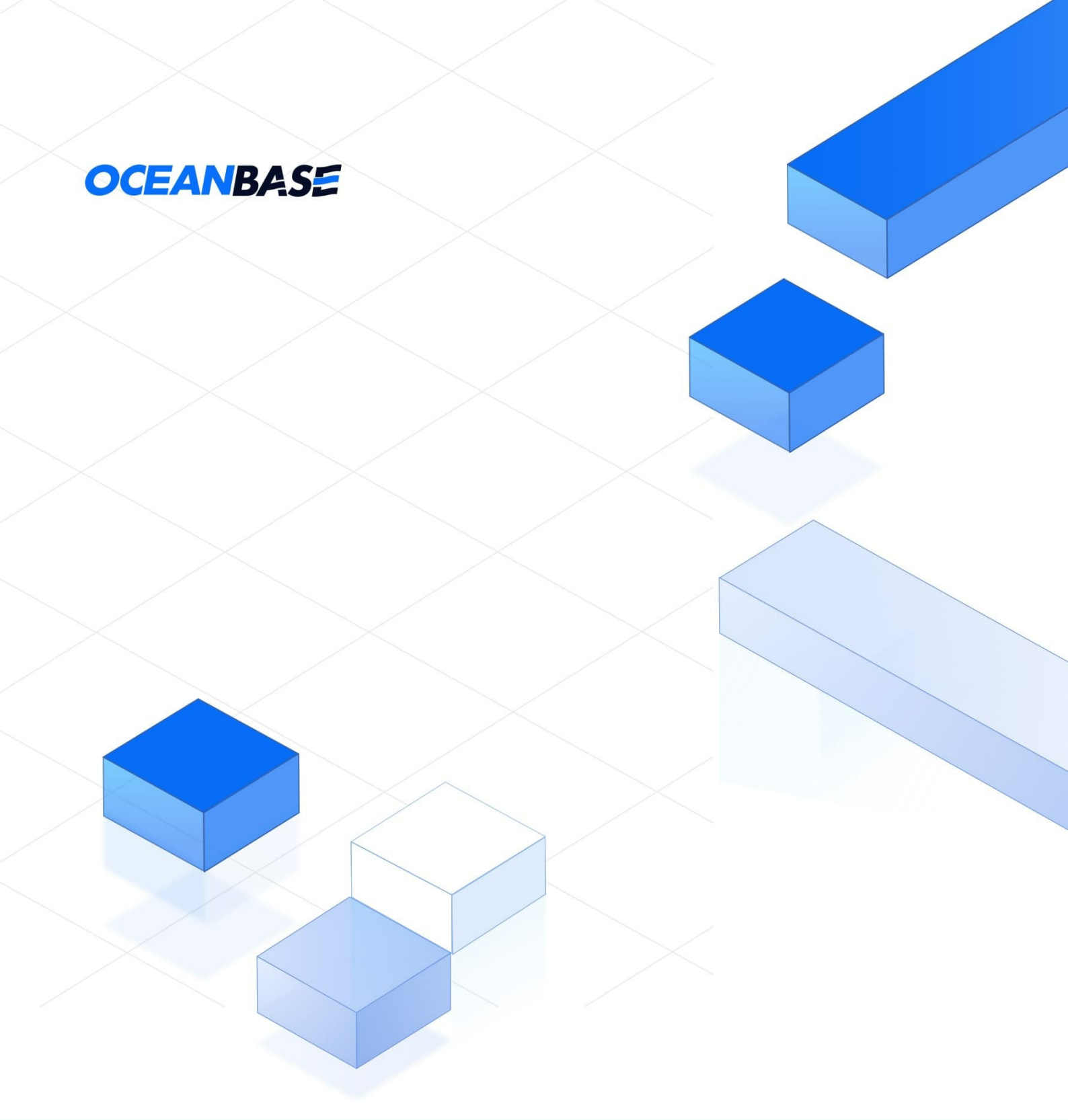


OCEANBASE



OceanBase 数据库

部署数据库

| 产品版本：V4.2.1

| 文档版本：202406181636

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。

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。	 危险 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。	 警告 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。	 注意 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。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击 设置> 网络> 设置网络类型 。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面，单击 确定 。
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 部署概述	14
1.1 部署方式	14
1.2 适用场景	14
1.3 相关文档	15
2 OceanBase 集群高可用部署方案简介	16
2.1 部署方案	16
2.1.1 方案一：同城三机房三副本部署	16
2.1.2 方案二：三地五中心五副本部署	16
2.1.3 方案三：同城两机房 主-备 部署	17
2.1.4 方案四：两地三中心 主-备 部署	17
2.1.5 方案五：同城三机房仲裁服务部署	18
2.1.6 方案六：三地五机房仲裁服务部署	19
2.1.7 方案七：两地三机房仲裁服务部署	20
2.2 相关文档	20
3 部署流程	21
3.0.0.1 说明	21
3.1 相关文档	21
4 准备服务器	22
4.1 使用 OCP 部署 OceanBase 集群	22
4.1.0.1 说明	23
4.1.0.2 说明	23
4.1.0.3 说明	23
4.1.0.4 说明	23
4.1.0.5 说明	23
4.2 使用命令行部署 OceanBase 集群	24
4.2.0.1 说明	24
4.2.0.2 说明	24
4.2.0.3 说明	24
4.2.0.4 说明	24
4.3 仲裁服务器	25
4.3.1 资源规划	25
4.3.1.1 说明	26
4.3.2 网络延迟要求	26

5	服务器配置	27
5.1	准备操作系统	27
5.1.1	OAT/OCP 操作系统	27
5.1.2	OceanBase 数据库操作系统	27
5.1.2.1	说明	28
5.2	OceanBase 服务器 BIOS 设置	28
5.2.1	特殊设置	28
5.2.2	BIOS 需要关闭以下选项	29
5.2.3	BIOS 需要配置以下选项	29
5.2.3.1	说明	30
5.3	磁盘挂载	30
5.3.0.1	说明	30
5.4	网卡设置	30
6	准备安装包	31
6.0.0.1	说明	31
6.1	使用 OCP 部署 OceanBase 集群	31
6.1.0.1	说明	31
6.1.0.2	注意	32
6.2	使用命令行部署 OceanBase 集群	32
6.2.0.1	说明	33
7	部署 OAT	34
7.1	前提条件	34
7.1.0.1	注意	34
7.2	操作步骤	34
7.2.1	方案一：使用 oat-all-in-one 安装包部署 OAT	34
7.2.1.1	说明	37
7.2.2	方案二：使用 OAT docker image 独立部署 OAT	37
7.2.2.1	说明	38
7.2.2.2	说明	39
7.3	后续操作	39
7.3.0.1	注意	40
7.4	相关文档	41
8	添加服务器	42
8.0.0.1	说明	42
8.1	前提条件	42
8.2	操作步骤	42
8.2.0.1	注意	45
8.2.0.2	注意	45
8.3	后续操作	46
8.4	相关文档	46

9	OCP 部署说明	47
9.1	描述	47
9.2	单节点部署	47
9.2.0.1	说明	47
9.3	高可用部署	47
9.3.0.1	说明	47
10	上传安装包	48
10.1	操作步骤	48
10.1.1	步骤一：上传 OCP 安装包到 OAT 的服务器上	48
10.1.1.1	注意	48
10.1.2	步骤二：将安装包添加到 OAT 服务中	48
10.2	相关文档	49
11	创建 MetaDB	50
11.0.0.1	说明	50
11.1	前提条件	50
11.2	操作步骤	50
11.2.0.1	注意	51
11.2.0.2	注意	51
11.2.0.3	说明	51
11.2.0.4	说明	51
11.2.0.5	注意	52
11.3	后续操作	53
11.4	相关文档	54
12	部署 OCP	55
12.0.0.1	说明	55
12.1	前提条件	55
12.2	操作步骤	55
12.2.0.1	注意	56
12.2.0.2	注意	56
12.2.0.3	注意	57
12.2.0.4	注意	58
12.2.0.5	注意	58
12.2.0.6	注意	58
12.3	后续操作	62
12.3.0.1	说明	63
12.4	相关文档	63
13	使用 OCP 部署单机集中式 OceanBase 数据库	64
13.0.0.1	说明	64
13.1	前提条件	64
13.2	操作步骤	64
13.3	后续操作	67
13.4	相关文档	67

14 使用 OCP 部署两副本加仲裁服务的 OceanBase 集群	69
14.0.0.1 说明	69
14.1 前提条件	69
14.2 操作步骤	69
14.2.1 步骤一：创建两副本集群	69
14.2.2 步骤二：新建仲裁服务	73
14.2.2.1 注意	74
14.2.2.2 注意	74
14.2.3 步骤三：为集群添加仲裁服务	75
14.3 后续操作	76
14.4 相关文档	76
15 使用 OCP 部署单副本 OceanBase 集群	78
15.0.0.1 说明	78
15.1 前提条件	78
15.2 操作步骤	78
15.3 后续操作	82
15.4 相关文档	83
16 使用 OCP 部署三副本 OceanBase 集群	84
16.0.0.1 说明	84
16.1 前提条件	84
16.2 操作步骤	84
16.3 后续操作	89
16.4 相关文档	89
17 （可选）部署 OBProxy	90
17.0.0.1 说明	90
17.1 前提条件	90
17.2 操作步骤	90
17.2.0.1 说明	92
18 使用 oatcli 初始化 OBServer 服务器	94
18.1 前提条件	94
18.2 操作步骤	94
18.2.0.1 说明	94
18.3 后续操作	105
18.4 相关文档	105
19 （可选）配置时钟源	106
19.1 前提条件	106
19.2 操作步骤	106
19.2.0.1 说明	106
19.2.0.2 说明	107

20 使用命令行部署单机集中式 OceanBase 数据库	110
20.1 注意事项	110
20.2 前提条件	110
20.3 操作步骤	110
20.3.1 步骤一：安装 RPM 包	110
20.3.1.1 说明	110
20.3.1.2 注意	111
20.3.2 步骤二：配置目录	111
20.3.2.1 说明	112
20.3.2.2 说明	113
20.3.3 步骤三：初始化 OceanBase 数据库	114
20.3.3.1 说明	114
20.3.3.2 说明	114
20.3.3.3 注意	118
20.4 后续操作	119
20.5 相关文档	119
21 使用命令行部署两副本加仲裁服务的 OceanBase 集群	120
21.0.0.1 说明	120
21.1 注意事项	120
21.2 前提条件	120
21.3 操作步骤	120
21.3.1 步骤一：部署 OceanBase 集群	120
21.3.1.1 说明	121
21.3.1.2 注意	121
21.3.1.3 说明	123
21.3.1.4 说明	124
21.3.1.5 说明	124
21.3.1.6 注意	124
21.3.1.7 说明	125
21.3.1.8 注意	128
21.3.2 步骤二：部署仲裁服务	129
21.3.2.1 说明	130
21.3.2.2 说明	130
21.3.2.3 说明	131
21.3.2.4 注意	132
21.3.3 步骤三：为集群添加仲裁服务	133
21.3.3.1 说明	133
21.3.4 步骤四：为 sys 租户开启仲裁服务	134
21.3.5 步骤五：查询租户仲裁服务状态	134
21.4 后续操作	135
21.5 相关文档	136

22 使用命令行部署单副本 OceanBase 集群	137
22.1 注意事项	137
22.2 前提条件	137
22.3 操作步骤	137
22.3.1 步骤一：安装 RPM 包	137
22.3.1.1 说明	137
22.3.2 步骤二：配置目录	138
22.3.2.1 说明	139
22.3.2.2 说明	140
22.3.3 步骤三：初始化 OceanBase 集群	140
22.3.3.1 说明	140
22.3.3.2 说明	141
22.3.3.3 注意	144
22.4 后续操作	145
22.5 相关文档	145
23 使用命令行部署三副本 OceanBase 集群	146
23.1 注意事项	146
23.2 前提条件	146
23.3 操作步骤	146
23.3.1 步骤一：安装 RPM 包	146
23.3.1.1 说明	146
23.3.1.2 注意	147
23.3.2 步骤二：配置目录	147
23.3.2.1 说明	148
23.3.2.2 说明	149
23.3.3 步骤三：初始化 OceanBase 集群	149
23.3.3.1 说明	150
23.3.3.2 注意	150
23.3.3.3 说明	150
23.3.3.4 注意	154
23.4 后续操作	155
23.5 相关文档	155

24	(可选) 部署 OBProxy	157
24.0.0.1	注意	157
24.1	操作步骤	157
24.1.1	步骤一：安装 OBProxy RPM 包	157
24.1.1.1	说明	157
24.1.2	步骤二：配置目录	158
24.1.2.1	说明	158
24.1.3	步骤三：初始化 OBProxy 账户	160
24.1.3.1	说明	160
24.1.4	步骤四：启动 OBProxy	161
24.1.4.1	注意	161
24.1.4.2	说明	161
24.2	后续操作	162
24.2.1	连接 OceanBase 数据库	163
24.2.2	(可选) 调整 OBProxy 参数	163
24.3	相关文档	164
25	OceanBase 数据库社区版部署概述	166
25.1	快速部署	166
25.2	标准部署	166
25.3	在线体验	166
26	软硬件要求	167
26.1	软件资源介绍	167
26.1.0.1	注意	167
26.2	硬件资源准备	167
26.2.0.1	注意	168
26.2.0.2	说明	169
27	(可选) 查看资源	170
27.0.0.1	说明	170
27.1	查看操作系统	170
27.1.1	Red Hat Enterprise Linux Server 7.X	170
27.1.2	CentOS 7.X	170
27.1.3	Anolis OS 8.X	170
27.1.4	Unbutu	171
27.1.5	Debian9	172
27.2	查看内核	172
27.3	查看内存	172
27.4	查看磁盘	173
27.5	查看网卡名称	173
27.5.0.1	注意	173

28	(可选) 设置无密码 SSH 登录	174
28.1	名词解释	174
28.2	操作步骤	174
28.2.1	(附录) 脚本参考	174
28.2.1.1	注意	176
29	(可选) 配置时钟源	177
29.0.0.1	说明	177
29.1	前提条件	177
29.2	操作步骤	177
29.2.0.1	注意	178
30	(可选) 规划磁盘	180
31	(可选) 配置 limits.conf	182
31.1	更改配置	182
31.1.0.1	说明	182
31.1.0.2	说明	183
31.2	查看配置	183
32	(可选) 配置 sysctl.conf	185
32.0.0.1	说明	185
32.1	更改配置	185
32.1.0.1	注意	186
32.2	加载配置	186
32.3	ARM 环境部署建议	186
33	(可选) 关闭防火墙和 SELinux	188
33.0.0.1	说明	188
33.1	关闭防火墙	188
33.2	关闭 SELinux	188
34	(可选) 创建用户	189
34.0.0.1	说明	189
35	通过 OBD 白屏部署 OceanBase 集群	191
35.1	前提条件	191
35.1.0.1	注意	191
35.2	准备软件	191
35.2.1	在线部署	191
35.2.2	离线部署	192
35.3	操作步骤	192
35.3.0.1	说明	192
35.3.0.2	说明	197
35.3.0.3	注意	197
35.4	相关操作	198
35.4.1	Java 环境验证	198
35.4.1.1	说明	198
35.4.2	管理部署后的集群	199
35.4.3	部署特定版本组件	200

36 通过 OCP 部署 OceanBase 集群	201
36.1 前提条件	201
36.2 操作步骤	201
36.2.0.1 说明	201
36.2.1 步骤一：添加机器到 OCP 资源池	201
36.2.1.1 说明	204
36.2.2 步骤二：创建 OceanBase 集群	205
36.2.2.1 注意	208
36.2.2.2 说明	209
36.2.2.3 说明	211
36.3 相关文档	212
37 使用命令行部署 OceanBase 数据库生产环境	213
37.0.0.1 说明	213
37.1 名词解释	213
37.2 前提条件	213
37.3 部署模式	214
37.3.0.1 说明	214
37.4 操作步骤	214
37.4.0.1 注意	214
37.4.1 步骤一：（可选）下载并安装 all-in-one 安装包	215
37.4.1.1 在线安装	215
37.4.1.2 离线安装	215
37.4.1.3 说明	215
37.4.2 步骤二：配置 OBD	215
37.4.2.1 说明	215
37.4.2.2 说明	216
37.4.2.3 说明	216
37.4.2.4 说明	216
37.4.2.5 说明	217
37.4.2.6 说明	217
37.4.2.7 注意	218
37.4.2.8 注意	225
37.4.2.9 注意	227
37.4.3 步骤三：部署 OceanBase 集群	228
37.4.3.1 说明	228
37.4.4 步骤四：连接 OceanBase 集群	229
37.4.4.1 注意	229
37.4.4.2 说明	230
37.4.5 步骤五：创建用户租户	230
37.5 相关操作	230
37.5.1 管理集群	230
37.5.2 卸载 all-in-one	231

38 在 Kubernetes 环境中部署 OceanBase 集群	232
38.1 前提条件	232
38.1.0.1 说明	232
38.2 部署 ob-operator	232
38.2.1 使用 Helm 部署	232
38.2.2 使用配置文件部署	232
38.2.2.1 ob-operator 自定义	233
38.3 部署 OceanBase 集群	233
38.3.1 直连 OceanBase 集群	237
38.4 部署 ODP	238
38.4.1 通过 ODP 连接 OceanBase 集群	241
38.5 监控 OceanBase 集群	242
38.5.1 部署 OceanBase Dashboard	242
38.5.2 访问 OceanBase Dashboard	243
38.5.3 查看监控指标	245
38.5.3.1 说明	245
38.6 相关文档	246
39 清理旧集群	247
39.1 清理旧集群建议	247
39.2 场景一：使用命令行部署的 OceanBase 集群	247
39.2.1 步骤一：停止和卸载服务	247
39.2.1.1 注意	247
39.2.2 步骤二：数据库文件清理	248
39.2.2.1 注意	248
39.3 场景二：使用 OBD 部署的 OceanBase 集群	248
39.3.0.1 注意	250
39.4 相关文档	250

1 部署概述

OceanBase 数据库提供企业版和社区版两种形态：

- OceanBase 数据库企业版：OceanBase 企业版是一款完全自研的企业级原生分布式数据库，在普通硬件上实现金融级高可用，首创 “三地五中心” 城市级故障自动无损容灾新标准，刷新 TPC-C 标准测试，单集群规模超过 1500 节点，具有云原生、强一致性、高度兼容 Oracle/MySQL 等特性。
- OceanBase 数据库社区版：兼容 MySQL 的单机分布式一体化数据库，具有原生分布式架构，支持金融级高可用、透明水平扩展、分布式事务、多租户和语法兼容等企业级特性。OceanBase 社区版数据库内核开源，同时社区版提供开放的接口和丰富的生态能力，支持企业或个人更好的实现定制化业务需求。

OceanBase 企业版和社区版的功能差异的详细信息，请参见 [企业版和社区版的功能差异](#)。

本手册将主要介绍 OceanBase 企业版和 OceanBase 社区版的部署方式和适用场景。

1.1 部署方式

OceanBase 企业版：

- 使用 OCP 部署 OceanBase 集群
- 使用 OBD 部署 OceanBase 集群
- 使用命令行部署 OceanBase 集群

OceanBase 社区版：

- 使用 OBD 部署 OceanBase 集群
- 使用 OCP 部署 OceanBase 集群
- 使用 Docker 镜像的方式部署 OceanBase 集群
- 在 Kubernetes 环境中使用 ob-operator 部署 OceanBase 集群

1.2 适用场景

OceanBase 企业版：

- 对于生产环境，建议使用 OCP 部署 OceanBase 集群。具体操作请参见 [使用 OCP 部署三副本 OceanBase 集群](#)。
- 对于非生产环境，可以使用命令行部署 OceanBase 集群。具体操作请参见 [使用命令行部署三副本 OceanBase 集群](#)。
- 对于非生产环境的快速体验场景，可以使用 OBD 部署 OceanBase 集群。具体操作参见 [快速体验 OceanBase 数据库](#) 中的 **方案一：部署 OceanBase 演示环境** 或 **方案二：部署 OceanBase 集群环境**。

OceanBase 社区版：

- 对于非原生支持的操作系统（比如 MAC 和 Windows）的快速体验场景，建议使用 Docker 镜像的方式进行部署。具体操作参见 [快速体验 OceanBase 数据库](#) 中的 **方案三：部署 OceanBase 容器环境** 一节。
- 对于原生支持的操作系统（Linux 系列，具体见支持的操作系统列表）的快速体验场景，建议使用 OBD 进行部署；具体操作参见 [快速体验 OceanBase 数据库](#) 中的 **方案一：部署 OceanBase 演示环境** 或 **方案二：部署 OceanBase 集群环境**。
- 对于线上环境，建议使用 OBD 进行标准部署。具体操作参见 [通过 OBD 白屏部署 OceanBase 集群](#)。
- 对于 kubernetes 环境，建议使用 ob-operator 的方式部署。具体操作参见 [在 Kubernetes 集群中部署 OceanBase 数据库](#)。

1.3 相关文档

- 有关 OceanBase 企业版部署流程的详细信息，请参见 [部署流程](#)。
- 有关 OceanBase 社区版的部署信息，请参见 [OceanBase 数据库社区版部署概述](#)。
- 有关 OBD 的信息，请参见 [什么是 OBD](#)。

2 OceanBase 集群高可用部署方案简介

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（单机房部署 OceanBase 集群）、机房（同城多机房部署 OceanBase 集群。机房以下统称：IDC）、城市（多城市部署 OceanBase 集群）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，且支持部署仲裁服务来降低成本。

2.1 部署方案

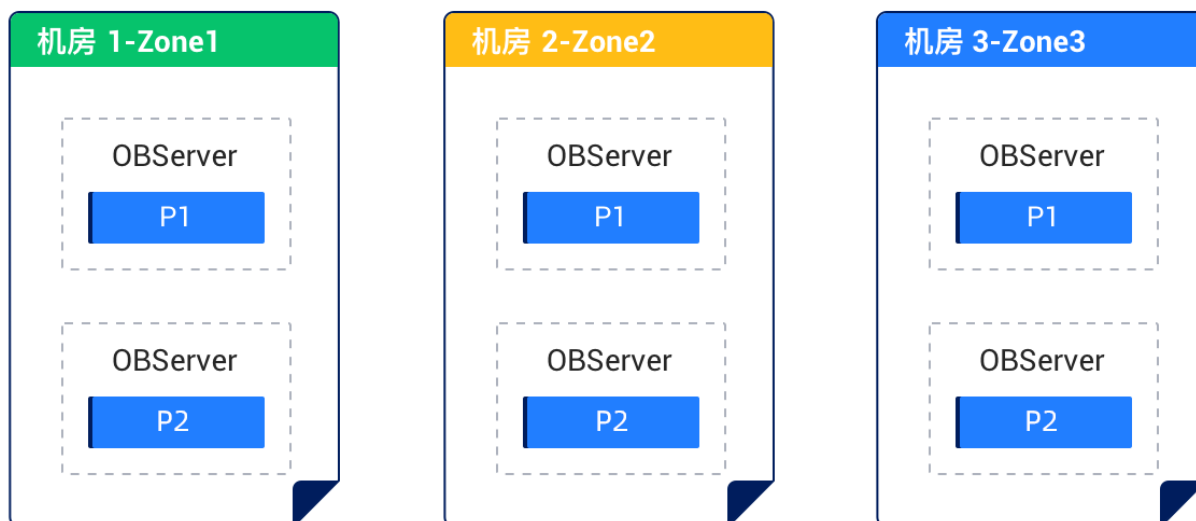
2.1.1 方案一：同城三机房三副本部署

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 RedoLog 日志，保证 RPO=0。
- 无法应对城市级的灾难。

部署方案示意图：

杭州



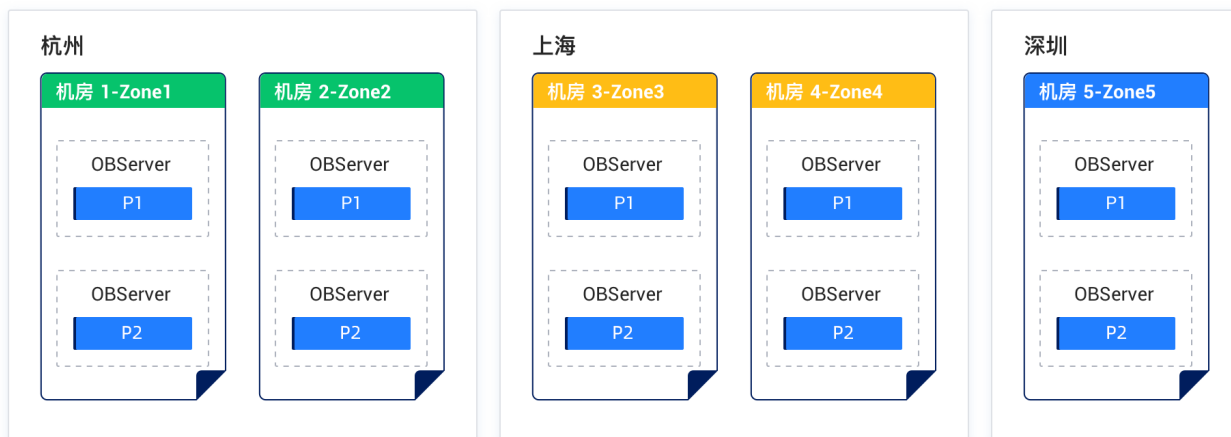
2.1.2 方案二：三地五中心五副本部署

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 RedoLog 的时延。

部署方案示意图：

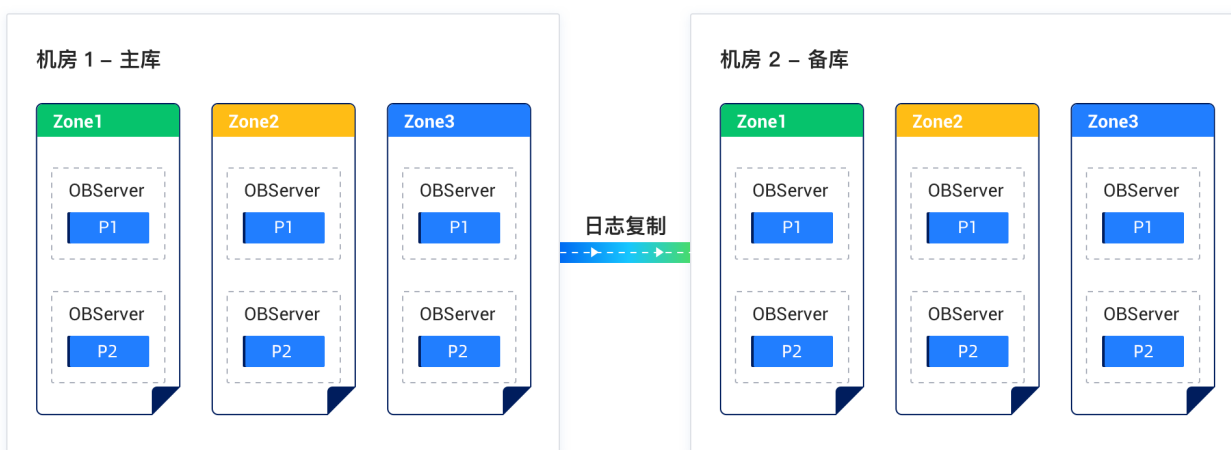


2.1.3 方案三：同城两机房 "主-备" 部署

特点：

- 每个机房都部署一个 OceanBase 集群，一个为主库一个为备库；每个集群有自己单独的 Paxos group，多副本一致性。
- "集群间" 通过 RedoLog 做数据同步，形式上类似传统数据库 "主从复制" 模式，从主库 "异步同步" 到备库，类似 Oracle Data Guard 中的 "最大性能" 模式。

