SET | SET VARIABLES | VARIABLES} system_var_name = expr [,system_var_name =
expr] ...
```

## 21.2.3.8 参数解释



参数	描述
tenant_name	指定租户名。最长 128 个字符，只能有大小写英文字母，数字和下划线，而且必须以字母或下划线开头，并且不能是 OceanBase 数据库的关键字。
IF NOT EXISTS	如果要创建的租户名已存在，并且没有指定 IF NOT EXISTS，则会报错。
RESOURCE_POOL_LIST	资源池列表，为创建租户时的必填项，暂时只支持一个 Resource Pool。
DEFAULT TABLEGROUP	设置租户默认表组信息，NULL 表示取消默认表组。如果不指定，默认为 NULL。
COMMENT	指定注释。
CHARACTER SET   CHARSET	指定租户的字符集。
COLLATE	指定字符序。
PRIMARY_ZONE	指定 Primary Zone。主副本分配到 Zone 内的优先级，逗号两侧优先级相同，分号左侧优先级高于右侧。比如 zone1,zone2;zone3。
LOCALITY	描述副本在 Zone 间的分布情况。例如：F@z1, F@z2,F@z3,R@z4 表示 z1、z2、z3 为全功能副本，z4 为只读副本。
system_var_name	<p>可选，指定租户的系统变量值，其中：</p> <ul style="list-style-type: none"> <li>变量 ob_compatibility_mode 用于指定租户的兼容模式，可选择 MySQL 或 Oracle 模式，并且只能在创建时指定。如果不指定 ob_compatibility_mode，则默认兼容模式为 MySQL 模式。</li> <li>变量 ob_tcp_invited_nodes 用于指定租户连接的白名单，即允许哪些客户端 IP 连接该租户。如果不调整 ob_tcp_invited_nodes 的值，则默认租户的连接方式为只允许本机的 IP 连接数据库。您也可以待租户创建成功后再修改其白名单设置，修改白名单的具体操作请参见 <a href="#">设置和查看租户白名单</a>。</li> </ul> <p>OceanBase 数据库中，租户可设置的系统变量请参见 <a href="#">系统变量</a>。</p>

## 21.2.3.9 示例

- 创建 MySQL 兼容模式租户。

```
obclient> CREATE TENANT IF NOT EXISTS mysql001
CHARSET='utf8mb4',
PRIMARY_ZONE='zone1;zone2,zone3',
RESOURCE_POOL_LIST=('pool001')
SET ob_tcp_invited_nodes='%';
Query OK, 0 rows affected
```

- 创建 Oracle 兼容模式租户。

```
obclient> CREATE TENANT IF NOT EXISTS oracle001
CHARSET='utf8mb4',
PRIMARY_ZONE='zone1;zone2,zone3',
RESOURCE_POOL_LIST=('pool002')
SET ob_tcp_invited_nodes='%',ob_compatibility_mode='oracle';
Query OK, 0 rows affected
```

## 21.2.4 步骤四：设置管理员用户密码

### 说明

使用 SQL 语句创建的租户，租户管理员用户密码为空。使用前请先设置密码。

示例如下：

将租户 `mysql001` 的 `root` 用户密码设置为 `*****`。

```
$ obclient -h10.10.10.1 -P2881 -uroot@mysql001 -p -A
Enter password:

obclient> SET PASSWORD = PASSWORD('*****');
Query OK, 0 rows affected
```

## 22 部署后验证

创建 OceanBase 租户后，您可以通过以下方式验证 OBProxy 服务与 OceanBase 数据库的连通性。

使用 MySQL 或 OBclient 客户端登录 OBProxy 代理服务的 ip:port，确保能联通 OceanBase 数据库并执行测试语句成功。

示例如下：