部署方案示意图：



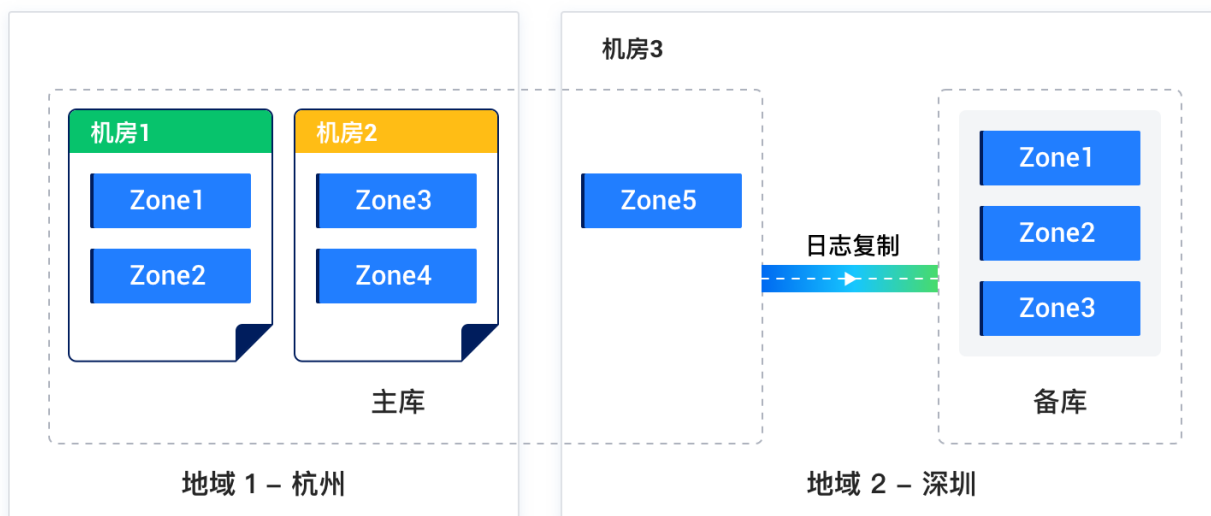
2.1.4 方案四：两地三中心 "主-备" 部署

特点：

- 主城市与备城市组成一个 5 副本的集群。任何主城市 IDC 的故障，最多损失 2 份副本，剩余的 3 份副本依然满足多数派。

- 备用城市建设一个独立的 3 副本集群，做为一个备库，从主库 "异步同步" 到备库。
- 一旦主城市遭遇灾难，备城市可以接管业务。

部署方案示意图：



2.1.5 方案五：同城三机房仲裁服务部署

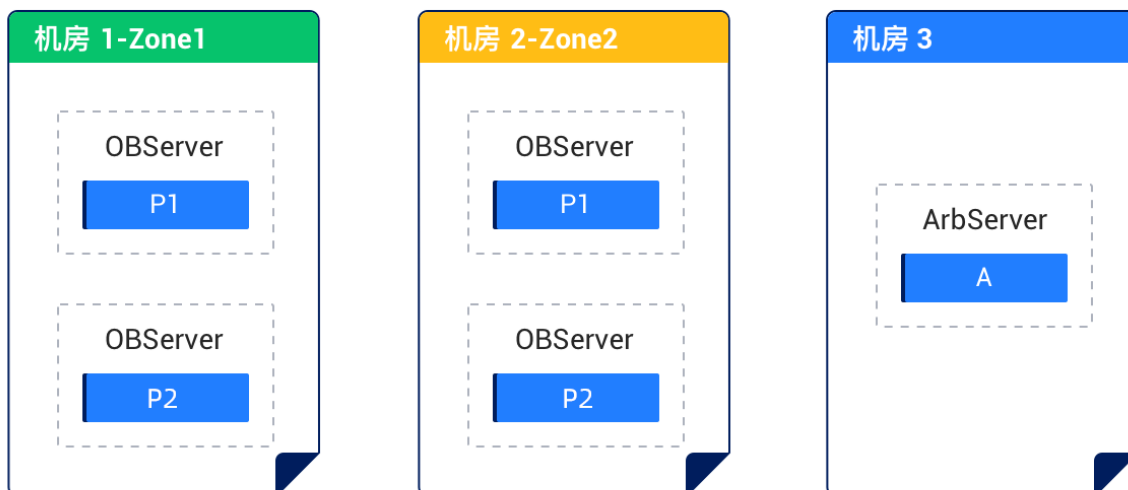
特点：

- 同城 3 个机房组成一个集群，机房间网络延迟一般在 0.5 ~ 2 ms 之间，其中两个机房放置全功能副本，分别作为一个 Zone，为了降低成本，第三机房部署仲裁服务（无需同步日志）。
- 机房级灾难时，剩余两个机房的副本可以选主、执行仲裁降级（全功能副本所在机房故障时），继续同步 RedoLog 日志，保证 RPO=0。
- 无法应对城市级的灾难。

有关仲裁服务的详细信息，请参见 [仲裁服务概述](#)。

部署方案示意图：

地域 – 杭州



2.1.6 方案六：三地五机房仲裁服务部署

特点：

- 三个城市，五个机房，城市 1 和城市 2 距离较近，部署全功能副本，城市 3 部署仲裁服务以降低成本（无需同步日志）。
- 任何一个 IDC 故障，剩余全功能副本依然满足多数派（3/4），可以确保 RPO=0。
- 任意两个 IDC 或者城市级故障，如果故障的都是全功能副本所属机房，剩余两个全功能副本不足多数派（2/4），可通过仲裁降级方式恢复服务（将故障的两个副本降级为 Learner），并确保 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 RedoLog 的时延。

部署方案示意图：

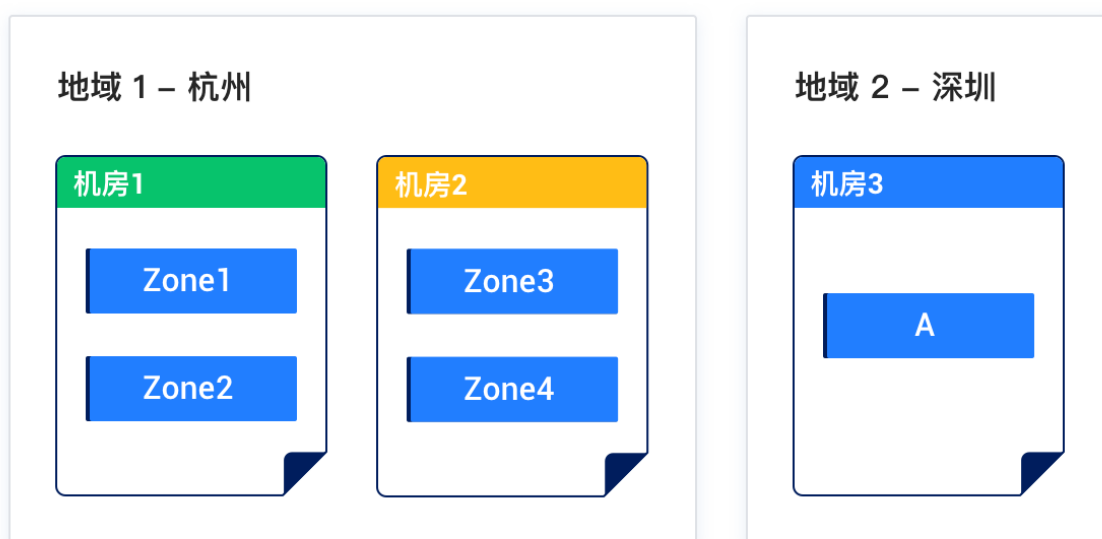


2.1.7 方案七：两地三机房仲裁服务部署

特点：

- 主城市有两个机房，分别包含两个 Zone，用于部署全功能副本。
- 备城市一个机房，部署仲裁服务，可降低部署成本以及跨城带宽开销。
- 主城市任意一个 IDC 故障，至多损失 2 份副本，此时可能不足多数派（2/4），可通过仲裁服务触发降级恢复，可以确保 RPO=0。
- 无法应对主城市灾难，备城市灾难无影响。

部署方案示意图：

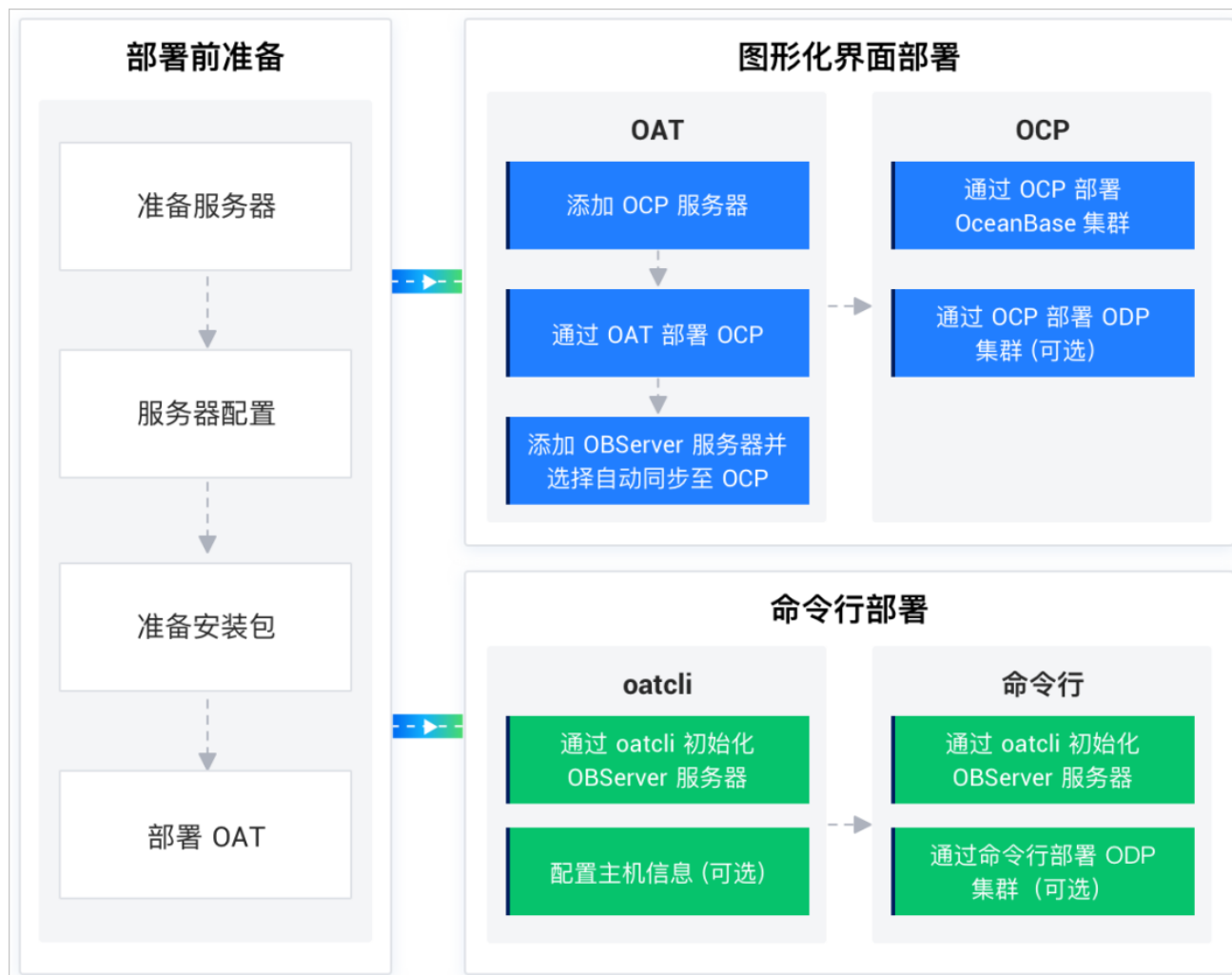


2.2 相关文档

- [容灾部署方案](#)
- [仲裁服务概述](#)

3 部署流程

本节主要介绍 OceanBase 集群的部署流程。



3.0.0.1 说明

oatcli 是管理 OceanBase 管理者工具 (OceanBase Admin Toolkit, 简称: OAT) 资源的命令行工具。

3.1 相关文档

- [准备服务器](#)
- [服务器配置](#)
- [准备安装包](#)
- [部署 OAT](#)

4 准备服务器

4.1 使用 OCP 部署 OceanBase 集群

服务器应满足的最低配置要求如下表所示：

部署产品	服务器数量	功能最低配置	性能最低配置	磁盘类型
OAT	1 台，可复用 OCP 服务器	CPU: 2 核 内存: 4 GB	N/A	N/A
OCP	1 台	CPU: 16 核 内存: 32 GB 磁盘: 1.5 TB 存储 (包含 OAT 所需资源) 4.1.0.1 说明 16C32G 的配置无法保证 OCP 的稳定性，建议只做验证使用。	CPU: 32 核 内存: 128 GB 磁盘: 1.5 TB, 万兆网卡 (包含 OAT 所需资源)	SSD 存储
OceanBase 集群	3 台 4.1.0.2 说明 如果部署单机版 OceanBase 数据库只需准备一台服务器。	CPU: 4 核 内存: 16 GB 4.1.0.3 说明 • 磁盘中日志盘需要内存的 4 倍以上，数据盘要满足目标数据量的存储。 • OBCore 服务器内存设置建议不超过 1T。	CPU: 32 核 内存: 256 GB, 万兆网卡 4.1.0.4 说明 • 磁盘中日志盘需要内存的 4 倍以上，数据盘要满足目标数据量的存储。 • OBCore 服务器内存设置建议不超过 1T。	SSD 存储
(可选) ODP	3 台，可复用 OBCore 节点服务器	CPU: 4 核 内存: 8 GB 磁盘: 200 GB	N/A	N/A

4.1.0.5 说明

- 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。

- 如果需要 OCP 管控服务提供高可用能力，则需要多台管控服务器进行高可用部署，详情请参见 [OCP 部署概述](#)。
- 在部署 OceanBase 集群后，是否需要部署 ODP 主要取决于您的应用场景和需求：
 - 如果您的应用需要对 OceanBase 集群进行高可用、负载均衡或连接池管理等操作，那么可以考虑部署 ODP。ODP 可以有效地解决 OceanBase 集群的连接数和负载均衡等问题，提高应用程序的稳定性和可靠性。
 - 如果您的应用只是简单地连接到 OceanBase 集群进行查询和操作，而不需要进行高可用、负载均衡或连接池管理等操作，那么可以不部署 ODP，直接使用 OceanBase 提供的客户端库连接到集群，也可以满足应用的需求。

ODP 的详细特性，请参见 [ODP 概述](#)。

4.2 使用命令行部署 OceanBase 集群

服务器应满足的最低配置要求如下表所示：

部署产品	服务器数量	功能最低配置	性能最低配置	磁盘类型
OAT	1 台	CPU：2 核 内存：4 GB	N/A	N/A
OceanBase 集群	3 台 4.2.0.1 说明 如果部署单机版 OceanBase 数据库只需准备一台服务器。	CPU：4 核 内存：16 GB 4.2.0.2 说明 ● 磁盘中日志盘需要内存的 4 倍以上，数据盘要满足目标数据量的存储。 ● OBDServer 服务器内存设置建议不超过 1T。	CPU：32 核 内存：256 GB，万兆网卡 4.2.0.3 说明 ● 磁盘中日志盘需要内存的 4 倍以上，数据盘要满足目标数据量的存储。 ● OBDServer 服务器内存设置建议不超过 1T。	SSD 存储
(可选) ODP	3 台，可复用 OBDServer 节点服务器	CPU：4 核 内存：8 GB 磁盘：200 GB	N/A	N/A

4.2.0.4 说明

- 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。
- 在部署 OceanBase 集群后，是否需要部署 ODP 主要取决于您的应用场景和需求：
 - 如果您的应用需要对 OceanBase 集群进行高可用、负载均衡或连接池管理等操作，那么可以考虑部署 ODP。ODP 可以有效地解决 OceanBase 集群的连接数和负载均衡等问题，提高应用程序的稳定性和可靠性。
 - 如果您的应用只是简单地连接到 OceanBase 集群进行查询和操作，而不需要进行高可用、负载均衡或连接池管理等操作，那么可以不部署 ODP，直接使用 OceanBase 提供的客户端库连接到集群，也可以满足应用的需求。

ODP 的详细特性，请参见 [ODP 概述](#)。

4.3 仲裁服务器

如果您准备使用三台服务器部署两副本的 OceanBase 集群和仲裁服务，那么仲裁服务器应满足以下所示的最小机器规格：

版本	支持的最小规格	带宽	备注
V4.1.0.0	• CPU：2 核 • 内存：8 GB • 磁盘：CLog 盘需要至少 5GB 的磁盘空间。	要求输入和输出的数据传输速率都至少为 20 Mbps。	不需要配置用于存储数据的磁盘空间（data 盘）；存放日志文件的 log 目录需要单独预留空间。
V4.1.0 BP1 及以后版本	• CPU：2 核 • 内存：4 GB • 磁盘：CLog 盘需要至少 5GB 的磁盘空间。	要求输入和输出的数据传输速率都至少为 20 Mbps。	不需要配置用于存储数据的磁盘空间（data 盘）；存放日志文件的 log 目录需要单独预留空间。

4.3.1 资源规划

CPU 型号参考：Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz

仲裁服务器的资源规划参考如下规则：

- 最小规格机器（2C4G）默认可支持 32 个用户租户和 sys 租户同时开启仲裁（2F 或 4F，primary_zone=RANDOM，unit_num=1）。
- 每新增 32 个单 Unit 租户（2F 或 4F，primary_zone=RANDOM，unit_num=1），仲裁服务器资源需相应地至少增加：1C1G，日志盘 2GB，网络带宽 20 Mbps。
- 对于 unit_num=N（N>1）的场景，可简单地当做 N 个单 Unit 租户处理。

4.3.1.1 说明

- 2F：两副本集群。
- 4F：四副本集群。
- `primary_zone=RANDOM`：在指定 `PRIMARY_ZONE` 时，其值可以设置为 `RANDOM`（必须大写），表示随机选择最高优先级内的任意一个 Zone 作为 Primary Zone。
- `unit_num=1`：资源池在目标 Zone 的 Unit 个数是 1。
- 单 Unit 租户：在一个 Zone 里的 Unit 个数是 1 的租户。

更多有关 `primary_zone` 和 `unit_num` 的信息，请参见 [创建租户](#)。

4.3.2 网络延迟要求

仲裁服务器和 OceanBase 集群节点之间的单程网络延迟小于等于 800ms。

5 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。

5.1 准备操作系统

5.1.1 OAT/OCP 操作系统

支持在下表所示的操作系统中部署 OAT/OCP。

操作系统	支持版本	服务器类型
RHEL	7.2 及以上版本	x86_64
CentOS	7.2 及以上版本	x86_64
openSUSE	12SP3 及以上	x86_64
AnolisOS	8.6 及以上版本	x86_64, ARM aarch64
Alibaba Cloud Linux	2,3	x86_64, ARM aarch64
中标麒麟	7.6	ARM aarch64
华为 EulerOS	2.0 SP8	ARM aarch64

5.1.2 OceanBase 数据库操作系统

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
Alibaba Cloud Linux	2	x86_64, ARM_64
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	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)
Debian	8.3 及以上	x86_64 (包括海光)

5.1.2.1 说明

操作系统需要配置网络和软件管理器（yum 或 zypper 源）。

5.2 OceanBase 服务器 BIOS 设置

5.2.3 特殊设置

- 在 Intel x86 架构的环境中：
 - 建议关闭 NUMA。
 - 建议修改配置文件 `/etc/sysctl.conf`，将参数 `vm.swappiness` 设置为 `0`，方法如下：

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

应用 `sysctl.conf` 配置文件里的值：

```
[root@xxx /]  
$sysctl -p
```

- 在 AMD 或者 ARM 架构的环境中，建议开启 Numa。
- 在 ARM 和海光架构的环境中，建议修改配置文件 `/etc/sysctl.conf`，将参数 `kernel.numa_balancing`、`vm.zone_reclaim_mode` 和 `vm.swappiness` 设置为 0，方法如下：

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

应用 `sysctl.conf` 配置文件里的值：

```
[root@xxx /]  
$sysctl -p
```

5.2.4 BIOS 需要关闭以下选项

- Cstate
- Pstate
- EIST
- Power saving

5.2.5 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
- Turbo Mode: 开启

- Energy performance：开启最大 performance

5.2.5.1 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。

5.3 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

- OBSERVER 节点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	observer 进程运行日志盘	建议 ext4
/data/log1	内存大小的 3~4 倍	observer 进程日志盘	建议 ext4
/data/1	取决于所需存储的数据大小	observer 进程数据盘	建议 ext4，大于 16 TB 使用 xfs

5.3.5.1 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
```

5.4 网卡设置

建议配置 2 块万兆网卡：

- bond 模式取名 bond0，绑定的模式可以选择 mode1 或 mode4，但建议选择 mode4。对于 mode4 模式，交换机需要配置 802.3ad 协议。
- 网卡名建议使用 eth0、eth1。

6 准备安装包

部署前，请确认已获取各产品使用的镜像和安装包，用于后续的安装部署。

6.0.0.1 说明

请在 OceanBase 官方网站左上角，单击 **资源** > [下载中心](#) 获取对应版本的安装包。如果未找到对应版本的安装包，请联系技术支持获取。

6.1 使用 OCP 部署 OceanBase 集群

使用 OCP 部署 OceanBase 集群使用的镜像和安装包如下表所示：

产品	组件	组件说明	推荐版本	安装包/镜像名
OceanBase 管理工具	OAT docker image 或者 oat-all-in-one 安装包			• OAT docker image: ■ X86_64 平台: oat_x.x.x_xxxxxxxx_x86.tgz ■ ARM_64 平台: oat_x.x.x_xxxxxxxx_arm.tgz • oat-all-in-one 安装包: ■ X86_64 平台: oat-all-in-one-x86-411.tar ■ ARM_64 平台: oat-all-in-one-arm-411.tar
	6.1.0.1 说明			
OceanBase 云平台	• 新环境推荐您使用 oat-all-in-one 安装包安装 OAT。	用于安装 OAT。	OAT V4.2.0	
	• oat-all-in-one 安装包提供了 Docker, OAT, MetaDB、NLB 镜像及常用工具软件包。			
OceanBase 云平台	OCP docker image	用于部署 OCP。	OCP V4.2.1	• X86_64 平台: ocpxxx.tar.gz • ARM_64 平台: ocpxxx.arm.tar.gz

	MetaDB docker image 6.1.0.2 注意 如果使用的 OAT docker image 安装 OAT, 安装 OCP 时还需要提供 MetaDB docker image。	用于部署 MetaDB、MonitorDB。	OCP	请联系技术支持获取最新 MetaDB 镜像。	• X86_64 平台: meta_xxxxxx_xxxxxx_x86_xxxx_xxxx.tgz • ARM_64 平台: meta_xxxxxx_xxxxxx_arm_xxx_xxxx.tgz
OceanBase 数据库	OceanBase RPM 包	用于部署 OceanBase 集群。	集群	OceanBase 数据库 V4.2.1	• X86_64 平台: oceanbase-x.x.x.x-xxxxxxxxxxx_xxxxxxxx.el7.x86_64.rpm • ARM_64 平台: oceanbase-x.x.x.x-xxxxxxxxxxx_xxxxxxxx.el7.aarch64.rpm
OceanBase 数据库代理 (可选)	ODP RPM 包	用于部署代理服务。		ODP V4.2.1	• X86_64 平台: obproxy-x.x.x.x-xxxxxxxxxxx_xx.el7.x86_64.rpm • ARM_64 平台: obproxy-x.x.x.x-xxxxxxxxxxx_xx.el7.aarch64.rpm

6.2 使用命令行部署 OceanBase 集群

使用命令行部署 OceanBase 集群使用的镜像和安装包如下表所示：

产品	组件	组件说明	推荐版本	安装包/镜像名
OceanBase 管理工具	OAT docker image 或者 oat-all-in-one 安装包 6.2.0.1 说明 - 新环境推荐您使用 oat-all-in-one 安装包安装 OAT。 - oat-all-in-one 安装包提供了 Docker, OAT, MetaDB、NLB 镜像及常用工具软件包。	用于安装 OAT。	OAT V4.2.0	- OAT docker image: - X86_64 平台: oat_x.x.x_xxxxxxxx_x86.tgz - ARM_64 平台: oat_x.x.x_xxxxxxxx_arm.tgz - oat-all-in-one 安装包: - X86_64 平台: oat-all-in-one-x86-411.tar - ARM_64 平台: oat-all-in-one-arm-411.tar
OceanBase 数据库	OceanBase RPM 包	用于部署 OceanBase 集群。	OceanBase 数据库 V4.2.1	- X86_64 平台: oceanbase-x.x.x.x-xxxxxxx.el7.x86_64.rpm - ARM_64 平台: oceanbase-x.x.x.x-xxxxxxx.el7.aarch64.rpm
OceanBase 数据库代理（可选）	ODP RPM 包	用于部署代理服务。	ODP V4.2.1	- X86_64 平台: obproxy-x.x.x.x-xxxxxxx.el7.x86_64.rpm - ARM_64 平台: obproxy-x.x.x.x-xxxxxxx.el7.aarch64.rpm

7 部署 OAT

本节主要介绍使用 Docker 部署 OAT。

7.1 前提条件

- 安装 OAT 服务器的操作系统满足要求。详细信息，请参见 [服务器配置](#)。
- 您已经安装并启动 Docker。推荐 Docker 版本为社区版 17.03 及以上。

7.1.0.1 注意

如果您使用 oat-all-in-one 安装包部署 OAT，oat-all-in-one 的 `install.sh` 脚本将自动安装 Docker，您无需再单独安装 Docker。

- OAT 的默认端口 `7000` 未被占用。
- 您拥有服务器的 `root` 用户权限。

7.2 操作步骤

7.2.1 方案一：使用 oat-all-in-one 安装包部署 OAT

1. 通过 `scp` 命令将 oat-all-in-one 安装包上传到服务器中。

其中 `oat_server_ip` 为部署 OAT 的服务器的 IP 地址，`oat_directory` 为部署存放安装包的目录，`oat_xxx.tar` 为安装包的名称。

```
scp <oat_directory/oat-all-in-one-xxx.tar> <oat_server_user>@<oat_server_ip>:  
oat_directory
```

示例如下：

```
[root@xxx /home/admin/rpm]# scp oat-all-in-one-x86-411.tar admin@xxx.xxx.  
xxx.xxx:/home/admin/rpm  
admin@xxx.xxx.xxx.xxx's password:  
oat-all-in-one-x86-411.tar 100% 1649MB 401.9MB/s 00:04
```

2. 解压 oat-all-in-one 安装包。

示例如下：

```
[root@xxx /home/admin/rpm]# tar -xf oat-all-in-one-x86-411.tar
```

3. 运行安装脚本 `install.sh`。

运行解压目录 `oat-all-in-one-x86` 中的安装脚本 `install.sh`。

```
sh oat-all-in-one-x86/install.sh
```

如果您的环境没有 Docker，运行此命令将自动安装 Docker，部署 OAT 并扫描已有的镜像和工具软件包。

根据提示，依次确认以下信息：

- 请指定 Docker 的根目录，默认值 `/docker`，可自定义。如果已安装 Docker，会跳过此步骤。
- 请指定 OAT 数据目录的路径，默认值 `/oat_data`，可自定义。
- 请指定 OAT HTTP 监听端口，默认值 `7000`，可自定义。
- 请指定 OAT 数据库端口，默认值 `3306`，可自定义。
- 请输入 OAT 管理员账户密码（设置初始登录密码），可自定义。
- 根据返回的 URL 访问 OAT，验证 OAT 服务的可用性。

示例如下：

```
[root@xxx /home/admin/rpm]# sh oat-all-in-one-x86/install.sh
```

根据提示信息，完成以下配置：

- Docker 的根目录使用默认值，直接回车。

```
Before installation, please set the config below:
```

```
Input the docker root dir: /docker
```

- OAT 数据目录的路径，修改为 `/home/admin/oat_data`，然后回车。

```
Input the OAT data dir: /home/admin/oat_data
```

- OAT HTTP 监听端口使用默认值，直接回车。

```
Input the OAT HTTP listen port: 7000
```

- d. OAT 数据库端口使用默认值，直接回车。

```
Input the OAT database port: 3306
```

- e. OAT 管理员账户密码（初始登录密码）使用默认值，直接回车。

```
Input the OAT admin user password(login password): *****
```

- f. 根据返回的 URL 访问 OAT，验证 OAT 服务的可用性。

返回结果如下：

```
Start prepare docker
Docker is already exists, start check...
Already installed docker check healthy, skip install docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.
service to /usr/lib/systemd/system/docker.service.
Start prepare OAT
Loaded image: reg.docker.alibaba-inc.com/oceanbase/oat:4.1.1
_20230512_x86
f4f5dbe127f62f9ec016dabfc6b1b272da2dd7794c255aed9e335cb83192ad2b
check OAT url http://127.0.0.1:7000/hc
curl: (7) Failed connect to 127.0.0.1:7000; Connection refused
OAT API not ready, please wait, sleep 5s retry...
curl: (7) Failed connect to 127.0.0.1:7000; Connection refused
OAT API not ready, please wait, sleep 5s retry...
curl: (7) Failed connect to 127.0.0.1:7000; Connection refused
OAT API not ready, please wait, sleep 5s retry...
curl: (7) Failed connect to 127.0.0.1:7000; Connection refused
OAT API not ready, please wait, sleep 5s retry...
OAT API ready
Copy images and binary_packages to OAT data dir
Trigger OAT scan api to find images and binary_packages
Trigger scan task success, please visit OAT web site and wait for scan task
```

```
finished
OAT is ready for visit
url is: http://xxx.xxx.xxx.xxx:7000
user/password is: XXXXX/*****
```

7.2.1.1 说明

- `install.sh` 脚本启动 OAT 容器默认使用 `--net host`（主机网络模式）。如您需要配置为其他网络模式，请下载 OAT 容器手动启动。
- x86（aarch64）版本的 `oat-all-in-one` 默认只包含 x86（aarch64）和 noarch 的镜像和软件包，如需 aarch64（x86）架构，请自行下载拷贝到 OAT 容器挂载的对应目录，然后发起扫描任务进行添加。

7.2.2 方案二：使用 OAT docker image 独立部署 OAT

1. 通过 `scp` 命令将 OAT 安装包上传到服务器中。

其中 `oat_server_ip` 为部署 OAT 的服务器的 IP 地址，`oat_directory` 为部署存放安装包的目录，`oat_xxx.tar` 为安装包的名称。

```
scp <oat_directory/oat_xxx.tar> <oat_server_user>@<oat_server_ip>:oat_directory
```

示例如下：

```
[root@xxx /home/admin/rpm]# scp oat_4.1.0_20230331_x86.tgz admin@xxx.xxx.xxx.xxx:/home/admin/rpm
admin@xxx.xxx.xxx's password:
oat_4.1.0_20230331_x86.tgz 100% 438MB 4.3MB/s 01:41
```

2. 挂载 OAT 目录。

在服务器创建 `/data_dir` 目录用来保存 OAT 的持久化数据。在 `/data_dir` 目录被挂载到 OAT 容器后，OAT 会自动创建 `/data_dir/logs`、`/data_dir/images` 和 `/data_dir/db` 目录，分别用于存放 OAT 的系统日志、组件和产品的 Docker 镜像以及 OAT 的数据库文件。示例如下：

```
[root@xxx /]# mkdir -p /data_dir
```

3. 将 OAT 安装包装载为镜像。

进入到存放 OAT 安装包的目录，执行以下语句：

```
[root@xxx /]# cd oat_directory
```

```
[root@xxx /oat_directory]# docker load -i oat_xxx.tar
```

示例如下：

```
[root@xxx /]# cd /home/admin/rpm
```

```
[root@xxx /home/admin/rpm]# docker load -i oat_4.1.0_20230331_x86.tgz
```

```
06f6bfff6616: Loading layer
```

```
[=====>] 230.8MB/230.8MB
```

```
e1505344677e: Loading layer
```

```
[=====>] 3.072kB/3.072kB
```

```
01ede0eada53: Loading layer
```

```
[=====>] 690.2MB/690.2MB
```

```
78073091fd9e: Loading layer
```

```
[=====>] 8.704kB/8.704kB
```

```
5d96997aeb89: Loading layer
```

```
[=====>] 232.8MB/232.8MB
```

```
17fa9a0a477e: Loading layer
```

```
[=====>] 156.1MB/156.1MB
```

```
Loaded image: reg.docker.alibaba-inc.com/oceanbase/oat:4.1.0_20230331_x86
```

4. 用 `docker images` 命令获取 OAT 镜像的标签。

7.2.2.2 说明

以下命令仅适用于本地只加载了一个 OAT 安装包的情况。对于装载多个安装包的情况，可以通过 `docker images` 查看显示内容，并使用冒号（:）连接前两列的内容。

```
oat_image=`docker images | grep oat | awk '{printf $1:"$2\n"}'`
```

示例如下：

```
[root@xxx /home/admin/rpm]# oat_image=`docker images | grep oat | awk '{printf $1:"$2\n"}'`
```

5. 运行 `docker run` 命令，启动 OAT。

其中 `$oat_image` 是 OAT 镜像的标签。

```
docker run -d --net host --name oat -v /data_dir:/data --restart on-failure:5  
$oat_image
```

7.2.2.3 说明

- OAT 的 HTTP 服务默认监听 7000 端口，可以指定 `-e HTTP_PORT=7001` 参数修改为其他端口。
- 您可以指定 `-e OAT_INITIAL_ADMIN_PASSWORD=xxx` 参数修改 OAT 的密码。
- OAT 内置了 MariaDB 作为数据存储，默认监听 3306 端口，可以指定 `-e DB_PORT=3307` 参数修改为其他端口。
- 建议使用 `--net host` 参数启动，因为 bridge 网络模式启动的容器可能被 docker0 网桥故障或操作系统参数 `ip_forward` 影响。

示例如下：

```
[root@xxx /home/admin/rpm]# docker run -d --net host --name oat -v /data_dir:  
/data --restart on-failure:5 `docker images | grep oat | awk '{printf $1:"$2\n"}'  
79978776c4d478d36b0b61d6ccfb186d39dbd2d29695a27937d0fc654b58ffb9
```

7.3 后续操作

OAT 部署成功后，您需要登录 OAT 来确认部署是否成功。

1. 在浏览器中输入 OAT 的访问地址，按回车键。

OAT 的访问地址：`http://oat_server_ip:7000`。

其中 `oat_server_ip` 为部署 OAT 的服务器的 IP 地址。

2. 如果在浏览器窗口中看到登录界面，表示已经成功安装并启动 OAT。

在显示的登录页面，使用 `admin` 账号登录系统。

7.3.2.1 注意

请联系 OceanBase 售后支持获取 OAT 默认用户名 `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。



3. 设置密码激活账号。

使用 `admin` 账号的默认密码登录，然后设置密码激活账号。



7.4 相关文档

更多安装 OAT 的信息，请参见 [安装 OAT](#)。

8 添加服务器

本文主要介绍如何将服务器添加到 OAT 资源池。

8.0.0.1 说明

本节以 OAT V4.2.0 版本为例提供操作指导。不同 OAT 版本的操作界面可能不同，请以实际界面为准。

8.1 前提条件

您已安装 OAT。详细信息，请参见 [部署 OAT](#)。

8.2 操作步骤

1. 登录 OAT。
2. 在左侧导航栏中，单击 **服务器** 进入 **服务器管理** 页面，在页面右上角，单击 **添加服务器**。
3. 在 **添加服务器** 页面中，输入基础信息。

基础信息

服务器 IP

按回车键可批量输入多个服务器 IP

SSH 端口

22

机房 (IDC) ?

请选择



凭据 ?

请选择



说明 (可选)

请输入

具体字段说明如下：

字段	描述
服务器 IP	待添加的服务器的外网网卡 IP。
SSH 端口	待添加的服务器的 SSH 访问端口。
机房（IDC）	待添加的服务器所属的机房。
凭据	待添加的服务器的访问凭据。当前支持用户名和密码及密钥两种类型的凭据。
说明（可选）	添加服务器的注释，便于更好地管理主机。

4. 在 **添加服务器** 页面中，进行初始化配置。

初始化配置 ☒

服务器用途 ☒ OB Server ☒ OB 产品服务

OB Server + OB 产品服务：适用于专门部署 MetaDB 或与其他产品服务共同部署（如 OCP），会执行磁盘检查并安装 docker。由于 MetaDB 使用 docker daemon 实现开机启动，因此不会设置 OB 开机自启动脚本，不适合部署业务 OB。

操作系统属主用户

用户密码

请牢记密码，也可复制密码并妥善保存

用户 UID:GID

docker 根目录

时钟同步 ☐

自动同步至 OCP ☐

具体字段说明如下：

字段	描述
服务器用途	待添加的服务器用途。
操作系统属主用户	安装、运行 OB Server 及 OB 产品服务的操作系统用户，可自定义设置。
用户密码	对应上述操作系统属主用户密码。
用户 UID:GID	操作系统属主用户对应的 UID 和 GID。
docker 根目录	Docker 数据目录。

- 时钟同步（可选）：默认为关闭。开启时钟同步，则需要填写以下信息：

8.2.0.1 注意

部署 OceanBase 集群的服务器和部署 OCP 的服务器需做时钟源同步。

字段	描述
时钟源 IP	时钟源 IP。
minpoll:	chrony 客户端向时钟源发起时间同步请求的最小时间间隔，单位为秒。
maxpoll:	chrony 客户端向时钟源发起时间同步请求的最大时间间隔，单位为秒。
maxslewrate:	用于控制 chrony 客户端时钟的最大调整速率，单位为 PPM（百万分之一）。
同时把此服务器作为次级时钟源（可选）	设置后当允许来自白名单中的客户端请求，禁止来自黑名单中的客户端请求前服务器可作为时钟源对外提供服务。

- 自动同步给 OCP 管理（可选）：默认为关闭。开启后需选择已安装的 OCP，如无可选 OCP，则默认置灰。在 OAT 中对服务器完成初始化设置后，自动调用 OCP 的 API 把服务器添加到指定的 OCP 中便于后续使用。

8.2.0.2 注意

- 此处已安装的 OCP 来自于 **产品服务 > 产品 > OCP** 中已安装的 OCP。安装 OCP 可参考 [部署 OCP](#)。
- 使用 OCP 部署 OceanBase 集群前，需要将部署 OceanBase 集群的机器添加到 OCP 资源池。
- 磁盘初始化（可选）：为保障 OceanBase 数据库及相关产品服务的稳定性，通常建议将服务所用的目录配置在独立的磁盘上。选择初始化磁盘后，程序将自动通过 `lvm` 创建对应 `vg`，并将对应大小的 `lv` 挂载到指定的挂载目录上。
 - 选择磁盘或分区：系统会自动识别未挂载的磁盘设备，用户可以选择需要挂载的磁盘设备，支持多选。
 - 挂载用途：分为 **软件安装路径**、**数据盘路径**、**日志盘路径** 和 **docker 根目录**，用户可根据实际要求选择挂载用途。
 - 挂载点：用户根据实际情况，填写相应的挂载路径。
 - 空间（/GB）：根据实际情况填写相应的空间大小，单位为 GB。

5. 单击 **确认**。

- 如果用户打开了磁盘初始化功能，提交时会出现弹窗展示检查项，当检查项不存在报错后，则可点击 **确定**，添加服务器。
- 如果用户未打开磁盘初始化功能，则直接提交，并进行初始化操作。

服务器提交成功后，可以在服务器列表中看到新添加的服务器。此时机器状态为 **初始化中**。用户可以在 **服务器管理** 页面，找到添加的服务器，在对应的 **操作** 列中，点击 **查看任务**，查看任务进度。如果服务器太多，可以在页面上方通过 IP 搜索查询。

如果遇到了报错的子任务，且该子任务不影响业务，可以查看日志信息后，手动将子任务设置为成功。

8.3 后续操作

机器添加成功后，您可以根据自己的需求部署 OCP：

- [部署 OCP](#)

OCP 部署和机器添加成功后，您可以根据自己的需求部署 OceanBase 集群：

- [使用 OCP 部署单机集中式 OceanBase 数据库](#)
- [使用 OCP 部署两副本加仲裁服务的 OceanBase 集群](#)
- [使用 OCP 部署单副本 OceanBase 集群](#)
- [使用 OCP 部署三副本 OceanBase 集群](#)

8.4 相关文档

- 更多添加服务器的信息，请参见 [添加服务器](#)。
- 在服务器完成初始化之后，可以进行该服务器的运维，运维内容包括修改服务器用途、时钟同步配置、磁盘初始化。运维服务器的详细信息，请参见 [运维服务器](#)。

9 OCP 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。

9.1 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。使用 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、网络和软件包等）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。

目前 OCP 支持单节点部署和高可用部署两种部署模式。

9.2 单节点部署

单节点部署是指将 OCP 的所有功能部署在同一个节点中，由该节点提供 OCP 的全部能力。包括：OCP 管理服务（OCP-Server）、元信息数据库（MetaDB）、监控数据库（MonitorDB）、OBProxy。

单节点部署具有如下优势：

节省资源，所有服务都部署在同一台主机上，最大程度的节省主机资源。

方便调试，所有请求都在同一台主机处理，没有分布式处理的问题，便于日志查找。

依赖少，无需依赖负载均衡服务。

9.2.0.1 说明

单节点部署常见于对可用性要求不高或快速验证的场景。

9.3 高可用部署

高可用性部署分为多集群模式和多可用区模式。多集群模式指异地多集群部署，提供跨城 OceanBase 主备库的管理，支持运维跨 OCP 集群的 OceanBase 主备库，包括 OceanBase 主备库的创建、删除、升级、日常切换和容灾切换等，实现双机房高可用容灾架构。

高可用部署情况下，当少数主机或主机上的服务不可用时，整个系统仍然可用。

9.3.0.1 说明

高可用部署常见于生产环境，对系统可用性要求较高。

10 上传安装包

在 OAT 上安装 OceanBase 产品或创建组件，您需要将每个产品的安装包（镜像）上传到 OAT 服务器或远程连接镜像。因此，在使用 OAT 部署 OCP 之前，您需要先上传 OCP 所需的安装包到安装 OAT 的服务器上。

本文介绍如何上传 OCP 所需的安装包至 OAT 服务器。

10.1 操作步骤

10.1.1 步骤一：上传 OCP 安装包到 OAT 的服务器上

在安装 OAT 后，您需要将安装包上传到 `/oat/data_dir/images` 目录中。

对于 OCP 部署，您需要向安装 OAT 的服务器上传两个镜像，包括 OCP docker image 和 MetaDB docker image。

10.1.1.1 注意

如果您是使用的 oat-all-in-one 安装包安装的 OAT，只需上传 OCP docker image。

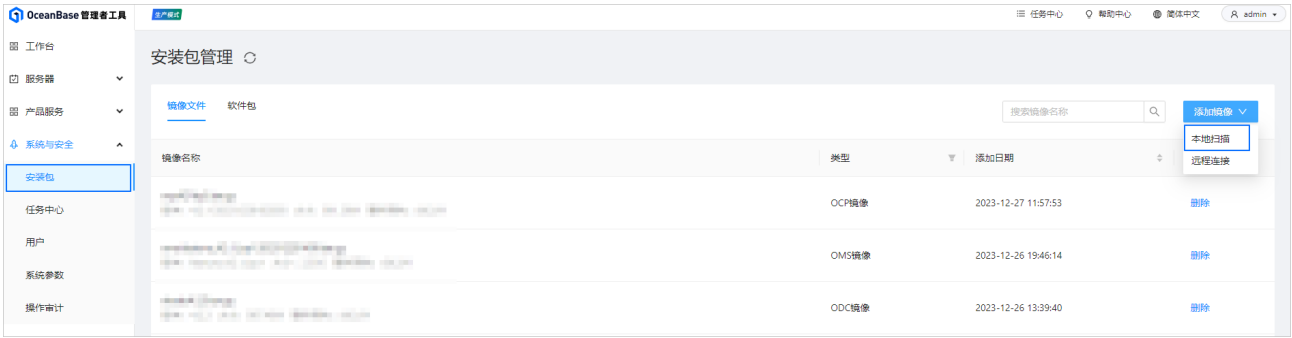
示例如下：