```
mysql -h[OBProxy的IP地址] -P2883 -uroot@[租户名]#[集群名] -Dtest -p'test' -c -Ne  
"SELECT 'ok'"
```

Warning: Using a password on the command line interface can be insecure.

+-----+

| ok |

+-----+

### 说明

连接参数解释如下：

- -h：提供 OceanBase 数据库连接 IP，通常是一个 OBProxy 域名。
- -u：提供租户的连接账户，格式有两种：用户名@租户名#集群名 或者 集群名：租户名:用户名。MySQL 租户的管理员用户名默认是 root。
- -P：提供 OceanBase 数据库连接端口，也是 OBProxy 的监听端口。
- -p：提供账户密码，为了安全可以不提供，改为在后面提示符下输入，密码文本不可见。
- -c：表示在 MySQL 运行环境中不要忽略注释。
- -Ne：表示在 MySQL 连接数据库时执行 SQL 语句。

如果返回 **ok**，则表示测试成功，客户端已通过代理服务连接到 OBCServer 节点上。

## 23 部署说明

OMS 是 OceanBase 提供的一种支持同构或异构 RDBMS 与 OceanBase 数据库之间进行数据交互的服务，它提供了数据的在线迁移和实时增量同步的数据复制能力。

## 24 修改配置文件

部署 OMS 前，需要在 OAT 的配置文件中添加 OMS 的基本配置信息，包括部署 OMS 机器的 IP 和 OMS Docker 容器的版本信息。

### 24.1 操作步骤

1. 以 root 用户登录曾执行过安装 OCP 的服务器。

#### 说明

安装 OCP 时本地生成了相关用户和密码信息

2. 进入 `/root/t-oceanbase-antman/obcluster.conf` 目录，修改配置文件中 OMS 相关的配置项。

根据环境和版本信息，主要修改以下四个配置项：

- OMS\_IP\_LIST：指要部署 OMS 的机器 IP 信息，多台用逗号分隔。
- OMS\_DOCKER\_IMAGE\_PACKAGE：OMS 镜像文件名。
- OMS\_IMAGE\_REPO：OMS 镜像 repo。
- OMS\_IMAGE\_TAG：OMS 镜像 TAG。

OMS 2.x/3.x 需要配置挂载目录：

- OMS\_PHYSICAL\_LOG\_DIR：挂载的日志目录。
- OMS\_PHYSICAL\_RUN\_DIR：挂载的运行目录。
- OMS\_PHYSICAL\_STORE\_DIR：挂载的数据存储目录。

在 OMS 2.1 之后的版本，自动安装 influxdb 用于查看监控, 2.1 之前版本会自动忽略。

- OMS\_INSTALL\_INFLUXDB=TRUE
- OMS\_INFLUXDB\_PHYSICAL\_DIR=/data/oms/influxdb
- OMS\_INFLUXDB\_IMAGE\_PACKAGE=influxdb\_1.8.tar.gz
- OMS\_INFLUXDB\_IMAGE\_REPO=influxdb
- OMS\_INFLUXDB\_IMAGE\_TAG=1.8

其他配置项默认不需要修改：

- OMS\_PORT：OMS drc 服务的端口号，默认为 8088。
- OMS\_METADB\_USER：OMS 的 MetaDB 用户，默认为 root。
- OMS\_METADB\_TENANT：OMS 的 MetaDB 租户，默认是 oms\_meta，创建在 OAT 的 obcluster 集群中。

- OMS\_METADB\_DBNAME: OMS 的 MetaDB database名, 默认是 oms\_meta。
- OMS\_CONTAINER\_NAME: OMS 容器的名字, 默认是 oms。
- OMS\_DOCKER\_CPUS=12: OMS 容器分配的 CPU 配额, 默认是 12C。
- OMS\_DOCKER\_MEMORY=24G: OMS 容器的内存配额, 默认是 24 GB。

修改后的示例如下:

```
##### OMS 基本配置 / MUST CHANGE ACCORDING
ENVIRONMENT / 根据环境必须修改 #####
OMS_IP_LIST=xxx.xxx.xxx.xxx
OMS_IMAGE_REPO=acs-reg.alipay.com/oceanbase/oms-all-in-one
OMS_IMAGE_TAG=1.1.1-beta
OMS_DOCKER_IMAGE_PACKAGE=oms.tgz

##### OMS ADVANCED SETTINGS / 基本不用修改
#####
OMS_PORT=8088
OMS_METADB_USER=root
OMS_METADB_TENANT=oms_tenant
OMS_METADB_TENANT_PASS=root
OMS_METADB_DBNAME=oms_meta
OMS_CONTAINER_NAME=oms
OMS_DOCKER_CPUS=12
OMS_DOCKER_MEMORY=24G
```

## 25 开始部署

### 25.1 前提条件

安装前，将确认已将 OMS 的 Docker image 保存成镜像文件，存放在 `/root/t-oceanbase-antman` 目录下。OMS 2.x/3.x 如需要查看监控信息，需要同时准备 influxdb 镜像并与 OMS 镜像存放在相同目录。

### 25.2 操作步骤

1. 以 root 用户登录曾执行过安装 OCP 的服务器。

#### 说明

安装 OCP 时本地生成了相关用户和密码信息

2. 执行以下命令，获取脚本帮助。

```
cd /root/t-oceanbase-antman
./install.sh -h
Usage: ./install.sh [OPTIONS]

Options:
-h, --help Print help and exit
-d, --debug Print debug information
-V, --version Print version
-i, --install-steps string For example 1,3-5,7-
-c, --clear-steps string For example 1,3-5,7-
-f, --config-file string Read in a config file
-l, --load-balance dns | haproxy

Steps:
1. ssh authorization
2. install load balancer (default: dns)
3. install ob server
```

4. init ocp metadb
5. install temp OCP
6. install obproxy
7. install OCP
8. POSTCHECK
9. install OMS (optional)
10. install ODC (optional)

根据返回结果，确认 OMS 的安装步骤为 9。

3. 执行以下命令，开始安装。

```
./install.sh -i 9
```

输出如下所示日志，则表示安装成功。

```
run install.sh with DEBUG=FALSE, INSTALL_STEPS= CLEAR_STEPS=9 CONFFILE=/root
/t-oceanbase-antman/obcluster.conf
[2019-08-16 14:26:03.135619] INFO [start antman API service]
LB_MODE=dns
[2019-08-16 14:26:03.155233] INFO [clear_step9: uninstalling oms and remove
docker, logfile: /root/t-oceanbase-antman/logs/uninstall_oms.log]
```

## 25.3 后续操作

如果后续不再需要 OMS，可以将其卸载。

执行以下命令，卸载 OMS。

```
cd /root/t-oceanbase-antman
./install.sh -c 9
```

输出如下所示日志，则表示卸载成功。

```
run install.sh with DEBUG=FALSE, INSTALL_STEPS= CLEAR_STEPS=9 CONFFILE=/root/t-
oceanbase-antman/obcluster.conf
[2019-08-16 14:26:03.135619] INFO [start antman API service]
```