```
[root@hostname /]# scp ./ocp.tar <oat_server>:/oat/data_dir/images/ocp.tar
[root@hostname /]# scp ./meta_xxx.tar <oat_server>:/oat/data_dir/images
/meta_xxx.tar
```

其中 `oat_server` 表示部署 OAT 的服务器的 IP 地址。

10.1.2 步骤二：将安装包添加到 OAT 服务中

1. 登录 OAT。
2. 在左侧导航栏，单击 **系统与安全 > 安装包** 进入 **安装包管理** 页面，然后单击右上角的 **添加镜像**，选择 **本地扫描**。



10.2 相关文档

[安装包管理](#)

11 创建 MetaDB

基于容器部署的 OceanBase 数据库服务，并内置了 OceanBase Proxy，便于外部直接访问。可用于 OCP、ODC 和 OMS 等产品的元数据存储。

本文介绍如何使用 OAT 创建 MetaDB。

11.0.0.1 说明

- 本节以 OAT V4.2.0 版本为例提供操作指导。不同 OAT 版本的操作界面可能不同，请以实际界面为准。
- 推荐在 OceanBase 中创建独立的租户用于 MetaDB。MetaDB 用于存储除了监控数据之外的其它持久化数据，推荐分配 CPU 和内存不少于 4 C，8 GB。
- 磁盘容量受 OCP 管控的集群规模、存储的文件资源和数据保留时间等因素的影响，一般推荐磁盘有 100 GB 以上。

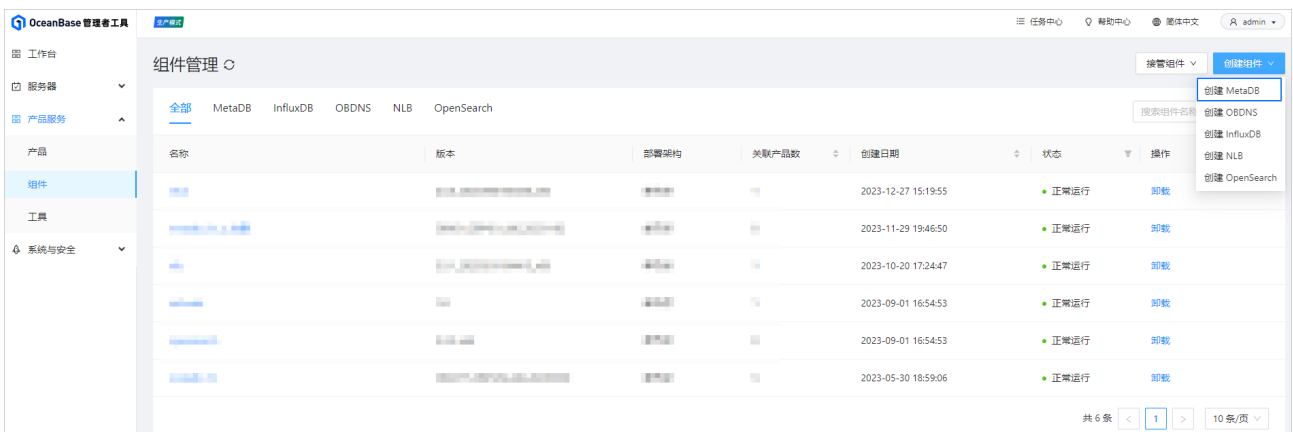
11.1 前提条件

在创建 MetaDB 之前，您需要确认以下信息：

- 您已安装 OAT。详细信息，请参见 [部署 OAT](#)。
- 您已使用超级管理员或运维人员权限的账号登录 OAT。账号的详细信息，请参见 [用户管理](#)。
- 您已添加待安装 MetaDB 的服务器。详细信息，请参见 [添加服务器](#)。

11.2 操作步骤

1. 在左侧导航栏，单击 **产品服务 > 组件** 进入 **组件管理** 页面，然后单击右上角的 **创建组件**，选择 **创建 MetaDB**。



2. 在 **创建 MetaDB** 页面，配置 **基础配置**。

参数	说明
MetaDB 镜像	MetaDB 的 Docker 镜像。
组件名称	组件的名称。不允许与其他组件重名。默认名称为 metadb。
服务器	待安装 MetaDB 的服务器 IP 地址。
自定义 Zone	开启自定义 Zone，需要填写 Zone 名称、机房、服务器、Root Server 位置。 11.2.0.1 注意 此参数与服务器参数二选一，当打开自定义 Zone 后，上述服务器参数会自动隐藏。
配置仲裁服务	可自由开启或者关闭，开启时需要填写仲裁服务地址，示例：xxx.xxx.xxx.xxx:xxxx。 11.2.0.2 注意 当 MetaDB 镜像处选择 V4.X 版本的 MetaDB 时，才能配置仲裁服务。
CPU	MetaDB 在服务器上可用的 CPU，默认为 32C。 11.2.0.3 说明 如果您准备的 OCP 服务器 CPU 和 内存 是 16 C 32 GB 的配置，建议您设置 MetaDB 在服务器上可用的 CPU 为 8 C。
内存	MetaDB 在服务器上可用的内存，默认为 90GB。 11.2.0.4 说明 如果您准备的 OCP 服务器 CPU 和 内存 是 16 C 32 GB 的配置，建议您设置 MetaDB 在服务器上可用的内存为 16 GB。
MetaDB 集群的名称	MetaDB 的集群名称。
root@sys 密码	MetaDB 集群 sys 租户的 root 用户密码。

基础配置

MetaDB 镜像

new_meta_421

版本

OB421_OBP421_x86_20231206

大小

-

硬件架构

x86_64

组件名称

metadb

自定义Zone

Zone名称

请输入

机房 (IDC)

请选择

服务器

请选择服务器

Root Server 位置

请选择

+ 新增Zone

配置仲裁服务

仲裁服务地址

请输入服务地址及端口

CPU

32

C

内存

90

G

MetaDB 集群名称

请输入

root@sys 密码

随机密码

3. 在 创建 MetaDB 页面，配置 安装路径（选填）。

创建 MetaDB 的安装路径配置参数如下：

参数	说明
安装目录	MetaDB 的安装路径，默认值为 <code>/home/admin/oceanbase</code> 。
数据目录	MetaDB 的数据存储路径，默认值为 <code>/data/1</code> 。
日志目录	MetaDB 的日志存储路径，默认值为 <code>/data/log1</code> 。
（可选）备份目录	MetaDB 的备份路径，默认值为空。

√ 安装路径（选填）

安装目录

/home/admin

/oceanbase

数据目录

/data/1

日志目录

/data/log1

备份目录（可选）

请输入

4. 在 创建 MetaDB 页面，配置 启动参数（选填）。

11.2.0.5 注意

如果您准备的 OCP 服务器 CPU 和 内存 是 16 C 32 GB 的配置，并且您设置 MetaDB 在服务器上可用的 CPU 和内存是 8 C 16 GB，您需要在 **observer 进程启动参数** 中设置参数 `system_memory` 的值为 5G，参数 `resource_hard_limit` 的值为 150。

创建 MetaDB 的启动参数配置参数如下：

参数	说明
MySQL 端口	MetaDB 的 MySQL 端口，默认值为 2881。
RPC 端口	MetaDB 的 RPC 端口，默认值为 2882。
(可选) OB 进程启动参数	您可以根据需要修改或添加 observer 进程启动参数。
OB Proxy 服务端口	OBProxy 的服务端口，默认值为 2883。
(可选) OB Proxy进程启动参数	您可以根据需要修改或添加 OBProxy进程启动参数。更多信息，请参见 OBProxy 配置参数说明 。
SSHD 端口	MetaDB 容器内的 SSHD 端口，默认值为 2022。 OCP 通过连接 SSHD 端口部署 OCP Agent。
root 密码	MetaDB 容器内的 root 用户密码。

The screenshot displays the '启动参数 (选项)' (Startup Parameters (Options)) configuration window. It is organized into several sections:

- OB 启动参数 (OB Startup Parameters):** Includes input fields for 'MySQL 端口' (MySQL Port) set to 2881 and 'RPC 端口' (RPC Port) set to 2882.
- OB 进程启动参数 (可选) (OB Process Startup Parameters (Optional)):** A table with columns '参数名' (Parameter Name), '参数值' (Parameter Value), '取值范围 / 参考值' (Value Range / Reference Value), and '操作' (Action). It includes a '+ 添加自定义参数' (Add Custom Parameter) button.
- OB Proxy 启动参数 (OB Proxy Startup Parameters):** Includes an input field for 'OB Proxy 服务端口' (OB Proxy Service Port) set to 2883.
- OB Proxy进程启动参数 (可选) (OB Proxy Process Startup Parameters (Optional)):** Similar to the OB process section, with a table and a '+ 添加自定义参数' button.
- SSHD 启动参数 (SSHD Startup Parameters):** Includes an input field for 'SSHD 端口' (SSHD Port) set to 2022 and a section for 'root 密码' (root password) with a masked input field and a '随机密码' (Random Password) button.

At the bottom right, there are three buttons: '取消' (Cancel), '重置' (Reset), and '提交' (Submit).

5. MetaDB 配置完成后，点击 **提交**。

如果您的配置有误，点击 **重置** 一键恢复默认配置。

11.3 后续操作

安装 MetaDB 后，您可以安装 OceanBase 的生态工具 OCP、ODC、OMS 和 OB Sharding。

11.4 相关文档

[创建和卸载 MetaDB](#)

12 部署 OCP

本文介绍如何使用 OAT 安装 OCP。

12.0.0.1 说明

本节以 OAT V4.2.0 版本为例提供操作指导。不同 OAT 版本的操作界面可能不同，请以实际界面为准。

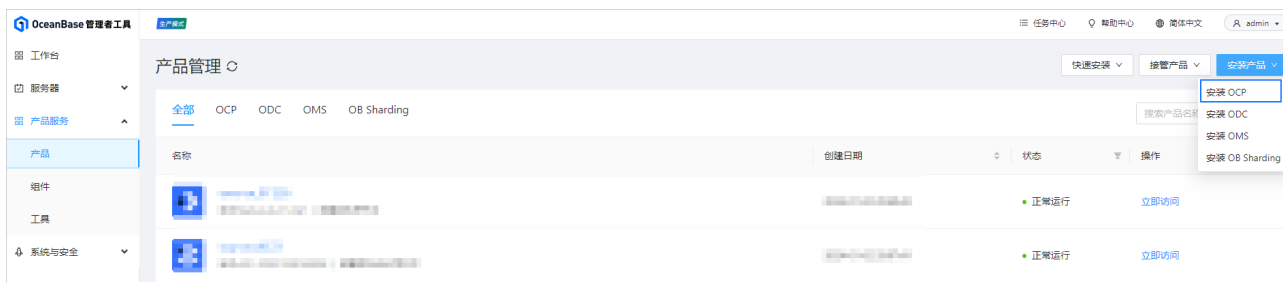
12.1 前提条件

在安装 OCP 之前，您需要确认以下信息：

- 您已安装 OAT。详细信息，请参见 [部署 OAT](#)。
- 您已使用超级管理员或运维人员权限的账号登录 OAT。账号的详细信息，请参见 [用户管理](#)。
- 您已准备安装 OCP 的服务器并且将其添加到 OAT 中。详细信息，请参见 [准备服务器](#) 和 [添加服务器](#)。
- 您已为 OCP 创建元数据库 MetaDB 组件。详细信息，请参见 [创建 MetaDB](#)。
- 您添加 OCP 安装包到 OAT 的服务器上。详细信息，请参见 [上传安装包](#)。

12.2 操作步骤

1. 在左侧导航栏，单击 **产品服务 > 产品** 进入 **产品管理** 页面，然后单击右上角的 **安装产品**，选择 **安装 OCP**。



2. 在 **安装 OCP** 页面，完成 **基础配置**。

参数	说明
OCP 镜像	OCP 的 Docker 镜像。
产品名称	产品名称，不允许与其他产品重名。

管理员密码	OCF 管理员密码。密码格式要求如下： ● 长度为 8~32 个字符。 ● 包含以下四种类型字符至少三种及以上：数字（0~9）、大写字母（A~Z）、小写字母（a~z）、特殊字符 <code>~!@#%^&*_-+=`(){}[]:;'.?/</code>。
多可用区模式	启用多可用区安装时，有以下参数： ● 可用区名称：可用区名称。 ● 服务器：待安装 OCF 的服务器 IP 地址。 ● OCF 代理 IP：此可用区的 OCF 访问入口 IP。 ● OCF 代理端口：此可用区的 OCF 访问入口端口号，默认值为 80。 ● MetaDB 代理 IP：MetaDB 代理 IP 地址。 ● MetaDB 代理端口：MetaDB 代理端口，默认值为 2883。
服务器	待安装 OCF 的服务器 IP 地址。
CPU	OCF 在服务器上可用的 CPU，默认为 8C。 12.2.0.1 注意 如果您准备的 OCF 服务器 CPU 和 内存 是 16 C 32 GB 的配置，并且您设置 MetaDB 在服务器上可用的 CPU 和内存是 8 C 16 GB，您需要设置 OCF 在服务器上可用的 CPU 为 8 C。
内存	OCF 在服务器上可用的内存，默认为 20GB。 12.2.0.2 注意 如果您准备的 OCF 服务器 CPU 和 内存 是 16 C 32 GB 的配置，并且您设置 MetaDB 在服务器上可用的 CPU 和内存是 8 C 16 GB，您需要设置 OCF 在服务器上可用的内存为 16 GB。
HTTP 端口	OCF 的 HTTP 端口号，默认为 8080。
多集群模式	启用多集群安装模式。
日志目录	OCF 的日志存储路径，默认为 <code>/home/admin/logs/ocf</code> 。

基础配置

OCF 镜像

ocp421bp2.tar.gz

版本

4.2.1-20231228162858

大小

961.39M

硬件架构

x86_64

产品名称

请输入

管理员密码

请输入或随机生成

随机生成

多可用区模式

服务器

请选择服务器

CPU

8

C

内存

20

G

HTTP 端口

8080

多集群模式

日志目录

/home/admin/logs

/ocp

3. 在 安装 OCP 页面，完成 MetaDB 配置。

MetaDB 类型有 已创建 MetaDB 和 其他 OB_MySQL 两种方式配置 OCP Meta 租户 和 OCP Monitor 租户：

- 对于新创建的 MetaDB，未创建过租户时，单击 已创建 MetaDB 页签，具体参数参数如下：

参数	说明
选择已创建的 MetaDB	为 OCP 选择已创建的 MetaDB。
是否接管 MetaDB 集群到 OCP	使用 OCP 的接管集群功能添加选择的 MetaDB,完成后可在 OCP 中对其进行管理，但同一个 MetaDB 只能被一个 OCP 接管。
OCF Meta 租户	OCF MetaDB 的专用租户。
租户名	MetaDB 租户名称。
租户密码	MetaDB 租户密码。
CPU	OCF Meta 租户可用的 CPU，默认为 2C。 12.2.0.3 注意 如果您设置 MetaDB 在服务器上可用的 CPU 和内存是 8 C 16 GB，您需要设置 OCP Meta 租户可用的 CPU 为 2 C。

内存	OCP Meta 租户可用的内存，默认为 5GB。 12.2.0.4 注意 如果您设置 MetaDB 在服务器上可用的 CPU 和内存是 8 C 16 GB，您需要设置 OCP Meta 租户可用的内存为 3 GB。
OCP Monitor 租户	OCP MonitorDB 的专用租户。
租户名	MonitorDB 租户名称。
租户密码	MonitorDB 租户密码。
CPU	OCP Monitor 租户可用的 CPU，默认为 4C。 12.2.0.5 注意 如果您设置 MetaDB 在服务器上可用的 CPU 和内存是 8 C 16 GB，您需要设置 OCP Monitor 租户可用的 CPU 为 2 C。
内存	OCP Monitor 租户可用的内存，默认为 16GB。 12.2.0.6 注意 如果您设置 MetaDB 在服务器上可用的 CPU 和内存是 8 C 16 GB，您需要设置 OCP Monitor 租户可用的内存为 3 GB。
接管 MetaDB 集群	勾选以接管 MetaDB 集群。接管后可以在 OCP 中看到此 MetaDB 集群。每个 MetaDB 集群只能被一个 OCP 接管。

MetaDB 配置

MetaDB 类型

已创建 MetaDB

其他 OB_MySQL

选择已创建的 MetaDB

metadb



是否接管 MetaDB 集群到 OCP



是



否

OCP Meta 租户

租户名

请输入

租户密码



随机密码

租户 CPU

2

C

租户 内存

3

G

OCP Monitor 租户

租户名

请输入

租户密码



随机密码

租户 CPU

2

C

租户 内存

3

G

- 已有租户时，单击 **其他 OB_MySQL** 页签，填入 **OCP Meta 租户** 和 **OCP Monitor 租户** 的连接信息。若填写的用户名非 root 的话，对应用户需要有创建数据库表的权限。

MetaDB 类型

已创建 MetaDB

其他 OB_MySQL

OCP meta租户连接信息

主机

端口

请输入

请输入

用户名

密码

请输入

请输入

OCP monitor租户连接信息

主机

端口

请输入

请输入

用户名

密码

请输入

请输入

4. 在 安装 OCP 页面，完成 负载均衡配置。

负载均衡模式有以下四种选择：

- 已创建的 NLB。

负载均衡配置

负载均衡模式

☒ 已创建的 NLB☐ 已创建 OBDNS☐ 其他负载均衡☐ 不使用

① NLB 会根据您填写的 VIP 和 VPORT，自动添加产品实际 HTTP 服务地址的负载均衡规则。另外如果您选择了“已创建 MetaDB”，NLB 还将自动添加到 MetaDB 的负载均衡规则。

选择已创建的 NLB

选择已配置的 VIP

HTTP 服务端口 (VPORT)

- 已创建 OBDNS。

负载均衡配置

负载均衡模式

☐ 已创建的 NLB☒ 已创建 OBDNS☐ 其他负载均衡☐ 不使用

① OBDNS 会根据您填写的 域名 和 VPORT，自动添加到产品实际 HTTP 服务地址的负载均衡规则。另外如果您选择了“已创建 MetaDB”，OBDNS 还将自动添加到 MetaDB 相关服务器的域名解析。

选择已创建的 OBDNS

HTTP 代理域名 ②

HTTP 代理端口 (VPORT) ②

- 其他负载均衡。

负载均衡配置

负载均衡模式

☐ 已创建的 NLB☐ 已创建 OBDNS☒ 其他负载均衡☐ 不使用

i 请将 HTTP 协议代理的负载均衡算法配置为 Hash 类型，避免后端服务会话异常。

MetaDB TCP 代理IP ?MetaDB TCP 代理端口 ?HTTP 代理IP ?HTTP 代理端口 (VPORT) ?

- 不使用：选择不使用负载均衡将无法获得高可用能力，默认使用第一台 IP 作为连接信息。

5. 在 **安装 OCP** 页面，完成 **OpenSearch 配置**。

OpenSearch 配置的模式分为如下三种：

- 已创建的 OpenSearch：用户可以在下拉框内选择通过 OAT 创建的 OpenSearch，关于 OpenSearch 的创建可参考 [创建和卸载 OpenSearch](#)。
- 外部的 OpenSearch：需要填写外部部署的 OpenSearch 的 URL、用户名、密码。
- 不使用。

6. OCP 配置完成后，点击 **提交**。

如果您的配置有误，点击 **重置** 一键恢复默认配置。

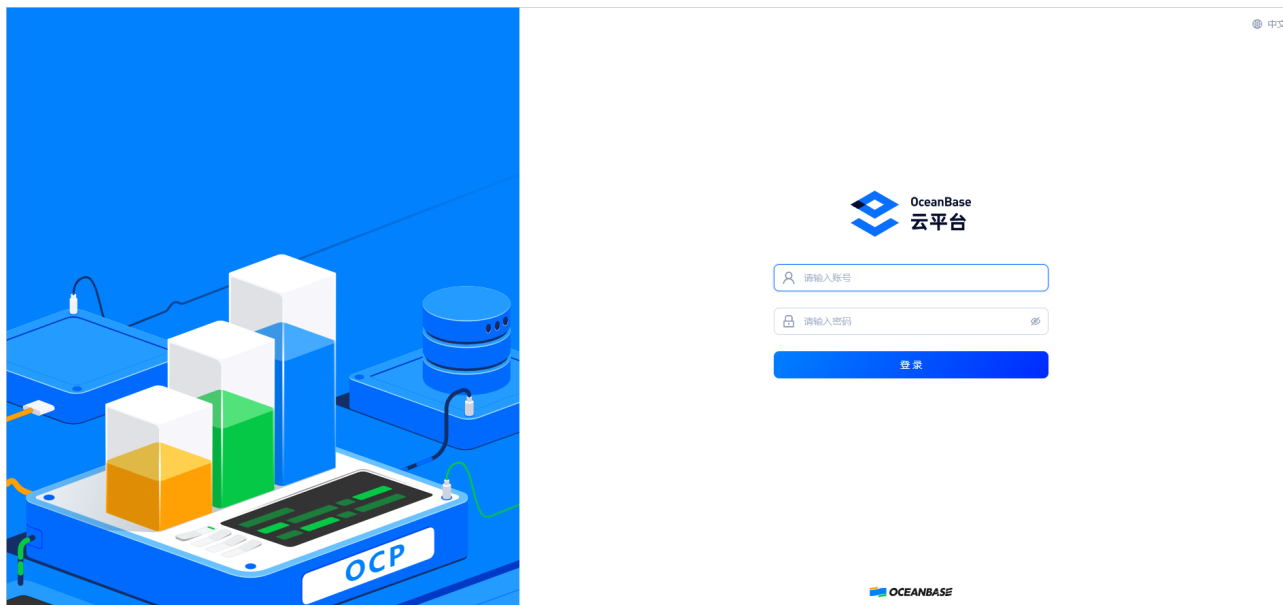
12.3 后续操作

OCP 部署完成后，您可以登录 OCP 进行部署后检查。

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。



12.3.0.1 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。

12.4 相关文档

- [准备服务器](#)。
- [部署 OAT](#)。
- [添加服务器](#)。
- [创建 MetaDB](#)。
- [安装、升级和卸载 OCP](#)

13 使用 OCP 部署单机集中式 OceanBase 数据库

单机集中式是 OceanBase 为用户推出的极简数据库架构，相较于分布式集群，单机集中式仅需一台主机，部署简单，即开即用。但无多副本及扩缩容能力，适用于开发测试以及数据安全要求不高的业务系统。

本文介绍如何使用 OCP 部署单机集中式 OceanBase 数据库。

13.0.0.1 说明

本节以 OCP V4.2.1 版本为例提供操作指导。不同 OCP 版本的操作界面可能不同，请以实际界面为准。仅支持创建 V4.1.0 及以上版本的 OceanBase 数据库。

13.1 前提条件

- 您已部署 OCP。详细信息，请参见 [部署 OCP](#)。
- 您已将部署 OceanBase 集群的服务器添加到 OCP 资源池中。详细信息，请参见 [添加服务器](#)。
- 当前登录 OCP 的用户有 `CREATE CLUSTER` 的权限。详细信息，请参见 [角色概述](#) 和 [OCP 默认角色](#)。
- 已获取 OceanBase 数据库 RPM 包。详细信息，请参见 [准备安装包](#)。

13.2 操作步骤

1. 登录 OCP。
2. 在左侧导航栏上，单击 **集群**。
3. 在 **集群** 页面的右上角，单击 **创建集群**。
4. 在 **创建集群** 页面右上角，单击 **单机集中式**。

5. 在 **创建集群** 页面右上角，设置集群基础信息。

配置	描述
集群名	自定义待管理的集群的名称。集群名称必须以英文字母开头，可支持大小写字母、数字和下划线，长度为 2 ~ 48 字符。
root@sys 密码	支持自定义或随机生成。密码需要满足以下复杂度条件： a. 长度：8 位 ~ 32 位。 b. 至少包含 2 个数字、2 个大写字母、2 个小写字母和 2 个特殊字符。 支持的特殊字符如下：~!@#%^&* _-+=` (){}[]:;'.?/ 此外，您可以单击 复制密码，将自定义或随机生成的密码复制到剪贴板。
OceanBase 版本	可以从列表中选择已有的 OceanBase 数据库版本，也可以在列表下方单击 添加版本 ，上传 OceanBase RPM 版本包。
Zone 名称	可保持默认也可自定义。Zone 名称必须以英文字母开头，支持大小写字母、数字和下划线，长度为 2~48。
主机	选择与 OceanBase 版本 架构一致的主机，仅可选择一台主机作为 OBServer & rootserver。当下拉列表中无可选主机时，可进行 添加主机 的操作。

集群 / 创建集群

< 创建集群 分布式集群 单机集中式

基本设置

集群名 OceanBase 版本

Zone 名称 主机

6. （可选）在 **创建集群** 页面，并配置集群参数。

您可以使用默认参数配置，也可以打开 **参数配置** 模块，并配置集群参数。

- 您可以逐个添加启动参数项并为其配置值。
- 您也可以通过单击 **选择参数模板**，然后选择一个参数模板，系统会将模板中的参数连同配置自动填充到此处。在未创建集群参数模板的情况下，您可单击 **新建集群模板**，为集群创建参数模板。

参数设置 ▾

参数修改 选择参数模板 ▾

参数名	参数值	取值类型	取值范围	操作
请选择 ▾	请输入	-	-	删除
+ 添加启动参数				

7. （可选）在 **创建集群** 页面，通过自定义设置，对路径及端口进行配置。

您可以使用默认路径和端口配置，也可以打开 **自定义设置** 模块，自定义 **路径设置** 和 **端口设置**。

- 配置路径：
 - 在 **路径配置** 模块，根据需要自定义 **软件安装路径**、**数据盘路径** 和 **日志盘路径**。
 - 配置完成后，单击 **测试**，校验路径是否可用。若不可用，您可根据测试结果进行排查，或将路径修改为其它可用路径。
- 配置端口：
 - 在 **端口配置** 模块，自定义 **SQL 端口** 和 **RPC 端口**。
 - 配置完成后，单击 **测试**，校验端口是否已被占用。若被占用，您需重新配置可用端口。

自定义设置 ☒

用户配置

操作系统属主用户 ②

admin

路径设置

软件安装路径

/home/admin /oceanbase

数据盘路径

/data/1

日志盘路径

测试

恢复默认

端口设置

SQL 端口

2881

^
v

RPC 端口

2882

^
v

测试

恢复默认

8. 完成配置后，单击右下角 **提交**。

13.3 后续操作

集群创建成功后，可根据业务需要创建用户租户。

- 使用 OCP 创建用户租户的详细信息，请参见 [新建主租户](#)。
- 使用命令行创建用户租户的详细信息，请参见 [创建租户](#)。

13.4 相关文档

- [部署 OCP](#)
- [添加服务器](#)
- [准备安装包](#)
- [角色概述](#)
- [OCP 默认角色](#)
- [连接方式概述](#)

- [创建单机集中式数据库](#)

14 使用 OCP 部署两副本加仲裁服务的 OceanBase 集群

OceanBase 数据库从 V4.1 版本开始支持了仲裁服务（Arbitration Service），不仅可以解决两地三中心场景下同城副本故障时 RT 变大的问题，还能降低跨城带宽开销，并将第三机房的成本降到极低。

本文介绍如何使用 OCP 部署一个包含两台机器的 OceanBase 集群和一个仲裁成员。

14.0.0.1 说明

- 本节以 OCP V4.2.1 版本为例提供操作指导。不同 OCP 版本的操作界面可能不同，请以实际界面为准。
- 单个 OceanBase 集群只能添加一个仲裁服务。

14.1 前提条件

- 您已部署 OCP。详细信息，请参见 [部署 OCP](#)。
- 您已将部署 OceanBase 集群的服务器添加到 OCP 资源池中。详细信息，请参见 [添加服务器](#)。
- 当前登录 OCP 的用户有 CREATE CLUSTER 的权限。详细信息，请参见 [角色概述](#) 和 [OCP 默认角色](#)。
- 已获取 OceanBase 数据库 V4.1.0 或以上版本的 RPM 包。详细信息，请参见 [准备安装包](#)。

14.2 操作步骤

14.2.1 步骤一：创建两副本集群

1. 登录 OCP。
2. 在左侧导航栏上，单击 **集群**。
3. 在 **集群** 页面右上角，单击 **创建集群**。
4. 在 **创建集群** 页面右上角，使用默认选择 **分布式集群**。
5. 在 **创建集群** 页面，设置集群的基础信息。

基本设置 如下表所示。

配置	描述
集群类型	可选择 主集群 或 备集群 ，此处选择 主集群 。
集群名	自定义待管理的集群的名称。集群名称必须以英文字母开头，可支持大小写字母、数字和下划线，长度为 2 ~ 48 字符。
root@sys 密码	支持自定义或随机生成。密码需要满足以下复杂度条件： a. 长度：8 位 ~ 32 位。 b. 至少包含 2 个数字、2 个大写字母、2 个小写字母和 2 个特殊字符。 支持的特殊字符如下：~!@#%^&* _-+=`(){}[]:;'.?/ 此外，您可以单击 复制密码 ，将自定义或随机生成的密码复制到剪贴板。
OceanBase 版本	可以从列表中选择已有的 OceanBase 数据库版本，也可以在列表下方单击 添加版本 ，上传 OceanBase RPM 版本包。
关联 OBProxy 集群	该选项用于关联已有的 OBProxy 集群，本例中不开启该选项。

基本设置

集群类型

☒ 主集群 ☐ 备集群

集群名

仅支持英文、数字、下划线

root@sys 密码

请输入或随机生成

随机生成

OceanBase 版本

请选择

关联 OBProxy 集群

☒ ☐

仅可选择与此 OceanBase 集群网络一致的 OBProxy 集群

6. 在 **创建集群** 页面，设置集群的部署模式信息。

> 产品版本：V4.2.1

70

默认添加 3 个 Zone 的信息，单击其中一个 Zone 后面的删除图标，保留两个 Zone。

配置	描述
Zone 名称	一般会有一个默认名称，您可以根据需要自定义名称。Zone 名称必须以英文字母开头，可支持大小写字母、数字和下划线，长度为 2 个 ~ 48 个字符。