```
LB_MODE=dns
```

```
[2019-08-16 14:26:03.155233] INFO [clear_step9: uninstalling oms and remove  
docker, logfile: /root/t-oceanbase-antman/logs/uninstall_oms.log]
```

## 26 部署后检查

OMS 部署完成后，您可以登录 OMS 进行部署后检查。

### 26.1 背景信息

不同 OMS 版本的操作界面可能不同，本节以 OMS V1.4.3 版本为例提供操作指导，OMS 的其他版本的操作请参考对应版本的《用户指南》文档。

### 26.2 操作步骤

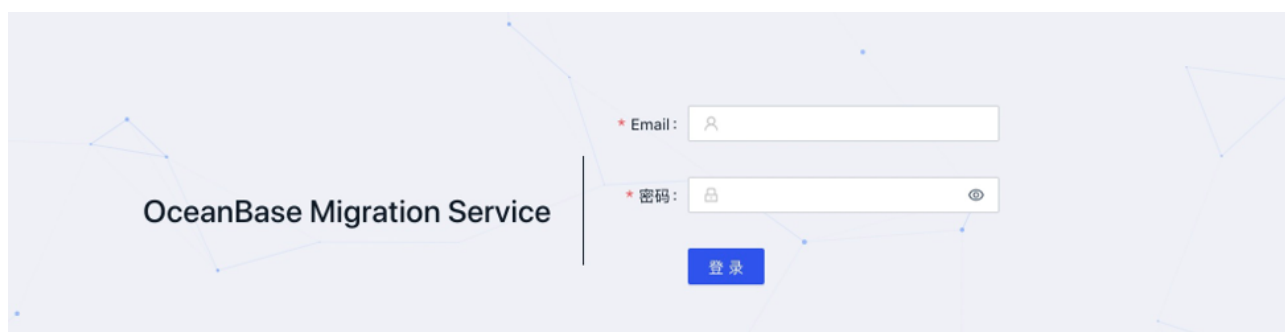
1. 通过 OMS 的访问地址访问 OMS。

访问地址为：`http://<OMS 服务器的 IP 地址>:8089`

2. 在弹出的登录页面，使用默认账号 `admin` 账号登录。

#### 说明

请联系 OceanBase 售后支持获取 `admin` 账号的默认密码。为保证账号安全，首次登录后请及时修改密码。



## 27 部署说明

ODC 是为 OceanBase 数据库量身打造的企业级数据库开发平台。ODC 支持连接 OceanBase 数据库中 MySQL 和 Oracle 模式下的数据库，同时为数据库开发者提供了数据库日常开发操作、WebSQL、SQL 诊断、会话管理和数据导入导出等功能。

## 28 修改配置文件

部署 ODC 前，需要在 OAT 的配置文件中填写 ODC Docker 容器的版本信息。

### 28.1 操作步骤

1. 以 root 用户登录曾执行过安装 OCP 的服务器。

#### 说明

安装 OCP 时本地生成了相关用户和密码信息。

2. 进入 `/root/t-oceanbase-antman/obcluster.conf` 目录，修改配置文件中 ODC 相关的配置项。

根据环境和版本信息，主要修改以下三个配置项：

- `odc_docker_image_package`：ODC 的镜像 tar 包。
- `ODC_image_REPO`：ODC 的镜像 repo。
- `ODC_image_TAG`：ODC 的镜像 TAG。

其他配置项默认不需要修改：

- `ODC_PORT`：ODC 服务的端口号，默认是 8989。
- `ODC_METADB_USER`：ODC 的 MetaDB 用户，默认是 root。
- `ODC_METADB_TENANT`：ODC 的 MetaDB 租户，创建在 OAT 的 obcluster 集群中。
- `ODC_METADB_DBNAME`：ODC 的 MetaDB database 名。
- `odc_container_name`：ODC 容器的名称，默认是 odc。
- `odc_docker_cpus`：ODC 容器分配的 CPU 配额，默认是 4C。
- `odc_docker_memory`：ODC 容器的内存配额，默认是 8 GB。

修改后的示例如下：

```
##### 如果准备部署ODC，请填写ODC基本配置 / Edit Configuration Of ODC
When Deploying ODC #####
odc_docker_image_package=odc.tar.gz
ODC_image_REPO=acs-reg.alipay.com/oceanbase/obodc
ODC_image_TAG=2.0.1

##### ODC 高级配置，基本不用修改 / ODC ADVANCED SETTINGS
```

```
#####
```

```
ODC_PORT=8989
```

```
ODC_METADB_USER=root
```

```
ODC_METADB_TENANT=odc_meta
```

```
ODC_METADB_DBNAME=odc
```

```
odc_container_name=odc
```

```
odc_docker_cpus=4
```

```
odc_docker_memory=8G
```

## 29 开始部署

### 29.1 前提条件

安装前，请确认已将 ODC 的 Docker image 保存成 odc.tar.gz，存放在 `/root/t-oceanbase-antman` 目录下。

### 29.2 操作步骤

1. 以 root 用户登录曾执行过安装 OCP 的服务器。

#### 说明

安装 OCP 时本地生成了相关用户和密码信息。

2. 执行以下命令，获取脚本帮助。