机房	Zone 所在的机房，每个 Zone 中的主机只能部署在同一个机房。
机型（可选）	如果选择了机型，后面主机列表会根据机型进行过滤。
CPU 架构	可选择 自动分配 或者 手动选择 。
主机	机器的 IP。您可以选择多个 IP。
Root Server 位置	您需选择一个 IP 作为 Root Service 所在的机器。多副本的 OceanBase 集群，其每个 Zone 都需指定一个 Root Service。
Zone 优先级排序	集群 sys 租户的主副本分布的优先级。可选项，如不指定，以第一个 Zone 为第一优先级。最高优先级的 Zone 会被视为 Primary Zone，且只能有一个。 您也可以开启该选项自定义 Zone 的优先级排序，排序方法如下： - 勾选左侧列表框中的一个或多个 Zone。左侧列表框中显示了当前集群中所有可选的 Zone。 - 单击中间的 > 按钮。此时勾选的 Zone 就会被移动到 优先级排序 列表中。同时勾选的多个 Zone 具有相同的优先级。 - 重复 a、b 动作，添加低一优先级的 Zone。 - 如需调整优先级，可在 优先级排序 列表中拖拽以调整顺序。列表中从上到下，优先级依次递减。

部署模式

Zone 名称 机房 机型 (可选) CPU 架构 主机 Root Service 位置

zone1 请选择 请选择 aarch64 请选择 请选择

zone2 请选择 请选择 aarch64 请选择 请选择

+ 新增Zone

Zone 优先级排序 ☒ 同时选择多个 Zone 穿梭到右侧后, 可设置为同一优先级

Zone 优先级排序 恢复为默认

7. (可选) 在 **创建集群** 页面, 并配置集群参数。

您可以使用默认参数配置, 也可以打开 **参数配置** 模块, 并配置集群参数。

- 您可以逐个添加启动参数项并为其配置值。
- 您也可以通过单击 **选择参数模板**, 然后选择一个参数模板, 系统会将模板中的参数连同配置自动填充到此处。在未创建集群参数模板的情况下, 您可单击 **新建集群模板**, 为集群创建参数模板。

参数设置

参数修改 选择参数模板

参数名	参数值	取值类型	取值范围	操作
请选择	请输入	-	-	删除

+ 添加启动参数

8. (可选) 在 **创建集群** 页面, 通过自定义设置, 对路径及端口进行配置。

您可以使用默认路径和端口配置, 也可以打开 **自定义设置** 模块, 自定义 **路径设置** 和 **端口设置**。

- 配置路径:
 - 在 **路径配置** 模块, 根据需要自定义 **软件安装路径**、**数据盘路径** 和 **日志盘路径**。
 - 配置完成后, 单击 **测试**, 校验路径是否可用。若不可用, 您可根据测试结果进行排查, 或将路径修改为其它可用路径。
- 配置端口:
 - 在 **端口配置** 模块, 自定义 **SQL 端口** 和 **RPC 端口**。
 - 配置完成后, 单击 **测试**, 校验端口是否已被占用。若被占用, 您需重新配置可用端口。

自定义设置 ☒

用户配置

操作系统属主用户 ②

admin

路径设置

软件安装路径

/home/admin

/oceanbase

数据盘路径

/data/1

日志盘路径

/data/log1

测试

恢复默认

端口设置

SQL 端口

2881

^

v

RPC 端口

2882

^

v

测试

恢复默认

9. 完成配置后，单击右下角 提交。

14.2.2 步骤二：新建仲裁服务

1. 在左导航栏单击 **集群**，进入集群页面。
2. 在总览页面的右上角，单击 **仲裁服务管理**。
3. 在 **仲裁服务管理** 页面，单击 **新建仲裁服务**。
4. 在 **新建仲裁服务** 面板中，设置仲裁服务的基础信息。

参数	说明
主机	选择部署仲裁服务的主机，同一台主机可以启动多个仲裁服务进程。支持根据 IP 搜索主机，同时支持在下拉栏中单击 添加主机 来新增可用主机。 14.2.2.1 注意 不支持选择已部署 OceanBase 服务的主机。
服务版本	可以从列表中选择已有的 OceanBase 集群版本，所选版本的硬件架构需与 主机 架构保持一致，且只支持 OceanBase V4.1.0 及以上版本的软件包；也可以在列表下方单击 添加版本，上传所需版本的软件包。
软件路径	软件的安装路径地址，默认路径 <code>/home/admin/oceanbase</code>。您需具有主机中该路径的 写 权限。
服务端口	默认为 <code>2882</code>，可根据实际情况进行更改，需确保所选端口未被占用。
说明（可选）	为仲裁服务补充相关说明，可选内容。
参数设置	设置仲裁服务启动参数。 14.2.2.2 注意 为了确保仲裁节点的日志目录能自动清理并防止磁盘写满，需要将 <code>enable_syslog_recycle</code> 设置为 <code>True</code>，并且将 <code>max_syslog_file_count</code> 设置为一个有效值。否则，日志文件会不断增长，可能导致磁盘空间耗尽。

×

新建仲裁服务

基本信息

主机
不支持选择已部署 OceanBase 服务的主机

请选择

服务版本

请选择

操作系统属主用户 [?]

admin

软件路径

/home/admin

/oceanbase

服务端口

2882

说明 (可选)

请输入

参数设置

▼

参数修改

参数名	参数值	取值类型	取值范围	操作
enable_syslog_recycle	FALSE	ENUM	[TRUE,FALSE]	删除
max_syslog_file_count	100	INT	[0,]	删除

+ 添加启动参数

确定

取消

5. 完成配置后，单击右下角 **确定**。可单击弹窗中的任务 ID 查看任务执行状态。

系统会对所填信息进行校验。若校验不通过，您可根据页面提示进行排查或修改。

14.2.3 步骤三：为集群添加仲裁服务

1. 在左导航栏单击 **集群**，进入集群页面。
2. 在 **集群** 页面的 **集群列表** 区域，选择待操作的集群并单击其集群名。
3. 在集群 **概览** 页面的右上角，单击 ... 图标，选择 **添加仲裁服务**。



4. 在 **添加仲裁服务** 面板中，为集群选择仲裁服务地址；也可以在列表下方单击 **新建服务**。



5. 完成选择后，单击右下角 **提交**。

14.3 后续操作

集群创建成功后，可根据业务需要创建用户租户。

- 使用 OCP 创建用户租户的详细信息，请参见 [新建主租户](#)。
- 使用命令行创建用户租户的详细信息，请参见 [创建租户](#)。

14.4 相关文档

- [部署 OCP](#)

- [添加服务器](#)
- [准备安装包](#)
- [角色概述](#)
- [OCP 默认角色](#)
- [连接方式概述](#)
- [新建集群](#)
- [新建仲裁服务](#)
- [添加仲裁服务](#)

15 使用 OCP 部署单副本 OceanBase 集群

OceanBase 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。

本文介绍如何使用 OCP 部署单副本 OceanBase 集群。

15.0.0.1 说明

本节以 OCP V4.2.1 版本为例提供操作指导。不同 OCP 版本的操作界面可能不同，请以实际界面为准。

15.1 前提条件

- 您已部署 OCP。详细信息，请参见 [部署 OCP](#)。
- 您已将部署 OceanBase 集群的服务器添加到 OCP 资源池中。详细信息，请参见 [添加服务器](#)。
- 当前登录 OCP 的用户有 `CREATE CLUSTER` 的权限。详细信息，请参见 [角色概述](#) 和 [OCP 默认角色](#)。
- 已获取 OceanBase 数据库 RPM 包。详细信息，请参见 [准备安装包](#)。

15.2 操作步骤

1. 登录 OCP。
2. 在左侧导航栏上，单击 **集群**。
3. 在 **集群** 页面右上角，单击 **创建集群**。
4. 在 **创建集群** 页面右上角，使用默认选择 **分布式集群**。
5. 在 **创建集群** 页面，设置集群的基础信息。

基本设置 如下表所示。

配置	描述
集群类型	可选择 主集群 或 备集群 ，此处选择 主集群 。
集群名	自定义待管理的集群的名称。集群名称必须以英文字母开头，可支持大小写字母、数字和下划线，长度为 2 ~ 48 字符。
root@sys 密码	支持自定义或随机生成。密码需要满足以下复杂度条件： a. 长度：8 位 ~ 32 位。 b. 至少包含 2 个数字、2 个大写字母、2 个小写字母和 2 个特殊字符。 支持的特殊字符如下：~!@#%^&* _-+=`(){}[]:;'.?/ 此外，您可以单击 复制密码，将自定义或随机生成的密码复制到剪贴板。
OceanBase 版本	可以从列表中选择已有的 OceanBase 数据库版本，也可以在列表下方单击 添加版本 ，上传 OceanBase RPM 版本包。
关联 OBProxy 集群	该选项用于关联已有的 OBProxy 集群，本例中不开启该选项。

基本设置

集群类型

☒ 主集群
 ☐ 备集群

集群名

OceanBase 版本

关联 OBProxy 集群

☒ 是
 ☐ 否

仅可选择与此 OceanBase 集群网络一致的 OBProxy 集群

6. 在 **创建集群** 页面，设置集群的部署模式信息。

默认添加 3 个 Zone 的信息，单击其中两个 Zone 后面的删除图标，保留一个 Zone。

每个 Zone 需要设置的信息及其说明如下表所示。

配置	描述
Zone 名称	一般会有一个默认名称，您可以根据需要自定义名称。Zone 名称必须以英文字母开头，可支持大小写字母、数字和下划线，长度为 2 个 ~ 48 个字符。
机房	Zone 所在的机房，每个 Zone 中的主机只能部署在同一个机房。
机型（可选）	如果选择了机型，后面主机列表会根据机型进行过滤。
CPU 架构	可选择 自动分配 或者 手动选择 。
主机	机器的 IP。您可以选择多个 IP。
Root Server 位置	您需选择一个 IP 作为 Root Service 所在的机器。多副本的 OceanBase 集群，其每个 Zone 都需指定一个 Root Service。
Zone 优先级排序	集群 sys 租户的主副本分布的优先级。可选项，如不指定，以第一个 Zone 为第一优先级。最高优先级的 Zone 会被视为 Primary Zone，且只能有一个。 您也可以开启该选项自定义 Zone 的优先级排序，排序方法如下： - 勾选左侧列表框中的一个或多个 Zone。左侧列表框中显示了当前集群中所有可选的 Zone。 - 单击中间的 > 按钮。此时勾选的 Zone 就会被移动到 优先级排序 列表中。同时勾选的多个 Zone 具有相同的优先级。 - 重复 a、b 动作，添加低一优先级的 Zone。 - 如需调整优先级，可在 优先级排序 列表中拖拽以调整顺序。列表中从上到下，优先级依次递减。

部署模式

Zone 名称: zone1 | 机房: 请选择 | 机型 (可选): 请选择 | CPU 架构: aarch64 | 主机: 请选择 | Root Service 位置: 请选择

+ 新增Zone

Zone 优先级排序: ☒ 同机房选择多个 Zone 穿越到右列后, 可设置为同一优先级

Zone: zone1

优先级排序: 优先级从上到下代表从高到低, 可能随部署 | 恢复为默认

7. (可选) 在 **创建集群** 页面, 并配置集群参数。

您可以使用默认参数配置, 也可以打开 **参数配置** 模块, 并配置集群参数。

- 您可以逐个添加启动参数项并为其配置值。
- 您也可以通过单击 **选择参数模板**, 然后选择一个参数模板, 系统会将模板中的参数连同配置自动填充到此处。在未创建集群参数模板的情况下, 您可单击 **新建集群模板**, 为集群创建参数模板。

参数设置

参数修改 | 选择参数模板

参数名	参数值	取值类型	取值范围	操作
请选择	请输入			删除

+ 添加启动参数

8. (可选) 在 **创建集群** 页面, 通过自定义设置, 对路径及端口进行配置。

您可以使用默认路径和端口配置, 也可以打开 **自定义设置** 模块, 自定义 **路径设置** 和 **端口设置**。

- 配置路径:
 - a. 在 **路径配置** 模块, 根据需要自定义 **软件安装路径**、**数据盘路径** 和 **日志盘路径**。
 - b. 配置完成后, 单击 **测试**, 校验路径是否可用。若不可用, 您可根据测试结果进行排查, 或将路径修改为其它可用路径。
- 配置端口:
 - a. 在 **端口配置** 模块, 自定义 **SQL 端口** 和 **RPC 端口**。
 - b. 配置完成后, 单击 **测试**, 校验端口是否已被占用。若被占用, 您需重新配置可用端口。

自定义设置 ☒

用户配置

操作系统属主用户 

admin

路径设置

软件安装路径

/home/admin

/oceanbase

数据盘路径

/data/1

日志盘路径

/data/log1

测试

恢复默认

端口设置

SQL 端口

2881




RPC 端口

2882




测试

恢复默认

9. 完成配置后，单击右下角 **提交**。

15.3 后续操作

集群创建成功后，可根据业务需要创建用户租户。

- 使用 OCP 创建用户租户的详细信息，请参见 [新建主租户](#)。
- 使用命令行创建用户租户的详细信息，请参见 [创建租户](#)。

15.4 相关文档

- [部署 OCP](#)
- [添加服务器](#)
- [准备安装包](#)
- [角色概述](#)
- [OCP 默认角色](#)
- [连接方式概述](#)
- [新建集群](#)

16 使用 OCP 部署三副本 OceanBase 集群

本文介绍如何使用 OCP 部署三副本 OceanBase 集群。

16.0.0.1 说明

本节以 OCP V4.2.1 版本为例提供操作指导。不同 OCP 版本的操作界面可能不同，请以实际界面为准。

16.1 前提条件

- 您已部署 OCP。详细信息，请参见 [部署 OCP](#)。
- 您已将部署 OceanBase 集群的服务器添加到 OCP 资源池中。详细信息，请参见 [添加服务器](#)。
- 当前登录 OCP 的用户有 `CREATE CLUSTER` 的权限。详细信息，请参见 [角色概述](#) 和 [OCP 默认角色](#)。
- 已获取 OceanBase 数据库 RPM 包。详细信息，请参见 [准备安装包](#)。

16.2 操作步骤

1. 登录 OCP。
2. 在左侧导航栏上，单击 **集群**。
3. 在 **集群** 页面右上角，单击 **创建集群**。
4. 在 **创建集群** 页面右上角，使用默认选择 **分布式集群**。
5. 在 **创建集群** 页面，设置集群的基础信息。

基本设置 如下表所示。

配置	描述
集群类型	可选择 主集群 或 备集群 ，此处选择 主集群 。
集群名	自定义待管理的集群的名称。集群名称必须以英文字母开头，可支持大小写字母、数字和下划线，长度为 2 ~ 48 字符。
root@sys 密码	支持自定义或随机生成。密码需要满足以下复杂度条件： a. 长度：8 位 ~ 32 位。 b. 至少包含 2 个数字、2 个大写字母、2 个小写字母和 2 个特殊字符。 支持的特殊字符如下：~!@#%^&* _-+=`(){}[]:;'.?/ 此外，您可以单击 复制密码 ，将自定义或随机生成的密码复制到剪贴板。
OceanBase 版本	可以从列表中选择已有的 OceanBase 数据库版本，也可以在列表下方单击 添加版本 ，上传 OceanBase RPM 版本包。
关联 OBProxy 集群	该选项用于关联已有的 OBProxy 集群，本例中不开启该选项。

基本设置

集群类型

☒ 主集群 ☐ 备集群

集群名

root@sys 密码

OceanBase 版本

仅支持英文、数字、下划线

请输入或随机生成

随机生成

请选择

关联 OBProxy 集群

☒ ☐

仅可选择与此 OceanBase 集群网络一致的 OBProxy 集群

6. 在 **创建集群** 页面，设置集群的部署模式信息。

默认添加 3 个 Zone 的信息。

每个 Zone 需要设置的信息及其说明如下表所示。

配置	描述
Zone 名称	一般会有一个默认名称，您可以根据需要自定义名称。Zone 名称必须以英文字母开头，可支持大小写字母、数字和下划线，长度为 2 个 ~ 48 个字符。
机房	Zone 所在的机房，每个 Zone 中的主机只能部署在同一个机房。
机型（可选）	如果选择了机型，后面主机列表会根据机型进行过滤。
CPU 架构	可选择 自动分配 或者 手动选择 。
主机	机器的 IP。您可以选择多个 IP。
Root Server 位置	您需选择一个 IP 作为 Root Service 所在的机器。多副本的 OceanBase 集群，其每个 Zone 都需指定一个 Root Service。
Zone 优先级排序	集群 sys 租户的主副本分布的优先级。可选项，如不指定，以第一个 Zone 为第一优先级。最高优先级的 Zone 会被视为 Primary Zone，且只能有一个。 您也可以开启该选项自定义 Zone 的优先级排序，排序方法如下： - 勾选左侧列表框中的一个或多个 Zone。左侧列表框中显示了当前集群中所有可选的 Zone。 - 单击中间的 > 按钮。此时勾选的 Zone 就会被移动到 优先级排序 列表中。同时勾选的多个 Zone 具有相同的优先级。 - 重复 a、b 动作，添加低一优先级的 Zone。 - 如需调整优先级，可在 优先级排序 列表中拖拽以调整顺序。列表中从上到下，优先级依次递减。

部署模式

Zone 名称

机房

机型 (可选)

CPU 架构

主机

Root Service 位置

zone1

请选择

请选择

aarch64

请选择

请选择

zone2

请选择

请选择

aarch64

请选择

请选择

zone3

请选择

请选择

aarch64

请选择

请选择

+ 新增Zone

Zone 优先级排序

☐ Zone

同时选择多个 Zone 穿梭到右侧后，可设置为同一优先级

zone1

zone2

zone3

优先级排序

优先级从上到下代表从高到低，可能抛异常

恢复默认

7. (可选) 在 **创建集群** 页面, 并配置集群参数。

您可以使用默认参数配置，也可以打开 **参数配置** 模块，并配置集群参数。

- 您可以逐个添加启动参数项并为其配置值。
- 您也可以通过单击 **选择参数模板**，然后选择一个参数模板，系统会将模板中的参数连同配置自动填充到此处。在未创建集群参数模板的情况下，您可单击 **新建集群模板**，为集群创建参数模板。

参数设置

参数修改

选择参数模板

参数名	参数值	取值类型	取值范围	操作
请选择	请输入	-	-	删除
+ 添加启动参数				

8. (可选) 在 **创建集群** 页面, 通过自定义设置, 对路径及端口进行配置。

您可以使用默认路径和端口配置，也可以打开 **自定义设置** 模块，自定义 **路径设置** 和 **端口设置**。

- **配置路径：**
 - a. 在 **路径配置** 模块，根据需要自定义 **软件安装路径**、**数据盘路径** 和 **日志盘路径**。
 - b. 配置完成后，单击 **测试**，校验路径是否可用。若不可用，您可根据测试结果进行排查，或将路径修改为其它可用路径。
- **配置端口：**
 - a. 在 **端口配置** 模块，自定义 **SQL 端口** 和 **RPC 端口**。
 - b. 配置完成后，单击 **测试**，校验端口是否已被占用。若被占用，您需重新配置可用端口。

自定义设置 

操作系统属主用户 ?

admin

路径设置

软件安装路径

/home/admin

/oceanbase

数据盘路径

/data/1

日志盘路径

/data/log1

测试

恢复默认

端口设置

SQL 端口

2881

^
v

RPC 端口

2882

^
v

测试

恢复默认

9. 完成配置后，单击右下角 **提交**。

16.3 后续操作

集群创建成功后，可根据业务需要创建用户租户。

- 使用 OCP 创建用户租户的详细信息，请参见 [新建主租户](#)。
- 使用命令行创建用户租户的详细信息，请参见 [创建租户](#)。

16.4 相关文档

- [部署 OCP](#)
- [添加服务器](#)
- [准备安装包](#)
- [角色概述](#)
- [OCP 默认角色](#)
- [连接方式概述](#)
- [新建集群](#)

17（可选）部署 OBProxy

OceanBase 数据库代理（OceanBase Database Proxy，简称：ODP，又称 OBProxy）是 OceanBase 提供的一种数据库代理中间件，它可以用于连接池管理、负载均衡、故障转移等功能。OBProxy 是一个可选的组件，它的部署与否取决于您的应用需求和场景。

本节主要介绍使用 OCP 创建 OBProxy 集群。

17.0.0.1 说明

- 本节以 OCP V4.2.1 版本为例提供操作指导。不同 OCP 版本的操作界面可能不同，请以实际界面为准。
- 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 OBServer 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，需要指定不同端口予以区分。

17.1 前提条件

- 您已部署 OCP。详细信息，请参见 [部署 OCP](#)。
- 当前登录的用户是 OBPROXY_MANAGER 角色，或有管理 OBProxy 的权限。详细信息，请参见 [角色概述](#) 和 [OCP 默认角色](#)。
- 已获取 OBProxy RPM 包。详细信息，请参见 [准备安装包](#)。

17.2 操作步骤

1. 登录 OCP。
2. 在左侧导航栏单击 OBProxy，进入 OBProxy 页面。
3. 在 OBProxy 页面，单击右上方的 **创建 OBProxy 集群**。

4. 在 创建 OBProxy 集群 页面，配置 基本设置。

参数	描述
集群名	您可根据实际业务情况，自定义集群名称，以英文字母开头，可包含英文、数字和下划线。
录入 Proxyro 账号的密码	Proxyro 账号用于 OBProxy 访问 OceanBase 集群，未录入则使用 Proxyro 账号的默认密码进行连接。密码格式要求如下： - 长度为 8 ~ 32 个字符。 - 只能包含字母、数字和特殊字符 <code>~!@#%^&*_-+=\ (){}[]:;'\',./?/。</code> - 大小写字母、数字和特殊字符都至少包含 2 个。
负载均衡管理	默认关闭状态。 - 负载均衡管理关闭状态需配置以下信息： - 访问地址：OBProxy 集群访问地址，仅用于生成租户的连接串，不影响实际使用。如果是 VIP 地址，需要自己申请并绑定到 OBProxy Server。 - 访问端口：默认为 2883，支持自定义端口号。 - 负载均衡管理开启状态需配置以下信息： - 负载均衡器：默认为 OBLB。阿里云 SLB 服务仅适用于专有云环境，建议联系技术支持同学进行协助。 - OBLB 服务：您可选择部署 OCP 时已配置的 OBLB 服务，或单击 添加 OBLB 服务 按钮，在右侧面板中新建 OBLB 服务。 - VIP：选择 OBLB 具体的 VIP 地址。 - 访问端口：默认为 2883，支持自定义端口号。 - 域名配置（可选）：用于指向 VIP 及端口的配置信息，平台未提供 VIP 与域名的映射关系，需自行准备域名解析服务。
启动方式	该 OBProxy 集群的启动方式，可取值： - ConfigUrl：多集群启动方式，即该 OBProxy 集群可访问多个 OceanBase 集群。 - RsList：单集群启动方式，即该 OBProxy 集群仅可访问创建 OBProxy 集群时指定的那个 OceanBase 集群，OBProxy 集群创建成功后不可追加可连接的 OceanBase 集群。

基本设置

集群名

请输入集群名

录入 Proxyro 账号的密码

请输入密码

Proxyro 账号用于 OBProxy 访问 OB 集群，未录入则使用 Proxyro 账号的默认密码进行连接

负载均衡管理

负载均衡器

阿里云 SLB

OBLB

OBLB 服务

请选择

VIP

请选择

访问端口

2883

域名配置 (可选)

请输入

启动方式

ConfigUrl

RsList

选择可连接 OceanBase 集群

请选择集群

仅可选择与此 OBProxy 集群网络一致的 OceanBase 集群

5. （可选）在 创建 OBProxy 集群 页面，配置 部署 OBProxy。

如需在创建 OBProxy 集群时就部署 OBProxy，可在该步骤就行配置。否则可跳过该步骤，待集群创建成功后，通过 接管 OBProxy 或 添加 OBProxy 向集群中添加 OBProxy。

打开配置 部署 OBProxy 的开关。参考下表填写部署信息。

参数	描述
SQL 端口	默认为 2883。
Exporter 端口	默认为 2884。
软件版本	选择要安装的 OBProxy 版本。 17.2.0.1 说明 若 可连接的 OceanBase 集群 中存在 V4.0.0 及以上版本的集群时，仅支持选择 V4.0.0 及以上版本的 OBProxy 软件包。
选择 OBProxy 主机	选择部署 OBProxy 所使用的主机，包括该主机的 机房、机型（可选）、主机。 系统默认显示两个主机的选择，仅够部署两个 OBProxy，如需部署更多，可单击 添加 OBProxy 来增加主机。如只部署 1 个 OBProxy，可单击如上图所示的主机后的删除图标来删除该主机。

部署 OBProxy

SQL 端口

2883

Exporter 端口

2884

软件版本

请选择

仅可选择 1.8.0 及以上版本的软件包

机房

机型 (可选)

主机

请选择

请选择

请选择

请选择

请选择

请选择

+ 添加 OBProxy

6. (可选) 在 创建 OBProxy 集群 页面，配置 参数设置。

打开 参数设置，添加或修改启动参数和其它参数。

参数设置

启动参数

参数名	参数值	取值类型	取值范围	操作
请选择	请输入	-	-	删除
+ 添加启动参数				

7. 完成配置后，单击右下角 提交。

18 使用 oatcli 初始化 OBServer 服务器

oatcli 是管理 OAT 资源的命令行工具，支持使用命令行添加服务器、安装 OceanBase 产品等。

18.1 前提条件

- 您已准备 OBServer 服务器并且已完成相关配置。详细信息，请参见 [准备服务器](#) 和 [服务器配置](#)。
- 您已安装 OAT。详细信息，请参见 [部署 OAT](#)。
- 您拥有 OAT 服务器的 root 用户权限。

18.2 操作步骤

1. 连接到安装 OAT 的服务器。

示例如下：

使用 `ssh` 连接到安装 OAT 的服务器。

```
C:\Users\user001>ssh admin@xxx.xxx.xxx.xxx  
admin@xxx.xxx.xxx.xxx's password:
```

2. 使用下面命令进入到 OAT 容器。

18.2.0.1 说明

执行下面命令，需要 `root` 权限。

```
docker exec -it oat bash
```

示例如下：

```
[admin@xxx /home/admin]  
$sudo docker exec -it oat bash
```

返回结果：

```
[root@xxx oat]#
```

3. （可选）可以使用以下命令查看 oatcli 的使用方法。

```
oatcli --help
```

更多 oatcli 命令信息，请参见 [oatcli](#)。

示例如下：

```
[root@xxx oat]# oatcli --help
```

返回结果：

Usage: oatcli [OPTIONS] COMMAND [ARGS]...

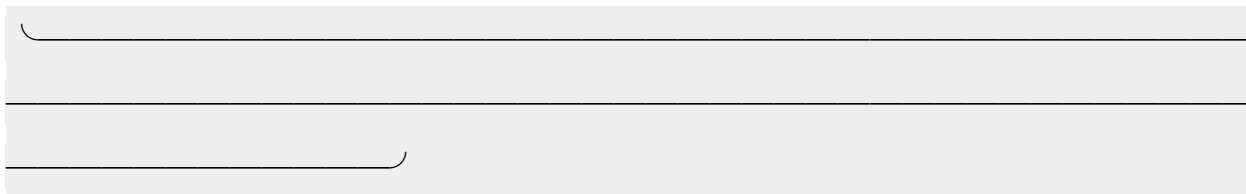
Command line tool for manage OAT resources

Options

```
| --install-completion Install completion for the current shell. |
| --show-completion Show completion for the current shell, to copy it or |
| customize the installation. |
| --help Show this message and exit. |
```

Commands

config	Config OAT API Server address, username and password
create	Create resource in OAT
dagrun	Operate dagrun and task status in OAT
delete	Delete resource in OAT
describe	Get resource detail info in OAT
get	Get resource info in OAT



4. 设置 OAT API 地址、用户名和密码来调用 OAT API。

```
oatcli config set [OPTIONS]
```

选项说明：

- `-s, --api-server TEXT`: `api_server`, 例如: `127.0.0.1:7000`。
- `-u, --username TEXT`: 用户名。
- `-p, --password TEXT`: 密码。
- `--help`: 显示此消息并退出。

示例如下：

```
[root@xxx oat]# oatcli config set -sxxx.xxx.xxx.1:7000 -uadmin -p*****
```

返回结果：

```
11:14:16 [INFO] update {'api_server': 'xxx.xxx.xxx.1:7000', 'username': 'admin',  
'password': '*****'} to /root/.oatcli/config.json
```

5. 创建 机房。

为服务器创建机房信息，方法如下：

```
oatcli create idc [OPTIONS] NAME REGION
```

参数说明：

- `NAME`: 机房名称 [必需]。
- `REGION`: 机房所在地域 [必需]。

选项说明：

- `--help`: 显示此消息并退出。

示例如下：

创建 **机房** 的名称为 `hz0`，地域为 `hangzhou`。

```
[root@xxx oat]# oatcli create idc hz0 hangzhou
```

返回结果：

```
11:20:05 [DEBUG] Starting new HTTP connection (1): xxx.xxx.xxx.1:7000
11:20:05 [DEBUG] http://xxx.xxx.xxx.1:7000 "POST /api/v2/idcs HTTP/1.1" 201 175
```

```
| {'id': 2, 'name': 'hz0', 'region': 'hangzhou'} |
```

6. 创建 凭据。

为服务器连接创建 SSH 凭证，方法如下：

```
oatcli create credential [OPTIONS] NAME AUTH_TYPE:{password|pubkey|external}
```

参数说明：

- **NAME**：凭证标识名称 [必需]。
- **AUTH_TYPE:{password|pubkey|external}**：授权类型 [必需]。
 - **password**：密码认证。
 - **pubkey**：公钥认证。
 - **external**：已配置免密。

选项说明：

- **-u, --username TEXT**：[默认值：root]。
- **-p, --password TEXT**：密码。
- **-k, --key-file PATH**：SSH 私钥文件。

- `--help`: 显示此消息并退出。

示例如下:

创建 **凭据** 的名称为 `root_148` , 用户类型为 `root` 用户, 授权类型为密码认证。

```
[root@xxx oat]# oatcli create credential -uroot -p***** root_148 password
```

返回结果:

```
11:39:17 [DEBUG] Starting new HTTP connection (1): xxx.xxx.xxx.1:7000
11:39:17 [DEBUG] http://xxx.xxx.xxx.1:7000 "POST /api/v2/credentials HTTP/1.1"
200 309
```

```
{
  'id': 1,
  'servers': [],
  'name': 'root_148',
  'authType': 'password',
  'username': 'root',
  'createTime': '2023-06-19T11:39:17.315921+08:00',
  'updateTime': '2023-06-19T11:39:17.315969+08:00'
}
```

7. 添加 OBCServer 服务器（添加并初始化服务器）。

```
oatcli create server [OPTIONS] IPS... INIT_TAG:{observer|obproduct|both}
```

选项说明：

- `-i, --idc TEXT`：机房名称:地域，必须已经存在 [必需]。
- `-c, --credential-name TEXT`：凭证名称，必须已经存在 [必需]。
- `--add-only | --no-add-only`：自定义选项，服务器是否初始化配置 [默认值：no-add-only]。
 - `--add-only`：仅添加服务器，不执行操作系统配置操作。
 - `--no-add-only`：执行操作系统配置操作。
- `--ssh-port INTEGER`：SSH 端口 [默认值：22]。
- `--docker-root-dir TEXT`：[默认路径：/docker]。
- `--admin-uid-gid TEXT`：[默认值：500:500]。
- `--admin-password TEXT`：密码。
- `--help`：显示此消息并退出。

参数说明：

- `IPS...`：服务器 IP [必需]。
- `INIT_TAG:{observer|obproduct|both}`：服务器用途 [必需]。
 - `observer`：安装 OBCServer。
 - `obproduct`：安装 OceanBase 产品服务。
 - `both`：安装 OBCServer 和 OceanBase 产品服务。

示例如下：

添加 **OBCServer** 服务器，使用的 **凭据** 为 `root_148`，使用的 **机房名称:地域** 为 `hz0:hangzhou`，设置 **服务器用途** 为 `observer`。

```
[root@xxx oat]# oatcli create server -ihz0:hangzhou -croot_148 xxx.xxx.xxx.2
observer
```

返回结果：

```
13:56:05 [DEBUG] Starting new HTTP connection (1): xxx.xxx.xxx.1:7000
13:56:06 [DEBUG] http://xxx.xxx.xxx.1:7000 "GET /api/v2/credentials?
name=root_148 HTTP/1.1" 200 389
13:56:06 [DEBUG] Starting new HTTP connection (1): xxx.xxx.xxx.1:7000
13:56:06 [DEBUG] http://xxx.xxx.xxx.1:7000 "GET /api/v2/idcs?
name=hzo&region=hangzhou HTTP/1.1" 200 255
13:56:06 [DEBUG] Starting new HTTP connection (1): xxx.xxx.xxx.1:7000
13:56:06 [DEBUG] http://xxx.xxx.xxx.1:7000 "POST /api/v2/servers/batchCreate
HTTP/1.1" 200 788
```

```
| { |
| 'contents': [ |
| { |
| 'id': 2, |
| 'idcInfo': {'id': 2, 'name': 'hzo', 'region': 'hangzhou'}, |
| 'ip': 'xxx.xxx.xxx.2', |
| 'sshPort': 22, |
| 'checkStatus': 'unknown', |
| 'operateStatus': 'init', |
| 'description': '', |
| 'isOio': False, |
| 'platform': {'system': '', 'node': '', 'release': '', 'version': '', 'machine': ''}, |
| 'hardware': {'cpu': 0, 'memory': 0, 'disk': 0}, |
| 'service': {'product': [], 'component': [], 'observer': []}, |
| 'runtimeInfo': {'dockerRootDir': '/docker', 'adminUidGid': '500:500', 'initTag':
['observer']}, |
```

```
| 'networkInfo': [], |  
| 'createTime': '2023-06-19T13:56:06.839271+08:00', |  
| 'updateTime': '2023-06-19T13:56:06.859580+08:00', |  
| 'relateDagrun': 5, |  
| 'credential': 1 |  
| } |  
| ], |  
| 'page': {'totalElements': 1, 'totalPages': 1, 'number': 1, 'size': 100} |  
| } |
```

8. 查看服务器是否添加完成。

查看服务器详细信息和相关的 dagrun 信息。

```
oatcli describe server [OPTIONS] IP
```

参数说明：

IP：具体服务器 IP [必需]。

选项说明：

--help：显示此消息并退出。

示例如下：

```
[root@xxx oat]# oatcli describe server xxx.xxx.xxx.2
```

返回结果：

```
14:41:01 [DEBUG] Starting new HTTP connection (1): xxx.xxx.xxx.1:7000  
14:41:01 [DEBUG] http://xxx.xxx.xxx.1:7000 "GET /api/v2/servers?ip=xxx.xxx.xxx.  
2 HTTP/1.1" 200 1085  
14:41:01 [DEBUG] Starting new HTTP connection (1): xxx.xxx.xxx.1:7000
```

```
14:41:01 [DEBUG] http://xxx.xxx.xxx.1:7000 "GET /api/v2/servers/2 HTTP/1.1"  
200 998
```

```
{ |  
  'id': 2, |  
  'idcInfo': {'id': 2, 'name': 'hz0', 'region': 'hangzhou'}, |  
  'ip': 'xxx.xxx.xxx.2', |  
  'sshPort': 22, |  
  'checkStatus': 'unknown', |  
  'operateStatus': 'idle', |  
  'description': '', |  
  'isOio': False, |  
  'platform': { |  
    'system': 'Linux', |  
    'node': 'h07g04228.sqa.eu95', |  
    'release': '3.10.0-327.ali2010.rc7.alios7.x86_64', |  
    'version': '#1 SMP Thu Jun 29 21:45:21 CST 2017', |  
    'machine': 'x86_64' |  
  }, |  
  'hardware': {'cpu': 32, 'memory': 125, 'disk': ''}, |  
  'service': {'observer': [], 'product': [], 'component': [], 'tool': []}, |  
  'runtimeInfo': {'dockerRootDir': '/docker', 'adminUidGid': '500:500', 'initTag':  
['observer']}, |  
  'networkInfo': [ |  
    {'nic': 'lo', 'ipWithMask': '127.0.0.1/8'}, |  
    {'nic': 'bond0', 'ipWithMask': 'xxx.xxx.xxx.2/26'}, |
```

```
| {'nic': 'bond0', 'ipWithMask': 'xxx.xxx.xxx.3/26'}, |  
| {'nic': 'docker0', 'ipWithMask': '172.17.0.1/16'} |  
| ], |  
| 'createTime': '2023-06-19T13:56:06.839271+08:00', |  
| 'updateTime': '2023-06-19T14:35:31+08:00', |  
| 'relateDagrun': 5, |  
| 'credential': 1 |  
| } |
```

14:41:01 [DEBUG] Starting new HTTP connection (1): xxx.xxx.xxx.1:7000

14:41:02 [DEBUG] http://xxx.xxx.xxx.1:7000 "GET /api/v2/dagruns/5 HTTP/1.1"
200 2727

14:41:02 [INFO] relate dagrun info:

id	innerState	executeTime	startTime
----	------------	-------------	-----------

check_ssh	success	2023-06-19T05:56:06.842182Z	2023-06-19T05:56:07.970263Z
-----------	---------	-----------------------------	-----------------------------

create_admin_user	success	2023-06-19T05:56:06.842182Z	2023-06-19T05:56:09.060737Z
-------------------	---------	-----------------------------	-----------------------------

config_docker	success	2023-06-19T05:56:06.842182Z	2023-06-19T05:56:10.319578Z
---------------	---------	-----------------------------	-----------------------------

```
| config_deps | success | 2023-06-19T05:56:06.842182Z | 2023-06-19T05:56:10.319575Z |
| config_os | success | 2023-06-19T05:56:06.842182Z | 2023-06-19T05:56:10.319600Z |
| config_chrony | success | 2023-06-19T05:56:06.842182Z | 2023-06-19T05:56:10.319587Z |
| precheck | success | 2023-06-19T05:56:06.842182Z | 2023-06-19T05:56:23.563900Z |
| update_server | success | 2023-06-19T05:56:06.842182Z | 2023-06-19T06:35:31.197149Z |
|_____|_____|_____|_____|
|_____|_____|_____|_____|
|_____|_____|_____|_____|
```

9. 在 OAT 中获取资源信息。

```
oatcli get [OPTIONS] COMMAND [ARGS]...
```

选项说明：

- `--help`：显示此消息并退出。

命令说明：

- `component`：获取组件信息。
- `credential`：获取凭证信息。
- `idc`：获取机房信息。
- `image`：获取镜像信息。
- `product`：获取产品信息。
- `server`：获取服务器信息。

示例如下：

获取服务器列表信息。

```
[root@xxx oat]# oatcli get server
```

返回结果：

```
14:35:50 [DEBUG] Starting new HTTP connection (1): xxx.xxx.xxx.1:7000
14:35:50 [DEBUG] http://xxx.xxx.xxx.1:7000 "GET /api/v2/servers HTTP/1.1" 200
1085
```

id	ip	sshPort	idcInfo	description	operateStatus
2	xxx.xxx.xxx.2	22	{'id': 2, 'name': 'hz0', 'region': 'hangzhou'}		idle

18.3 后续操作

服务器初始化成功后，您可以根据自己的需求部署 OceanBase 集群：

- [使用命令行部署单机集中式 OceanBase 数据库](#)
- [使用命令行部署两副本加仲裁服务的 OceanBase 集群](#)
- [使用命令行部署单副本 OceanBase 集群](#)
- [使用命令行部署三副本 OceanBase 集群](#)

18.4 相关文档

- [准备服务器](#)
- [部署 OAT](#)
- [oatcli](#)

19（可选）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 2 s 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 2 s 以内。本文介绍如何为集群内的多台机器配置 NTP 时钟源。

如果您是单机安装 OceanBase 数据库，则无需设置。

19.1 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。

19.2 操作步骤

按以下步骤配置 NTP 时钟同步。

19.2.0.1 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。

19.2.0.2 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 xxx.xxx.xxx.xxx 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
```

20 使用命令行部署单机集中式 OceanBase 数据库

单机集中式是 OceanBase 为用户推出的极简数据库架构，相较于分布式集群，单机集中式仅仅包含一个 Zone 并且 Zone 中只有一个 OBDServer 节点。但无多副本及扩缩容能力，适用于开发测试以及数据安全要求不高的业务系统。

本文介绍如何使用命令行部署单机集中式 OceanBase 数据库。

20.1 注意事项

如果部署的 OceanBase 数据库要实现资源隔离，那么在部署 OceanBase 数据库之前要先配置 cgroup。

有关资源隔离和配置 cgroup 的更多信息，请参见 [资源隔离概述](#) 和 [配置 cgroup](#)。

20.2 前提条件

在安装 OceanBase 数据库之前，需要确认以下信息：

- OBDServer 服务器已完成相关配置。详细信息，请参见 [服务器配置](#) 和 [使用 oatcli 初始化 OBDServer 服务器](#)。
- 已获取 OceanBase 数据库 RPM 包。详细信息，请参见 [准备安装包](#)。

20.3 操作步骤

20.3.1 步骤一：安装 RPM 包

1. 安装 OceanBase 数据库 RPM 包。

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

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

20.3.1.1 说明

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

示例如下：

```
[root@xxx /home/admin/rpm]# rpm -ivh oceanbase-4.2.0.0-100000052023073123.el7.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
1:oceanbase-4.2.0.0-100000052023073#####
[100%]
```

2. （可选）安装 OceanBase 客户端。

OceanBase 客户端 OBClient 是 OceanBase 数据库专用的命令行客户端工具，通过 OBClient 您可以连接 OceanBase 数据库的 MySQL 租户和 Oracle 租户。如果只连接 OceanBase 数据库的 MySQL 租户，也可以通过 MySQL 客户端（mysql）连接 OceanBase 数据库。

20.3.1.2 注意

由于 OBClient V2.2.1 之前的版本依赖于 `libobclient`，所以需要先安装 `libobclient`。请联系技术支持获取 OBClient RPM 包；如果您使用 OBClient V2.2.1 之前的版本还需获取 `libobclient` RPM 包。

示例如下：

```
[root@xxx /home/admin/rpm]# rpm -ivh obclient-2.2.1-20221122151945.el7.aliases7.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
1:obclient-2.2.1-20221122151945.el7#####
[100%]

##核查是否安装成功##
[root@xxx /home/admin/rpm]# which obclient
/usr/bin/obclient
```

20.3.2 步骤二：配置目录

1. （可选）清理目录。

新机器第一次部署 OceanBase 数据库时不用清理目录。

如果是以下情况选择直接清理旧的 OceanBase 目录：

- 清理之前的 OceanBase 环境。
- 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。

```
[root@xxx /home/admin]# rm -rf /data/1/$cluster_name
[root@xxx /home/admin]# rm -rf /data/log1/$cluster_name
[root@xxx /home/admin]# rm -rf /home/admin/oceanbase/store/$cluster_name
[root@xxx /home/admin]# rm -rf /home/admin/oceanbase/log/* /home/admin
/oceanbase/etc/*config*
```

其中 `$cluster_name` 为集群名。

示例如下：

```
[root@xxx /home/admin]# rm -rf /data/1/obdemo
[root@xxx /home/admin]# rm -rf /data/log1/obdemo
[root@xxx /home/admin]# rm -rf /home/admin/oceanbase/store/obdemo
[root@xxx /home/admin]# rm -rf /home/admin/oceanbase/log/* /home/admin
/oceanbase/etc/*config*
```

2. 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。

20.3.2.3 说明

OceanBase 数据库支持将 `slog` 从数据（data）盘中独立出来，即 `slog` 与数据文件无需在同一个磁盘上。支持配置所使用的磁盘，将 `slog` 与 `clog` 共用一块 SSD 固态硬盘。更多有关 OceanBase 数据库安装目录的信息，请参见 [OBServer 节点安装目录结构](#)。

```
##切换到admin用户##
[root@xxx /home/admin]# su - admin

##使用admin用户执行以下命令##
```

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

其中 `$cluster_name` 为集群名。

示例如下：

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

20.3.2.4 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
```

```
|-- slog -> /data/1/obdemo/slog
|-- sstable -> /data/1/obdemo/sstable
```

6 directories, 0 files

20.3.3 步骤三：初始化 OceanBase 数据库

20.3.3.5 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。

必须在 `admin` 用户下启动 observer 进程，启动语句如下：

```
cd /home/admin/oceanbase && /home/admin/oceanbase/bin/observer {-I $ip |
-i $devname} -P $rpc_port -p $sql_port -z $zone_name -d /home/admin
/oceanbase/store/$cluster_name -r '$ip:2882:2881' -c $cluster_id -n
$cluster_name -o "system_memory=30G,datafile_size=500G,
config_additional_dir=/data/1/$cluster_name/etc3;/data/log1/$cluster_name
/etc2"
```

参数解释：

参数	说明
<code>-I</code> <code>-i</code>	<code>-I</code>：指定待启动的节点 IP。在多机部署场景下，不能指定 127.0.0.1 作为目标 IP。建议使用指定 IP（例如：<code>-I 10.10.10.1</code>）来启动节点。 <code>-i</code>：指定网卡名，可通过 <code>ifconfig</code> 命令查看。 20.3.3.6 说明 支持通过同时指定 IP 和网卡名（例如：<code>-I 10.10.10.1 -i eth0</code>）来启动节点，但不建议这么用。
<code>-p</code>	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，初始化目录时创建的目录。除集群名字 <code>\$cluster_name</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 Root Service 信息。
	指定集群启动参数（配置项）列表，为可选项。可以为多个配置项指定值，多个配置项的值之间使用英文逗号进行分隔。请根据实际需要，选择适当的启动参数并为每个参数指定合适的值，以优化集群的性能和资源利用率。以下是一些常用的集群启动配置项：

-O

- [cpu_count](#): 用于设置系统 CPU 总数。
- [system_memory](#): 用于设置系统预留给租户 ID 为 500 的租户的内存容量，即指定 OceanBase 数据库内部保留内存。在机器内存较少的情况下可以考虑将该值调小，需要注意的是，在性能测试时可能会出现内存不足的问题。
- [memory_limit](#): 用于设置可用的总内存大小。
- [datafile_size](#): 用于设置数据文件占用磁盘可用空间的大小。即指定 OceanBase 数据库数据文件 `sstable` 的大小（一次性初始化），根据 `/data/1/` 可用空间评估，建议不少于 100G。
- [datafile_disk_percentage](#): 磁盘数据文件占用磁盘总空间的百分比。
- [datafile_next](#): 用于设置磁盘数据文件自动扩容的步长。
- [datafile_maxsize](#): 用于设置磁盘数据文件自动扩容的最大空间。
- [config_additional_dir](#): 用于设置本地存储配置文件的多个目录，为了冗余存储多份配置文件。
- [log_disk_size](#): 用于设置 Redo 日志磁盘的大小。
- [log_disk_percentage](#): 用于设置 Redo 日志占用其所在磁盘总空间的百分比。
- [syslog_level](#): 用于设置系统日志级别。
- [syslog_io_bandwidth_limit](#): 用于设置系统日志所能占用的磁盘 IO 带宽上限，超过带宽上限容量的系统日志将被丢弃。
- [max_syslog_file_count](#): 用于设置在回收日志文件之前可以容纳的日志文件数量。
- [enable_syslog_recycle](#): 用于设置是否开启回收系统日志的功能。

上述配置项中的 `datafile_size`、`datafile_disk_percentage`、`datafile_next` 和 `datafile_maxsize` 配合使用可以达到磁盘数据文件的自动扩容的目的，详细信息请参见 [配置磁盘数据文件的动态扩容](#)。更多集群配置信息，请参见 [配置项总览-集群级配置项](#)。

示例如下：

```
[root@xxx /home/admin]# su - admin
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
/observer -l 10.10.10.1 -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"
```

可以通过下面命令查看 observer 进程是否启动成功：

- `netstat -ntlp` 命令，如果监听到了 2881 与 2882 端口，则说明进程启动成功。
- `ps -ef|grep observer` 命令可以返回 observer 进程信息。

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 11111/observer
tcp 0 0 0.0.0.0:2882 0.0.0.0:* LISTEN 11111/observer
... ..

-bash-4.2$ ps -ef|grep observer
admin 11111 0 43 17:58 ? 00:00:14 /home/admin/oceanbase/bin/observer -l
10.10.10.1 -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
```

2. 集群 bootstrap 操作。

通过 OBClient 客户端连接已启动的 observer 进程，密码为空。

```
[root@xxx /home/admin]# obclient -h127.0.0.1 -uroot -P2881 -p*****

obclient [(none)]> SET SESSION ob_query_timeout=1000000000;
Query OK, 0 rows affected

obclient [(none)]> ALTER SYSTEM BOOTSTRAP ZONE 'zone1' SERVER '10.10.10.1:
2882';
Query OK, 0 rows affected
```

20.3.3.7 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 节点相关目录权限不对、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。

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` 库即可。

```
obclient [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| LBACSYS |
| mysql |
| oceanbase |
| ORAAUDITOR |
| SYS |
| 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 [(none)]> ALTER USER root IDENTIFIED BY '*****';
```

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

20.4 后续操作

集群创建成功后，可根据业务需要创建用户租户。

使用命令行创建用户租户的详细信息，请参见 [创建租户](#)。

20.5 相关文档

- [通过 OBClient 连接 OceanBase 租户](#)
- [系统配置项总览](#)
- [ALTER USER](#)
- [用户租户介绍](#)
- [CREATE TENANT](#)
- [单机在线转分布式部署](#)

21 使用命令行部署两副本加仲裁服务的 OceanBase 集群

OceanBase 数据库从 V4.1 版本开始支持仲裁服务（Arbitration Service），不仅可以解决两地三中心场景下同城副本故障时 RT 变大的问题，还能降低跨城带宽开销，并将第三机房的成本降到极低。

本节主要介绍使用命令行部署一个包含两台机器的 OceanBase 集群和一个仲裁服务。

21.0.0.1 说明

- 单个 OceanBase 集群最多只能使用一个仲裁服务。
- 仲裁服务目前只支持单机部署。

21.1 注意事项

如果部署的 OceanBase 数据库要实现资源隔离，那么在部署 OceanBase 数据库之前要先配置 cgroup。

有关资源隔离和配置 cgroup 的更多信息，请参见 [资源隔离概述](#) 和 [配置 cgroup](#)。

21.2 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 [服务器配置](#)、[（可选）配置时钟源](#) 和 [使用 oatcli 初始化 OBServer 服务器](#)。
- 已获取 OceanBase 数据库 RPM 包。详细信息，请参见 [准备安装包](#)。

21.3 操作步骤

21.3.1 步骤一：部署 OceanBase 集群

1. 在两台 OBServer 服务器中分别安装 RPM 包。

a. 安装 OceanBase 数据库 RPM 包。

切换至存放 OceanBase 数据库 RPM 包的目录：

```
[root@xxx /]# cd $rpm_dir
```

安装 OceanBase 数据库 RPM 包：

```
[root@xxx $rpm_dir]# rpm -ivh $rpm_name [--prefix=/home/admin  
/oceanbase]
```

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

21.3.1.1 说明

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

示例如下：

```
[root@xxx /home/admin/rpm]# rpm -ivh oceanbase-4.2.0.0-  
100000052023073123.el7.x86_64.rpm  
Preparing... ##### [100%]  
Updating / installing...  
1:oceanbase-4.2.0.0-  
100000052023073##### [100%]
```

b. （可选）安装 OceanBase 客户端。

OceanBase 客户端 OBClient 是 OceanBase 数据库专用的命令行客户端工具，通过 OBClient 您可以连接 OceanBase 数据库的 MySQL 租户和 Oracle 租户。如果只连接 OceanBase 数据库的 MySQL 租户，也可以通过 MySQL 客户端（mysql）连接 OceanBase 数据库。

21.3.1.2 注意

由于 `obclient` V2.2.1 之前的版本依赖于 `libobclient`，所以需要先安装 `libobclient`。请联系技术支持获取 OBClient RPM 包和 `libobclient` RPM 包。