```
cd /root/t-oceanbase-antman
./install.sh -h
Usage: ./install.sh [OPTIONS]

Options:
-h, --help Print help and exit
-d, --debug Print debug information
-V, --version Print version
-i, --install-steps string For example 1,3-5,7-
-c, --clear-steps string For example 1,3-5,7-
-f, --config-file string Read in a config file
-l, --load-balance dns | haproxy

Steps:
1. ssh authorization
2. install load balancer (default: dns)
3. install ob server
```

4. init ocp metadb
5. install temp OCP
6. install obproxy
7. install OCP
8. POSTCHECK
9. install OMS (optional)
10. install ODC (optional)

根据返回结果，确认 ODC 的安装步骤为 10。

3. 执行以下命令，开始安装。

```
./install.sh -i 10
```

输出如下所示日志，表示安装成功。

```
run install.sh with DEBUG=FALSE, INSTALL_STEPS= CLEAR_STEPS=10 CONFFILE=
/root/t-oceanbase-antman/obcluster.conf
[2019-08-16 14:26:03.135619] INFO [start antman API service]
LB_MODE=dns
[2019-08-16 14:26:03.155233] INFO [clear_step9: uninstalling odc and remove
docker, logfile: /root/t-oceanbase-antman/logs/uninstall_odc.log]
```

## 29.3 后续操作

如果后续不再需要 ODC，可以将其卸载。

执行以下命令，卸载 ODC。