示例如下：

```
[root@xxx /home/admin/rpm]# rpm -ivh obclient-2.2.1-20221122151945.el7.  
alias7.x86_64.rpm  
Preparing... ##### [100%]  
Updating / installing...  
1:obclient-2.2.1-20221122151945.  
el7##### [100%]
```

核查是否安装成功：

```
[root@xxx /home/admin/rpm]# which obclient
/usr/bin/obclient
```

2. 配置 observer 进程目录。

a. （可选）清理目录。

新机器第一次部署 OceanBase 数据库时不用清理目录。

如果是以下情况选择直接清理旧的 OceanBase 目录：

- 清理之前的 OceanBase 环境。
- 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。

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

其中 `$cluster_name` 为集群名。

示例如下：

```
[root@xxx /home/admin]# kill -9 `pidof observer`
[root@xxx /home/admin]# rm -rf /data/1/obdemo
[root@xxx /home/admin]# rm -rf /data/log1/obdemo
[root@xxx /home/admin]# rm -rf /home/admin/oceanbase/store/obdemo
[root@xxx /home/admin]# rm -rf /home/admin/oceanbase/log/* /home
/admin/oceanbase/etc/*config*
```

b. 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。

21.3.1.3 说明

OceanBase 数据库支持将 `slog` 从数据（data）盘中独立出来，即 `slog` 与数据文件无需在同一个磁盘上。支持配置所使用的磁盘，将 `slog` 与 `clog` 共用一块 SSD 固态硬盘。更多有关 OceanBase 数据库安装目录的信息，请参见 [OBServer 节点安装目录结构](#)。

使用下面语句切换到 `admin` 用户：

```
[root@xxx /home/admin]# su - admin
```

使用 `admin` 用户执行以下命令创建相关目录：

```
mkdir -p /data/1/$cluster_name/{etc3,sstable,slog}
```

```
mkdir -p /data/log1/$cluster_name/{clog,etc2}
```

```
mkdir -p /home/admin/oceanbase/store/$cluster_name
```

使用 `admin` 用户执行以下命令创建软连接：

```
for t in {etc3,sstable,slog};do ln -s /data/1/$cluster_name/$t /home/admin/oceanbase/store/$cluster_name/$t; done
```

```
for t in {clog,etc2};do ln -s /data/log1/$cluster_name/$t /home/admin/oceanbase/store/$cluster_name/$t; done
```

其中 `$cluster_name` 为集群名。

示例如下：

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

```
-bash-4.2$ for t in {clog,etc2};do ln -s /data/log1/obdemo/$t /home/admin/oceanbase/store/obdemo/$t; done
```

21.3.1.4 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
|  |-- slog -> /data/1/obdemo/slog
|  |-- sstable -> /data/1/obdemo/sstable
6 directories, 0 files
```

3. 初始化 OceanBase 集群。

21.3.1.5 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。

a. 启动 observer 进程。

在每个节点的 `admin` 用户下，启动 observer 进程。

21.3.1.6 注意

两个副本下，每个节点启动参数并不完全相同。启动 `observer` 时只需指定 `Root Service` 所在的 2 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。

```
cd /home/admin/oceanbase && /home/admin/oceanbase/bin/observer {-l  
$ip | -i $devname} -P $rpc_port -p $sql_port -z $zone_name -d /home/admin  
/oceanbase/store/$cluster_name -r '$ip:2882:2881' -c $cluster_id -n  
$cluster_name -o "system_memory=30G,datafile_size=500G,  
config_additional_dir=/data/1/$cluster_name/etc3;/data/log1  
/$cluster_name/etc2"
```

参数解释：

参数	说明
<code>-l -i</code>	• <code>-l</code>：指定待启动的节点 IP。在多机部署场景下，不能指定 127.0.0.1 作为目标 IP。建议使用指定 IP（例如：<code>-l 10.10.10.1</code>）来启动节点。 • <code>-i</code>：指定网卡名，可通过 <code>ifconfig</code> 命令查看。 21.3.1.7 说明 支持通过同时指定 IP 和网卡名（例如：<code>-l 10.10.10.1 -i eth0</code>）来启动节点，但不建议这么用。
<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 observer 进程所属的 Zone。
<code>-d</code>	指定集群主目录，初始化目录时创建的目录。除集群名字 <code>\$cluster_name</code> 外，其他不要变动。
<code>-c</code>	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。
<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 Root Service 信息。

-O

指定集群启动参数（配置项）列表，为可选项。可以为多个配置项指定值，多个配置项的值之间使用英文逗号进行分隔。请根据实际需要，选择适当的启动参数并为每个参数指定合适的值，以优化集群的性能和资源利用率。以下是一些常用的集群启动配置项：

- [cpu_count](#)：用于设置系统 CPU 总数。
- [system_memory](#)：用于设置系统预留租户 ID 为 500 的租户的内存容量，即指定 OceanBase 数据库内部保留内存。在机器内存较少的情况下可以考虑将该值调小，需要注意的是，在性能测试时可能会出现内存不足的问题。
- [memory_limit](#)：用于设置可用的总内存大小。
- [datafile_size](#)：用于设置数据文件占用磁盘可用空间的大小。即指定 OceanBase 数据库数据文件 sstable 的大小（一次性初始化），根据 /data/1/ 可用空间评估，建议不少于 100G。
- [datafile_disk_percentage](#)：磁盘数据文件占用磁盘总空间的百分比。
- [datafile_next](#)：用于设置磁盘数据文件自动扩容的步长。
- [datafile_maxsize](#)：用于设置磁盘数据文件自动扩容的最大空间。
- [config_additional_dir](#)：用于设置本地存储配置文件的多个目录，为了冗余存储多份配置文件。
- [log_disk_size](#)：用于设置 Redo 日志磁盘的大小。
- [log_disk_percentage](#)：用于设置 Redo 日志占用其所在磁盘总空间的百分比。
- [syslog_level](#)：用于设置系统日志级别。
- [syslog_io_bandwidth_limit](#)：用于设置系统日志所能占用的磁盘 IO 带宽上限，超过带宽上限容量的系统日志将被丢弃。
- [max_syslog_file_count](#)：用于设置在回收日志文件之前可以容纳的日志文件数量。
- [enable_syslog_recycle](#)：用于设置是否开启回收系统日志的功能。

上述配置项中的 `datafile_size`、`datafile_disk_percentage`、`datafile_next` 和 `datafile_maxsize` 配合使用可以达到磁盘数据文件的自动扩容的目的，详细信息请参见 [配置磁盘数据文件的动态扩容](#)。更多集群配置信息，请参见 [配置项总览-集群级配置项](#)。

示例如下：

在 OBCServer 服务器中启动 observer 进程，以 10.10.10.1 和 10.10.10.2 为例，启动 observer 进程。

zone1:

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

可以通过下面命令查看 observer 进程是否启动成功：

- `netstat -ntlp` 命令，如果监听到了 2881 与 2882 端口，则说明进程启动成功。
- `ps -ef|grep observer` 命令可以返回 observer 进程信息。

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 11112/observer
tcp 0 0 0.0.0.0:2882 0.0.0.0:* LISTEN 11112/observer
... ..

-bash-4.2$ ps -ef|grep observer
admin 11112 0 40 16:18 ? 00:00:17 /home/admin/oceanbase/bin/observer -l
10.10.10.1 -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 -c 10001 -n obdemo -
o system_memory=30G,datafile_size=500G,config_additional_dir=/data/1
/obdemo/etc3;/data/log1/obdemo/etc2
```

b. 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 -h127.0.0.1 -uroot -P2881 -p*****

obclient [(none)]> SET SESSION ob_query_timeout=1000000000;
Query OK, 0 rows affected

obclient [(none)]> ALTER SYSTEM BOOTSTRAP ZONE 'zone1' SERVER
'10.10.10.1:2882',ZONE 'zone2' SERVER '10.10.10.2:2882';
Query OK, 0 rows affected
```

21.3.1.8 注意

如果这一步失败报错了，其原因很可能就是 `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 目录从头开始部署。

c. 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 [(none)]> SHOW DATABASES;
```

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

d. 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 [(none)]> ALTER USER root IDENTIFIED BY '*****';  
Query OK, 0 rows affected
```

21.3.2 步骤二：部署仲裁服务

1. 在仲裁 Server 服务器中安装 OceanBase RPM 包。

切换至存放 OceanBase 数据库 RPM 包的目录：

```
[root@xxx /]# cd $rpm_dir
```

安装 OceanBase 数据库 RPM 包：

```
[root@xxx $rpm_dir]# rpm -ivh $rpm_name [--prefix=/home/admin/oceanbase]
```

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

21.3.2.9 说明

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

示例如下：

```
[root@xxx /home/admin/rpm]# rpm -ivh oceanbase-4.2.0.0-100000052023073123.el7.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
1:oceanbase-4.2.0.0-100000052023073#####
[100%]
```

2. 配置仲裁 server 进程目录

在仲裁 Server 服务器中创建仲裁 server 进程的 `clog` 目录。

21.3.2.10 说明

- 启动仲裁 server 进程时，`-d` 指定的 `store` 目录中需要确保提前创建好 `clog` 目录。
- 需要避免 `store` 目录所在的磁盘空间被完全占满，否则会导致仲裁 Server 的功能出现异常。

使用下面命令切换到 `admin` 用户：

```
[root@xxx /home/admin]# su - admin
```

使用 `admin` 用户执行以下命令创建相关目录：

```
mkdir -p /home/admin/oceanbase/store/clog
```

3. 启动仲裁 server 进程。

在仲裁 Server 服务器的 `admin` 用户下，以仲裁模式启动仲裁 server 进程，指定必要的启动参数。启动语句如下：

```
cd /home/admin/oceanbase && /home/admin/oceanbase/bin/observer -m
arbitration -P $rpc_port -d $obdir/store [{"-l $ip | -i $devname}] [-l $log_level] [-o
"parameters_list"]
```

参数解释：

参数	说明
-m	指定仲裁模式，取值为 <code>arbitration</code> 。
-P	指定仲裁 server 进程的 RPC 端口号，一般指定为 <code>2882</code> 。
-d	指定存储目录路径。通常为 <code>/home/admin/oceanbase/store</code> ，需确保该目录下已创建好 <code>clog</code> 子目录。
-l -i	可选项。 • <code>-l</code>：指定待启动的节点 IP。 • <code>-i</code>：指定网卡名，可通过 <code>ifconfig</code> 命令查看。 21.3.2.11 说明 支持通过同时指定 IP 和网卡名（例如：<code>-l 10.10.10.1 -i eth0</code>）来启动节点，但不建议这么用。
-l	指定日志级别。可选项，默认值是 <code>WDIAG</code> 。取值范围：{ <code>DEBUG</code> , <code>TRACE</code> , <code>WDIAG</code> , <code>EDIAG</code> , <code>INFO</code> , <code>WARN</code> , <code>ERROR</code> }，有关日志的详细信息，请参见 日志级别 。

-o

指定配置项列表，可选项。可为多个配置项指定值，中间用英文逗号分隔。以下是启动仲裁 server 进程常用的配置项：

- [cpu_count](#)
- [system_memory](#)
- [memory_limit](#)
- [__easy_memory_limit](#)
- [log_disk_size](#)
- [log_disk_percentage](#)
- [syslog_level](#)
- [syslog_io_bandwidth_limit](#)
- [max_syslog_file_count](#)
- [enable_syslog_recycle](#)

21.3.2.12 注意

为了确保仲裁节点的日志目录能自动清理并防止磁盘写满，需要将 `enable_syslog_recycle` 设置为 `True`，并且将 `max_syslog_file_count` 设置为一个有效值。否则，日志文件会不断增长，可能导致磁盘空间耗尽。

示例如下：

在仲裁 Server 服务器中启动仲裁 server 进程，以 `10.10.10.3` 为例，启动仲裁 server 进程。

```
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
/observer -m arbitration -P 2882 -d /home/admin/oceanbase/store -l 10.10.10.3
-o "enable_syslog_recycle=True,max_syslog_file_count=100"
```

可以通过下面命令查看 observer 进程是否启动成功：

- `netstat -ntlp` 命令，如果监听到了 2882 端口，则说明进程启动成功。
- `ps -ef|grep observer` 命令可以返回 observer 进程信息。

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 0.0.0.0:2882 0.0.0.0:* LISTEN 18231/observer
...

-bash-4.2$ ps -ef|grep observer
admin 18231 0 3 10:33 ? 00:00:00 /home/admin/oceanbase/bin/observer -m
arbitration -P 2882 -d /home/admin/oceanbase/store -l 10.10.10.3
...
```

21.3.3 步骤三：为集群添加仲裁服务

为集群添加仲裁服务，命令如下：

```
ALTER SYSTEM ADD ARBITRATION SERVICE "$arb_server_ip:$arb_server_port";
```

21.3.3.13 说明

仲裁 server 进程地址将被记录到 sys 租户的一张内部表中，可以通过视图 DBA_OB_ARBITRATION_SERVICE 查看，该指令为同步操作。

示例如下：

```
obclient [(none)]> ALTER SYSTEM ADD ARBITRATION SERVICE "10.10.10.3:2882";
Query OK, 0 rows affected

obclient [(none)]> SELECT * FROM oceanbase.DBA_OB_ARBITRATION_SERVICE;
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| CREATE_TIME | MODIFY_TIME | ARBITRATION_SERVICE_KEY | ARBITRATION_SERVICE |
| PREVIOUS_ARBITRATION_SERVICE | TYPE |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

```
| 2023-03-06 17:29:36 | 2023-03-06 17:29:36 | default | 10.10.10.3:2882 | NULL | ADDR |
+-----+-----+-----+
+-----+-----+-----+
1 row in set
```

21.3.4 步骤四：为 sys 租户开启仲裁服务

可以在创建租户时即为租户开启仲裁服务，新建租户时为租户开启仲裁服务的详细信息，请参见 [CREATE TENANT](#)。

如果在创建租户时未开启仲裁服务，也可以待租户创建成功后，为租户开启仲裁服务。

执行以下语句给已有租户开启仲裁服务。

```
ALTER TENANT $tenant_name [SET] enable_arbitration_service = true;
```

示例如下：

给集群的 `sys` 租户开启仲裁服务。

```
obclient [(none)]> ALTER TENANT sys SET enable_arbitration_service = true;
Query OK, 0 rows affected
```

21.3.5 步骤五：查询租户仲裁服务状态

1. 查看租户的仲裁服务是否开启。

可以通过视图 `DBA_OB_TENANTS` 查询租户的信息，通过 `arbitration_service_status` 这一列确认仲裁服务的开启状态。

`arbitration_service_status` 取值如下：

- `ENABLED`：表示该租户已开启仲裁服务。
- `ENABLING`：表示该租户仲裁服务开启过程中。
- `DISABLED`：表示该租户已关闭仲裁服务。
- `DISABLING`：表示该租户仲裁服务关闭过程中。

示例如下：

```
obclient [(none)]> SELECT TENANT_ID,TENANT_NAME,
ARBITRATION_SERVICE_STATUS FROM oceanbase.DBA_OB_TENANTS WHERE
tenant_name = 'sys';
+-----+-----+-----+
| TENANT_ID | TENANT_NAME | ARBITRATION_SERVICE_STATUS |
+-----+-----+-----+
| 1 | sys | ENABLED |
+-----+-----+-----+
1 row in set
```

2. 查看当前仲裁服务是否可用。

租户的仲裁服务开启成功表示该租户在开启时刻已创建的日志流都可以使用仲裁服务，但开启之后新创建的日志流可能缺少仲裁服务，因为创建日志流是非严格模式执行的，不保证仲裁服务一定创建成功，对于这种情况，用户可通过如下 SQL 语句确认该租户所有日志流是否都有仲裁服务。

##查询租户内没有仲裁副本的日志流列表##

```
(SELECT distinct ls_id FROM GV$OB_LOG_STAT WHERE tenant_id = xxx) except
(SELECT ls_id FROM GV$OB_LOG_STAT WHERE tenant_id = xxx and role = 'LEADER'
and arbitration_member = "$arb_server_ip:$arb_server_port");
```

示例如下：

查询 `sys` 租户内没有仲裁副本的日志流列表，返回空集说明当前租户的仲裁服务是可用状态。

```
obclient [oceanbase]> (SELECT distinct ls_id FROM GV$OB_LOG_STAT WHERE
tenant_id = 1) except
(SELECT ls_id FROM GV$OB_LOG_STAT WHERE tenant_id = 1 and role = 'LEADER'
and arbitration_member = "10.10.10.3:2882");
Empty set
```

21.4 后续操作

集群创建成功后，可根据业务需要创建用户租户。

使用命令行创建用户租户的详细信息，请参见 [创建租户](#)。

21.5 相关文档

- [通过 OBClient 连接 OceanBase 租户](#)
- [系统配置项总览](#)
- [ALTER USER](#)
- [ALTER TENANT](#)
- [用户租户介绍](#)
- [CREATE TENANT](#)
- [开启仲裁服务](#)
- [关闭仲裁服务](#)
- [修改降级超时时间](#)
- [替换仲裁服务](#)
- [删除仲裁服务](#)

22 使用命令行部署单副本 OceanBase 集群

OceanBase 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。

本节主要介绍通过使用命令行部署单副本 OceanBase 集群。

22.1 注意事项

如果部署的 OceanBase 数据库要实现资源隔离，那么在部署 OceanBase 数据库之前要先配置 cgroup。

有关资源隔离和配置 cgroup 的更多信息，请参见 [资源隔离概述](#) 和 [配置 cgroup](#)。

22.2 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 [服务器配置](#) 和 [使用 oatcli 初始化 OBServer 服务器](#)。
- 已获取 OceanBase 数据库 RPM 包。详细信息，请参见 [准备安装包](#)。

22.3 操作步骤

22.3.1 步骤一：安装 RPM 包

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
```

22.3.1.1 说明

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

示例如下：

```
[root@xxx /home/admin/rpm]# rpm -ivh oceanbase-4.2.0.0-
100000052023073123.el7.x86_64.rpm
Preparing... ##### [100%]
```

```
Updating / installing...
```

```
1:oceanbase-4.2.0.0-100000052023073#####  
[100%]
```

2. 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：

```
[root@xxx $rpm_dir]# rpm -ivh obclient-1.2.6-20210510164331.el7.alios7.x86_64.  
rpm
```

```
##核查是否安装成功##
```

```
[root@xxx $rpm_dir]# which obclient  
/usr/bin/obclient
```

22.3.2 步骤二：配置目录

1. 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
```

2. 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。

22.3.2.2 说明

OceanBase 数据库支持将 `slog` 从数据（data）盘中独立出来，即 `slog` 与数据文件无需在同一个磁盘上。支持配置所使用的磁盘，将 `slog` 与 `clog` 共用一块 SSD 固态硬盘。更多有关 OceanBase 数据库安装目录的信息，请参见 [OBServer 节点安装目录结构](#)。

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

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

22.3.2.3 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
|-- slog -> /data/1/obdemo/slog
|-- sstable -> /data/1/obdemo/sstable

6 directories, 0 files
```

22.3.3 步骤三：初始化 OceanBase 集群

22.3.3.4 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。

必须在 `admin` 用户下启动 observer 进程，启动语句如下：

```
cd /home/admin/oceanbase && /home/admin/oceanbase/bin/observer {-I $ip |
-i $devname} -P $rpc_port -p $sql_port -z $zone_name -d /home/admin
```

```
/oceanbase/store/$cluster_name -r '$ip:2882:2881' -c $cluster_id -n
$cluster_name -o "system_memory=30G,datafile_size=500G,
config_additional_dir=/data/1/$cluster_name/etc3;/data/log1/$cluster_name
/etc2"
```

参数解释：

参数	说明
<code>-l -i</code>	• <code>-l</code>：指定待启动的节点 IP。在多机部署场景下，不能指定 127.0.0.1 作为目标 IP。建议使用指定 IP（例如：<code>-l 10.10.10.1</code>）来启动节点。 • <code>-i</code>：指定网卡名，可通过 <code>ifconfig</code> 命令查看。 22.3.3.5 说明 支持通过同时指定 IP 和网卡名（例如：<code>-l 10.10.10.1 -i eth0</code>）来启动节点，但不建议这么用。
<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 observer 进程所属的 Zone。
<code>-d</code>	指定集群主目录，初始化目录时创建的目录。除集群名字 <code>\$cluster_name</code> 外，其他不要变动。
<code>-c</code>	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。
<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 Root Service 信息。

-O

指定集群启动参数（配置项）列表，为可选项。可以为多个配置项指定值，多个配置项的值之间使用英文逗号进行分隔。请根据实际需要，选择适当的启动参数并为每个参数指定合适的值，以优化集群的性能和资源利用率。以下是一些常用的集群启动配置项：

- [cpu_count](#)：用于设置系统 CPU 总数。
- [system_memory](#)：用于设置系统预留给租户 ID 为 500 的租户的内存容量，即指定 OceanBase 数据库内部保留内存。在机器内存较少的情况下可以考虑将该值调小，需要注意的是，在性能测试时可能会出现内存不足的问题。
- [memory_limit](#)：用于设置可用的总内存大小。
- [datafile_size](#)：用于设置数据文件占用磁盘可用空间的大小。即指定 OceanBase 数据库数据文件 `sstable` 的大小（一次性初始化），根据 `/data/1/` 可用空间评估，建议不少于 100G。
- [datafile_disk_percentage](#)：磁盘数据文件占用磁盘总空间的百分比。
- [datafile_next](#)：用于设置磁盘数据文件自动扩容的步长。
- [datafile_maxsize](#)：用于设置磁盘数据文件自动扩容的最大空间。
- [config_additional_dir](#)：用于设置本地存储配置文件的多个目录，为了冗余存储多份配置文件。
- [log_disk_size](#)：用于设置 Redo 日志磁盘的大小。
- [log_disk_percentage](#)：用于设置 Redo 日志占用其所在磁盘总空间的百分比。
- [syslog_level](#)：用于设置系统日志级别。
- [syslog_io_bandwidth_limit](#)：用于设置系统日志所能占用的磁盘 IO 带宽上限，超过带宽上限容量的系统日志将被丢弃。
- [max_syslog_file_count](#)：用于设置在回收日志文件之前可以容纳的日志文件数量。
- [enable_syslog_recycle](#)：用于设置是否开启回收系统日志的功能。

上述配置项中的 `datafile_size`、`datafile_disk_percentage`、`datafile_next` 和 `datafile_maxsize` 配合使用可以达到磁盘数据文件的自动扩容的目的，详细信息请参见 [配置磁盘数据文件的动态扩容](#)。更多集群配置信息，请参见 [配置项总览-集群级配置项](#)。

示例如下：

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
/observer -l 10.10.10.1 -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"
```

可以通过下面命令查看 observer 进程是否启动成功：

- `netstat -ntlp` 命令，如果监听到了 2881 与 2882 端口，则说明进程启动成功。
- `ps -ef|grep observer` 命令可以返回 observer 进程信息。

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 11113/observer
tcp 0 0 0.0.0.0:2882 0.0.0.0:* LISTEN 11113/observer
... ..

-bash-4.2$ ps -ef|grep observer
admin 11113 0 43 17:58 ? 00:00:14 /home/admin/oceanbase/bin/observer -l
10.10.10.1 -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
```

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 -h127.0.0.1 -uroot -P2881 -p*****  
  
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
```

22.3.3.6 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
```

22.4 后续操作

集群创建成功后，可根据业务需要创建用户租户。

使用命令行创建用户租户的详细信息，请参见 [创建租户](#)。

22.5 相关文档

- [通过 OBClient 连接 OceanBase 租户](#)
- [系统配置项总览](#)
- [ALTER USER](#)
- [用户租户介绍](#)
- [CREATE TENANT](#)

23 使用命令行部署三副本 OceanBase 集群

本文介绍如何使用命令行部署三副本 OceanBase 集群。

23.1 注意事项

如果部署的 OceanBase 数据库要实现资源隔离，那么在部署 OceanBase 数据库之前要先配置 cgroup。

有关资源隔离和配置 cgroup 的更多信息，请参见 [资源隔离概述](#) 和 [配置 cgroup](#)。

23.2 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 [服务器配置](#)、[（可选）配置时钟源](#) 和 [使用 oatcli 初始化 OBServer 服务器](#)。
- 已获取 OceanBase 数据库 RPM 包。详细信息，请参见 [准备安装包](#)。

23.3 操作步骤

23.3.1 步骤一：安装 RPM 包

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
```

23.3.1.1 说明

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

示例如下：

```
[root@xxx /home/admin/rpm]# rpm -ivh oceanbase-4.2.0.0-
100000052023073123.el7.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
```

```
1:oceanbase-4.2.0.0-100000052023073#####
[100%]
```

2. （可选）安装 OceanBase 客户端。

OceanBase 客户端 OBClient 是 OceanBase 数据库专用的命令行客户端工具，通过 OBClient 您可以连接 OceanBase 数据库的 MySQL 租户和 Oracle 租户。如果只连接 OceanBase 数据库的 MySQL 租户，也可以通过 MySQL 客户端（mysql）连接 OceanBase 数据库。

23.3.1.2 注意

由于 obclient V2.2.1 之前的版本依赖于 libobclient，所以需要先安装 libobclient。请联系技术支持获取 OBClient RPM 包和 libobclient RPM 包。

示例如下：

```
[root@xxx /home/admin/rpm]# rpm -ivh obclient-2.2.1-20221122151945.el7.
alios7.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
1:obclient-2.2.1-20221122151945.el7#####
[100%]

##核查是否安装成功##
[root@xxx /home/admin/rpm]# which obclient
/usr/bin/obclient
```

23.3.2 步骤二：配置目录

1. 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
```