```
cd /root/t-oceanbase-antman
./install.sh -c 10
```

输出如下所示日志，表示卸载成功。

```
run install.sh with DEBUG=FALSE, INSTALL_STEPS= CLEAR_STEPS=10 CONFFILE=/root
/t-oceanbase-antman/obcluster.conf
[2019-08-16 14:26:03.135619] INFO [start antman API service]
```

LB\_MODE=dns

[2019-08-16 14:26:03.155233] INFO [clear\_step9: uninstalling odc and remove  
docker, logfile: /root/t-oceanbase-antman/logs/uninstall\_odc.log]



## 30 部署后检查

ODC 部署完成后，您可以登录 ODC 进行部署后检查。

不同 ODC 版本的操作界面可能不同，本节以 ODC V2.3.0 版本为例提供操作指导，ODC 的其他版本的操作请参考对应版本的《用户指南》文档。

### 30.1 访问 ODC

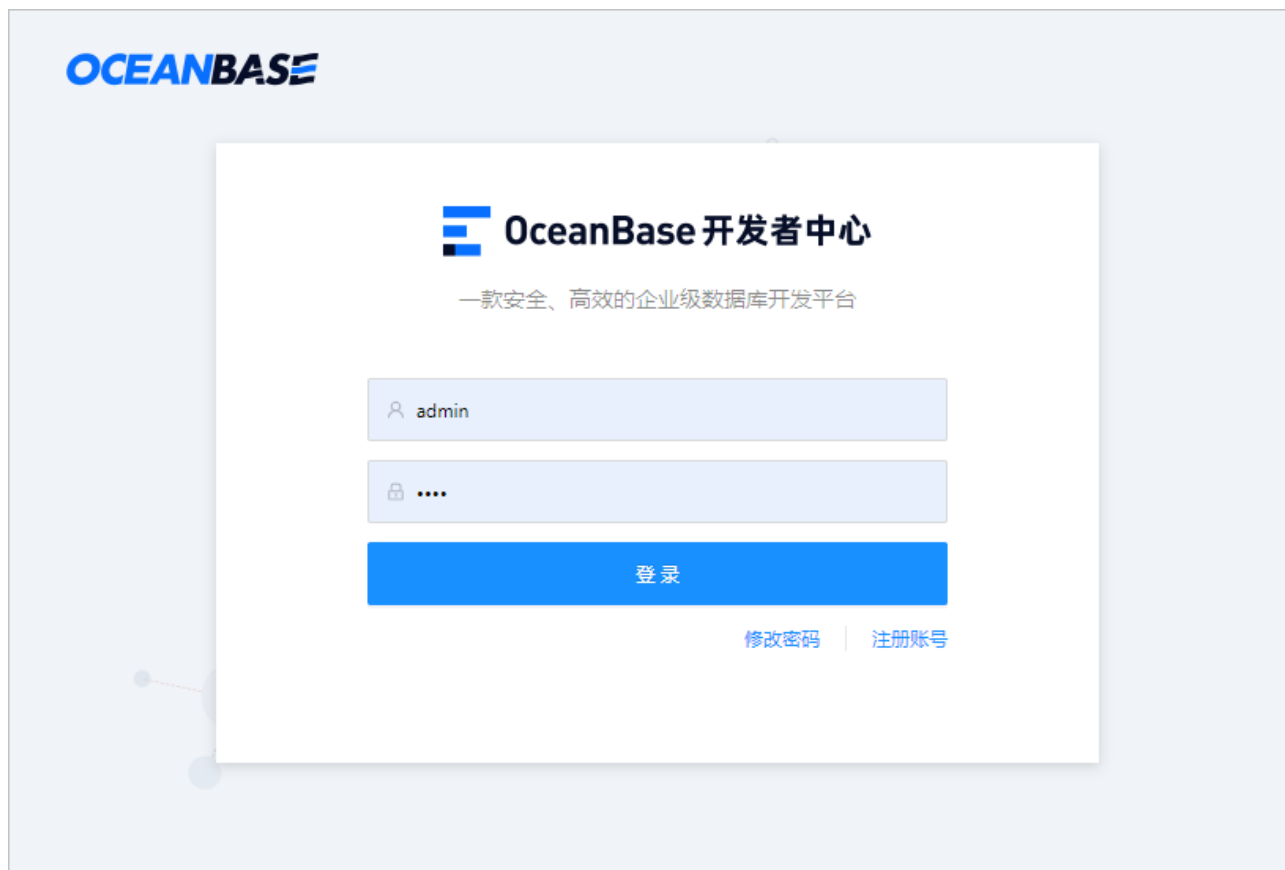
1. 通过 ODC 的访问地址访问 ODC。

访问地址为：`http://<ODC 服务器的 IP 地址>:8989`

2. 在弹出的登录页面，使用默认账号 `admin` 账号登录。

说明

请联系 OceanBase 售后支持获取 `admin` 账号的默认密码。为保证账号安全，首次登录后请及时修改密码。



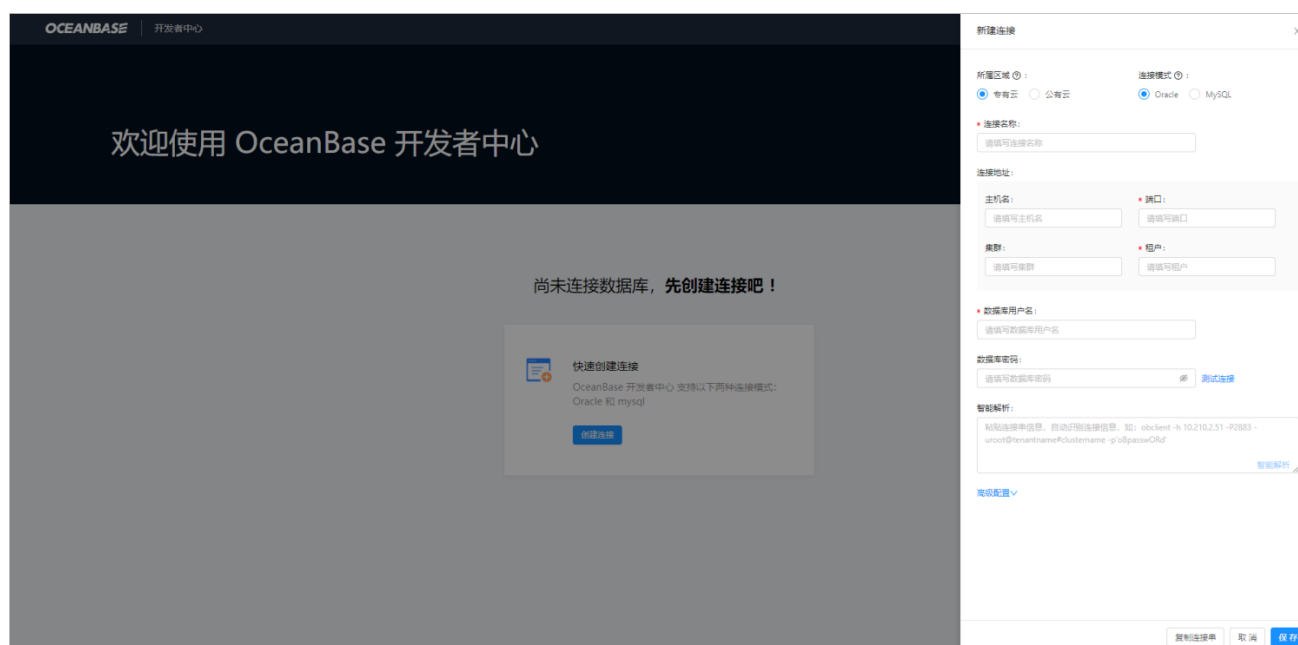
### 30.2 通过 ODC 连接 OceanBase 数据库

1. 进入 ODC 后，单击页面中的 **创建连接** 按钮以弹出创建连接页签。

2. 在创建连接页签需要填写或选择以下信息，如果您拥有 OceanBase 数据库连接串，那么以下信息均可从连接串中直接获得（`mysql -h 主机名 -P 端口号 -u 数据库用户名 @租户名 # 集群名 -D 默认数据库 -p '数据库密码'`）或直接使用页签中提供的智能解析工具进行填充：

信息项	说明
所属区域	选择连接示例所在的区域 公有云（阿里云上的数据库实例） 或 专有云（在自己环境中搭建的数据库实例）。
连接模式	选择 MySQL 或 Oracle。
连接名称	指定该连接的名称。
连接地址	主机名：在租户信息页面中获得主机名信息。端口号：在租户信息页面中获得端口号信息。集群：要连接的数据库所在集群的名称。租户：要连接的数据库所在租户的名称。
默认数据库（选填）	连接 MySQL 租户中的数据库时，可选填该连接要使用的默认数据库，连接 Oracle 租户中的数据库时则无需填写该项。
数据库用户名	在租户中创建的账号（MySQL 模式下该账户必须有访问默认数据库的权限）。
数据库密码	在租户中创建的账号的密码。单击数据库密码框后的 测试连接 按钮，测试是否能够连接到目标数据库。如果配置信息有误，会有异常提示，例如用户或密码不对、网络异常等。用户只能够连接本租户下的数据库，否则会报出测试失败的错误。
智能解析	粘贴连接串信息，ODC 会自动识别并填充连接信息。
SQL 查询超时时间	单击页面下方 高级配置 标签弹出该项。支持自定义 SQL 超时时间，默认 60 s，超过指定的时间时查询自动终止。
查询 sys 租户视图	单击页面下方 高级配置 标签弹出该项。账户：拥有 sys 租户视图查看权限的账号。密码：账号对应的密码。输入账户和密码信息后，单击密码框后的 测试连接 按钮测试账户信息是否正确。ODC V2.3.0 后使用数据导入导出和查看分区信息功能时需要在此项配置 root 账号。

- 单击页签下方的 **保存** 按钮，即可将当前的配置信息持久保存在 ODC 连接列表中。再次登录 ODC 时您可以在连接配置列表中直接选择该连接，保存连接配置时不要求测试连接一定成功。



- 在页签下方，ODC 还提供了 **复制连接串** 按钮。单击该按钮，ODC 会将您当前编辑的连接信息拷贝为连接串格式以方便您获得当前配置的连接信息。
- 当您不需要保存当前的连接配置时，可单击页签下方的 **取消** 按钮，ODC 会弹出弹窗跟您确认是否退出当前新建连接窗口。