2. 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。

23.3.2.3 说明

OceanBase 数据库支持将 `slog` 从数据（data）盘中独立出来，即 `slog` 与数据文件无需在同一个磁盘上。支持配置所使用的磁盘，将 `slog` 与 `clog` 共用一块 SSD 固态硬盘。更多有关 OceanBase 数据库安装目录的信息，请参见 [OBServer 节点安装目录结构](#)。

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

23.3.2.4 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  
|-- slog -> /data/1/obdemo/slog  
|-- sstable -> /data/1/obdemo/sstable  
  
6 directories, 0 files
```

23.3.3 步骤三：初始化 OceanBase 集群

23.3.3.5 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。

23.3.3.6 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。

```
cd /home/admin/oceanbase && /home/admin/oceanbase/bin/observer {-I $ip |
-i $devname} -P $rpc_port -p $sql_port -z $zone_name -d /home/admin
/oceanbase/store/$cluster_name -r '$ip:2882:2881' -c $cluster_id -n
$cluster_name -o "system_memory=30G,datafile_size=500G,
config_additional_dir=/data/1/$cluster_name/etc3;/data/log1/$cluster_name
/etc2"
```

参数解释：

参数	说明
<code>-I -i</code>	• <code>-I</code>：指定待启动的节点 IP。在多机部署场景下，不能指定 127.0.0.1 作为目标 IP。建议使用指定 IP（例如：<code>-I 10.10.10.1</code>）来启动节点。 • <code>-i</code>：指定网卡名，可通过 <code>ifconfig</code> 命令查看。 23.3.3.7 说明 支持通过同时指定 IP 和网卡名（例如：<code>-I 10.10.10.1 -i eth0</code>）来启动节点，但不建议这么用。
<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 observer 进程所属的 Zone。
<code>-d</code>	指定集群主目录，初始化目录时创建的目录。除集群名字 <code>\$cluster_name</code> 外，其他不要变动。
<code>-c</code>	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。
<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 Root Service 信息。
	指定集群启动参数（配置项）列表，为可选项。可以为多个配置项指定值，多个配置项的值之间使用英文逗号进行分隔。请根据实际需要，选择适当的启动参数并为每个参数指定合适的值，以优化集群的性能和资源利用率。以下是一些常用的集群启动配置项：

-O

- [cpu_count](#): 用于设置系统 CPU 总数。
- [system_memory](#): 用于设置系统预留给租户 ID 为 500 的租户的内存容量，即指定 OceanBase 数据库内部保留内存。在机器内存较少的情况下可以考虑将该值调小，需要注意的是，在性能测试时可能会出现内存不足的问题。
- [memory_limit](#): 用于设置可用的总内存大小。
- [datafile_size](#): 用于设置数据文件占用磁盘可用空间的大小。即指定 OceanBase 数据库数据文件 `sstable` 的大小（一次性初始化），根据 `/data/1/` 可用空间评估，建议不少于 100G。
- [datafile_disk_percentage](#): 磁盘数据文件占用磁盘总空间的百分比。
- [datafile_next](#): 用于设置磁盘数据文件自动扩容的步长。
- [datafile_maxsize](#): 用于设置磁盘数据文件自动扩容的最大空间。
- [config_additional_dir](#): 用于设置本地存储配置文件的多个目录，为了冗余存储多份配置文件。
- [log_disk_size](#): 用于设置 Redo 日志磁盘的大小。
- [log_disk_percentage](#): 用于设置 Redo 日志占用其所在磁盘总空间的百分比。
- [syslog_level](#): 用于设置系统日志级别。
- [syslog_io_bandwidth_limit](#): 用于设置系统日志所能占用的磁盘 IO 带宽上限，超过带宽上限容量的系统日志将被丢弃。
- [max_syslog_file_count](#): 用于设置在回收日志文件之前可以容纳的日志文件数量。
- [enable_syslog_recycle](#): 用于设置是否开启回收系统日志的功能。

上述配置项中的 `datafile_size`、`datafile_disk_percentage`、`datafile_next` 和 `datafile_maxsize` 配合使用可以达到磁盘数据文件的自动扩容的目的，详细信息请参见 [配置磁盘数据文件的动态扩容](#)。更多集群配置信息，请参见 [配置项总览-集群级配置项](#)。

示例如下：

zone1:

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
/observer -l 10.10.10.1 -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"
```

zone2:

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
/observer -l 10.10.10.2 -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"
```

zone3:

```
[root@xxx admin]# su - admin
-bash-4.2$ cd /home/admin/oceanbase && /home/admin/oceanbase/bin
/observer -l 10.10.10.3 -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"
```

可以通过下面命令查看 observer 进程是否启动成功：

- `netstat -ntlp` 命令，如果监听到了 2881 与 2882 端口，则说明进程启动成功。
- `ps -ef|grep observer` 命令可以返回 observer 进程信息。

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 11114/observer

tcp 0 0 0.0.0.0:2882 0.0.0.0:* LISTEN 11114/observer

... ..

```
-bash-4.2$ ps -ef|grep observer
```

```
admin 11114 0 40 16:18 ? 00:00:17 /home/admin/oceanbase/bin/observer -l
10.10.10.1 -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
```

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 -h127.0.0.1 -uroot -P2881 -p*****
```

```
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
```

23.3.3.8 注意

如果这一步失败报错了，其原因很可能就是 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 目录从头开始部署。

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
```

23.4 后续操作

集群创建成功后，可根据业务需要创建用户租户。

使用命令行创建用户租户的详细信息，请参见 [创建租户](#)。

23.5 相关文档

- [通过 OBClient 连接 OceanBase 租户](#)
- [系统配置项总览](#)
- [ALTER USER](#)
- [用户租户介绍](#)

- [CREATE TENANT](#)

24（可选）部署 OBProxy

OceanBase 数据库代理（OceanBase Database Proxy，简称：ODP，又称 OBProxy）是 OceanBase 提供的一种数据库代理中间件，它可以用于连接池管理、负载均衡、故障转移等功能。OBProxy 是一个可选的组件，它的部署与否取决于您的应用需求和场景。

本节主要介绍通过使用 RPM 包安装 OBProxy。

24.0.0.1 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 OBServer 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。

24.1 操作步骤

24.1.1 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
```

24.1.1.1 说明

OBProxy 软件会默认安装在目录 /opt/taobao/install 下。如果需要指定其他目录，可以通过设置 --prefix 参数来指定，例如安装在 /home/admin 目录下：rpm -ivh --prefix=/home/admin \$rpm_name。

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%]
```

24.1.2 步骤二：配置目录

1. （可选）创建到 obproxy 的软链接。

24.1.2.2 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

24.1.3 步骤三：初始化 OBProxy 账户

OBProxy 需要跟后端 OBCServer 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。

24.1.3.3 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 -hxxx.xxx.xxx.1 -P2881 -uroot@sys -p*****

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

24.1.4 步骤四：启动 OBProxy

24.1.4.4 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 "xxx.xxx.xxx.1:2881;xxx.xxx.xxx.2:2881;xxx.xxx.xxx.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
```

24.1.4.5 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 节点 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 "xxx.xxx.xxx.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 xxx.xxx.xxx.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: xxx.xxx.xxx.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 xxx.xxx.xxx.1:2881 -p 2883 -o
observer_sys_password=7c4a8d09ca3762af61e59520943dc26494f8941b,
enable_strict_kernel_release=false,enable_cluster_checkout=false,
enable_metadb_used=false -c test3241
```

24.2 后续操作

24.2.5 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 -hxxx.xxx.xxx.1 -P2883 -uroot@sys#test3241 -p*****
Welcome to the OceanBase. Commands end with ; or \g.
...
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 -hxxx.xxx.xxx.1 -P2883 -utest3241:sys:root -p*****
Welcome to the OceanBase. Commands end with ; or \g.
.....
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)]>
```

24.2.6 （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
```

24.3 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](#)。

25 OceanBase 数据库社区版部署概述

OceanBase 数据库社区版支持多种部署方式：

- 如果想要快速部署单机版的 OceanBase 进行功能体验，可以参考快速部署部分。
- 如果想要在生产环境中进行标准化部署，可以参照标准部署部分。
- 如果想要进行在线体验，可以参考在线体验部分。

25.1 快速部署

- 对于非原生支持的操作系统（比如 MAC 和 Windows），建议使用 Docker 镜像的方式进行部署；具体操作参见 [快速体验 OceanBase 数据库](#) 中 **方案三：部署 OceanBase 容器环境** 一节。
- 对于原生支持的操作系统（Linux 系列，具体见支持的操作系统列表），建议使用 OBD 进行部署；具体操作参见 [快速体验 OceanBase 数据库](#) 中 **方案一** 和 **方案二**。

25.2 标准部署

- 对于线上环境，建议使用 OBD 进行标准部署；具体操作参见 [通过 OBD 白屏部署 OceanBase 集群](#)。
- 对于 kubernetes 环境，建议使用 ob-operator 的方式部署；具体操作参见 [在 Kubernetes 集群中部署 OceanBase 数据库](#)。

25.3 在线体验

对于需要在线体验的用户，可以登录 [OceanBase 体验站](#) 进行功能体验。

26 软硬件要求

OceanBase 数据库是一个分布式集群产品，在生产环境中至少要求三台机器，学习环境可以部署单机版本。OceanBase 数据库的部署跟传统数据库的部署相比，存在很多共同的地方，对操作系统硬件、软件设置等会有一些最佳实践建议，这些是 OceanBase 数据库发挥高性能稳定运行的基础。

26.1 软件资源介绍

下表列出了不同操作系统版本适用的安装包：

操作系统	安装包下载地址
Alibaba Cloud Linux 2、CentOS 7、RedHat 7、Ubuntu 16.X、Debian 9.X	el7 下载地址
Alibaba Cloud Linux 3、CentOS 8、RedHat 8、Debian 10、Anolis OS 8.X	el8 下载地址

26.1.0.1 注意

建议您下载最新版本的安装包。

26.2 硬件资源准备

OceanBase 数据库在以下配置进行了系统性测试验证，推荐您使用以下软硬件配置：

项目	描述
----	----

系统	• Alibaba Cloud Linux 2/3 版本（内核 Linux 3.10.0 版本及以上） • Anolis OS 8.X 版本（内核 Linux 3.10.0 版本及以上） • Red Hat Enterprise Linux Server 7.X 版本、8.X 版本（内核 Linux 3.10.0 版本及以上） • CentOS Linux 7.X 版本、8.X 版本（内核 Linux 3.10.0 版本及以上） • Debian 9.X 版本及以上版本（内核 Linux 3.10.0 版本及以上） • Ubuntu 16.X 版本及以上版本（内核 Linux 3.10.0 版本及以上） • SUSE / OpenSUSE 15.X 版本及以上版本（内核 Linux 3.10.0 版本及以上） • KylinOS V10 版本 • 统信 UOS V20 版本 • 中科方德 NFSCina 4.0 版本及以上 • 浪潮 Inspur kos 5.8 版本
物理 CPU	• 测试环境最低要求 2 核。 • 生产环境最低要求 4 核，推荐 32 核及以上。 • 性能测试场景推荐 24 核及以上。
内存	• 测试环境最低要求 6 GB。 • 生产环境最低要求 16 GB，长期使用要求不低于 32 GB，推荐设置在 256 GB 至 1024 GB 范围内。 • 性能测试场景推荐设置在 128 GB 至 1024 GB 范围内。 26.2.0.1 注意 当部署多个集群时，推荐使用 OCP 进行统一运维管理，当部署集群比较少时，建议使用 OBD 进行安装部署。
磁盘类型	使用 SSD 存储
磁盘存储空间	最低要求 20 GB
文件系统	EXT4 或 XFS，当数据超过 16 TB 时，仅支持 XFS 类型

网卡	千兆互联及以上
----	---------

更多软硬件平台适配正在测试中。

26.2.0.2 说明

如果您部署 OceanBase 集群，确保集群内的所有机器配置相同。这里性能测试环境和生产环境的资源要求只是建议。在社区版后续版本，会进一步降低对内存的要求。

对于 Red Hat 操作系统 RHEL 9 之前版本，需要运行以下命令，手动关闭透明大页：

```
echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
```

对于 Red Hat 操作系统 RHEL 9 版本或 CentOS 操作系统，需要运行以下命令，手动关闭透明大页：

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

27（可选）查看资源

如果您计划部署分布式 OceanBase 集群，建议您在部署 OceanBase 集群时，所有的机器使用相同的软硬件配置。详细信息请参考 [软硬件要求](#)。

如果您计划单机部署 OceanBase 数据库，可以跳过此步骤。

27.0.0.1 说明

本文中的配置文件指使用 OBD 部署 OceanBase 数据库时需要配置的文件。更多配置文件信息，参考 [配置文件示例](#)。

27.1 查看操作系统

27.1.1 Red Hat Enterprise Linux Server 7.X

运行以下命令，查看操作系统信息：

```
cat /etc/redhat-release
```

返回结果如下：

```
Red Hat Enterprise Linux Server release 7.2 (Maipo)
```

27.1.2 CentOS 7.X

运行以下命令，查看操作系统信息：

```
cat /etc/redhat-release
```

返回结果如下：

```
CentOS Linux release 7.2.1511 (Core)
```

27.1.3 Anolis OS 8.X

运行以下命令，查看操作系统信息：

```
cat /etc/os-release
```

返回结果如下：

```
NAME="Anolis OS"
VERSION="8.2"
ID="anolis"
ID_LIKE="rhel fedora centos"
VERSION_ID="8.2"
PLATFORM_ID="platform:an8"
PRETTY_NAME="Anolis OS 8.2"
ANSI_COLOR="0;31"
HOME_URL="https://openanolis.cn/"
```

27.1.4 Unbutu

运行以下命令，查看操作系统信息：

```
cat /etc/os-release
```

返回结果如下：

```
NAME="Ubuntu"
VERSION="20.04.2 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.2 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
```

27.1.5 Debian9

运行以下命令，查看操作系统信息：

```
cat /etc/os-release
```

返回结果如下：

```
PRETTY_NAME="Debian GNU/Linux 9 (stretch)"
NAME="Debian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
VERSION_CODENAME=stretch
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

27.2 查看内核

运行以下命令，查看内核信息：

```
uname -r
```

OceanBase 数据库要求操作系统为 3.10.0 及以上。

27.3 查看内存

运行以下命令，查看内存：

```
free -g
```

如果可用内存小于配置文件中的 `memory_limit` 值，请清理缓存或者修改配置 `memory_limit`，将 `memory_limit` 修改为小于可用内存的值。

执行以下命令，清理缓存：

```
sudo sysctl -w vm.drop_caches=3  
# 或  
sudo echo 3 > /proc/sys/vm/drop_caches
```

27.4 查看磁盘

运行以下命令，查看磁盘：

```
df -h
```

确保配置文件中的 `data_dir`、`redo_dir` 和 `home_path` 对应的磁盘已经完成挂载。
`data_dir` 和 `redo_dir` 对应目录为空，`data_dir` 对应目录的磁盘已经使用率必须低于 4%。

27.5 查看网卡名称

您需要为配置文件中的 `devname` 配置项指定网卡。示例配置文件如下：

```
# Please set devname as the network adaptor's name whose ip is in the setting of  
severs.  
# if set severs as "127.0.0.1", please set devname as "lo"  
# if current ip is 192.168.1.10, and the ip's network adaptor's name is "eth0", please  
use "eth0"  
devname: eth0
```

执行以下命令，查看网卡名称：

```
ifconfig
```

27.5.5.1 注意

若要执行 `ifconfig` 命令，您必须先安装 `net-tools` 依赖。

28（可选）设置无密码 SSH 登录

28.1 名词解释

- 中控机器

安装 OBD 的机器，用于管理 OceanBase 集群，存储 OceanBase 集群安装包和集群配置信息。

- 目标机器

安装 OceanBase 数据库的机器。

28.2 操作步骤

在中控机中按以下步骤设置无密码 SSH 登录：

1. 在中控机器上运行以下命令查看密钥是否存在：

```
ls ~/.ssh/id_rsa.pub
```

如结果显示密钥已经存在，则无需生成新的密钥。

2. （可选）在中控机器上运行以下命令生成 SSH 公钥和私钥：

```
ssh-keygen -t rsa
```

3. 将中控机的公钥复制到目标机器的 authorized_keys 文件中：

```
ssh-copy-id -i ~/.ssh/id_rsa.pub <user>@<server_ip>
```

28.2.1（附录）脚本参考

推荐您使用以下脚本，执行以下脚本需要拥有 sudo 权限。

```
#!/usr/bin/bash

SERVERS("<user>@<server_ip1>" "<user>@<server_ip2>" "<user>@<server_ip3>")
PASSWORD="*****"

keygen() {
sudo yum -y install expect
expect -c "
```

```
spawn ssh-keygen -t rsa
expect {
*(~/.ssh/id_rsa):* { send -- \r;exp_continue}
*(y/n)* { send -- y\r;exp_continue}
*Enter* { send -- \r;exp_continue}
*(y/n)* { send -- y\r;exp_continue}
*Enter* { send -- \r;exp_continue}
eof {exit 0}
}
expect eof
"
}
copy(){
expect -c "
set timeout -1
spawn ssh-copy-id $1
expect {
*(yes/no)* { send -- yes\r; exp_continue }
*password:* { send -- $PASSWORD\r; exp_continue}
eof {exit 0}
}
expect eof
"
}
ssh_copy_id_to_all(){
keygen ;
for host in ${SERVERS[@]}
do
copy $host
```

```
done  
}  
ssh_copy_id_to_all
```

28.2.1.1 注意

您必须将脚本前两行的 **SERVERS** 列表和 **PASSWORD** 替换成您自己的实际机器列表和密码。

29（可选）配置时钟源

本文介绍如何为集群内的多台机器配置 NTP 时钟源，如果您计划单机部署 OceanBase 数据库，或者做个人测试，亦或者已配置 NTP 时钟同步，则无需设置。

如果您计划部署分布式 OceanBase 集群，需要保证集群内各机器的时间同步，否则集群无法启动，服务在运行时也会出现异常。OceanBase 集群允许的时钟偏差不能超过 2s。当超过 2s 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。

29.0.0.1 说明

- 本文基于 x86 架构的 CentOS Linux 7.9 镜像环境提供操作指导，其余环境可能略有不同，请以实际环境为准。
- OceanBase 自 V4.0.0 起支持动态修改时钟，修改的间隔在 2s 以内不会对数据正确性和集群稳定运行带来影响。
- 若部署 OCP Express 或 OCP 的节点和 OceanBase 集群不在一个节点时，也需为 OCP Express 或 OCP 的节点与 OceanBase 集群节点做时钟源同步，防止使用过程中因时钟偏差引发 OCP Express 或 OCP 操作异常。

29.1 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。

29.2 操作步骤

按以下步骤配置 NTP 时钟同步。

1. 在每台机器上执行以下命令安装 NTP。

```
yum install ntp ntpdate -y
```

2. 运行以下命令查看服务器的 NTP 连接。

```
[root@centos-01 ~]# ntpq -4p
remote refid st t when poll reach delay offset jitter
=====
=====
*time6.aliyun.co xxx.xxx.xxx.xxx 2 u 712 1024 377 21.951 4.253 4.208
```

3. （可选）开启 NTP 服务端服务。

如果您已经有了 NTP 服务端服务器，则可跳过此步骤。本文档使用目标机器 10.10.10.1 作为 NTP 服务端服务器。

a. 编辑配置文件。

以 root 用户登录 NTP 服务端服务器，运行以下命令编辑配置文件：

```
vi /etc/ntp.conf
```

在配置文件中添加以下内容：

```
server <your_ntp_server_ip>
```

保存并关闭配置文件。

b. 运行以下命令重启 NTP 服务端服务：

```
service ntpd restart
```

c. 运行以下命令查看 NTP 服务端服务是否启动成功：

```
ps -ef | grep -i ntpd
```

d. 运行以下命令设置 NTP 服务开机自启动：

```
chkconfig ntpd on
```

4. 开启 NTP 客户端服务。

29.2.0.1 注意

NTP 服务端服务启动后，您需要等待 5 分钟再开启 NTP 客户端服务。否则系统会提示 "no server suitable for synchronization found"。

a. 使用 root 用户登录 OBCServer 节点服务器。

b. 运行以下命令停止 NTP 服务：

```
service ntpd stop
```

c. 运行以下命令同步 NTP 服务端时间：

```
ntpdate 10.10.10.1
```

d. 编辑配置文件。

运行以下命令打开配置文件：

```
vi /etc/ntp.conf
```

在配置文件中添加以下内容：

```
server <your_ntp_server_ip>
```

e. 运行以下命令，启动 NTP 服务：

```
/etc/init.d/ntpd start
```

5. 运行以下命令验证配置是否成功。

运行命令：

```
ntpstat
```

返回以下结果：

```
synchronised to NTP server (xxx.xxx.xxx.xxx) at stratum 3  
time correct to within 2 ms  
polling server every 64 s
```

运行命令：

```
timedatectl
```

返回以下结果：

```
Local time: Thu 2021-04-22 11:02:32 CST  
Universal time: Thu 2021-04-22 03:02:32 UTC  
RTC time: Thu 2021-04-22 11:02:32  
Time zone: Asia/Shanghai (CST, +0800)  
NTP enabled: yes  
NTP synchronized: yes  
RTC in local TZ: yes  
DST active: n/a
```

执行如上命令能返回对应结果说明 NTP 服务生效。

30 （可选）规划磁盘

OceanBase 数据库的服务器依赖数据盘、事务日志盘和 OceanBase 数据库的安装盘。如果您是个人用户，您可以将所有数据放到一块盘上并跳过此步骤。如果您是 enterprise 用户，建议将数据分别挂载至三块磁盘。

如果您的机器上没有三块磁盘，或者您使用的是 RAID 磁盘阵列，您需要对磁盘或者磁盘阵列的逻辑卷进行分区。建议您采用以下方案分区：

- 数据盘

数据盘用来存储基线数据，路径由配置参数 `data_dir` 指定。在您首次启动 OceanBase 集群时，`${data_dir}/{sstable,slog}` 将自动创建。数据盘的大小由 `datafile_disk_percentage / datafile_size` 参数决定，您也可在部署完成后通过 `datafile_next` 和 `datafile_maxsize` 配置项完成磁盘文件的动态扩容，详细的介绍可参见 [配置磁盘数据文件的动态扩容](#)。

- 事务日志盘

事务日志盘的路径由配置参数 `clog_dir` 指定。建议您将事务日志盘的大小设置为 OceanBase 数据库内存的 3 倍到 4 倍及以上。在您首次启动 OceanBase 集群时，`${clog_dir}` 将自动创建，事务日志盘包含多个固定大小的文件，您可以根据您的需要自动创建和清除事务日志。事务日志达到磁盘总量的 80% 时，将触发自动清除。但是，只有在事务日志对应的内存数据已经合并至基线数据中时，事务日志才能被删除。

在相同数据量的情况下，事务日志的大小约为内存数据大小的三倍。因此事务日志盘所需空间上限与两次合并后的数据总量成正比。经验公式：事务日志文件大小 = 增量数据内存上限的 3~4 倍。

- OceanBase 数据库安装盘

OceanBase 数据库安装盘的路径由配置参数 `home_path` 指定。建议您为 OceanBase 数据库安装盘预留至少 200 G 空间以保存 7 天及以上的日志。OceanBase 数据库的 RPM 包安装目录位于 `${home_path}` 下。其中，基线数据文件和事务日志文件会通过软链接分别指向独立的数据盘和事务日志盘。OceanBase 数据库的运行日志位于 `${home_path}/log` 下。运行日志会不断增长，并且 OceanBase 数据库无法自动删除运行日志，因此您需要定时删除运行日志。

磁盘划分后，执行以下命令检查磁盘划分情况：

```
df -h
```

返回以下结果：

```
Filesystem Size Used Avail Use% Mounted on
devtmpfs 31G 0 31G 0% /dev
tmpfs 31G 0 31G 0% /dev/shm
tmpfs 31G 516K 31G 1% /run
tmpfs 31G 0 31G 0% /sys/fs/cgroup
/dev/vda1 493G 171G 302G 37% /
tmpfs 6.2G 0 6.2G 0% /run/user/0
/dev/vdd1 984G 77M 934G 1% /data
/dev/vdc1 196G 61M 186G 1% /redo
/dev/vdb1 492G 73M 467G 1% /home/admin/oceanbase
```

结果说明

- `/data` 为数据盘，大小为 1 TB。
- `/redo` 存放 redo 日志。
- `/home/admin/oceanbase` 存放 OceanBase 数据库的二进制文件和运行日志。

31 （可选）配置 limits.conf

本文介绍如何通过配置 `limits.conf` 限制进程数量。

如果您是个人用户，您可以跳过此步骤。如果您是 enterprise 用户，建议配置 `limits.conf`。

修改资源限制有如下两种方法：

- 通过启动时在会话级别修改。
- 通过配置文件 `/etc/security/limits.conf` 在全局级别修改。

OceanBase 数据库的进程涉及的限制包括线程最大栈空间大小（Stack）、最大文件句柄数（Open Files）和 core 文件大小（Core File Size）。

31.1 更改配置

31.1.0.1 说明

您必须在每台机器上分别执行此操作。

将会话级别的最大栈空间大小设置为 `unlimited`，最大文件句柄数设置为 `655350`，Core 文件大小设置为 `unlimited`。

执行以下命令，打开 `/etc/security/limits.conf` 配置文件：

```
sudo vim /etc/security/limits.conf
```

在 `/etc/security/limits.conf` 配置文件中添加以下内容：

```
root soft nfile 655350
root hard nfile 655350
* soft nfile 655350
* hard nfile 655350
* soft stack unlimited
* hard stack unlimited
* soft nproc 655360
* hard nproc 655360
```

```
* soft core unlimited
* hard core unlimited
```

31.1.0.2 说明

您需查看 `/etc/security/limits.d/20-nproc.conf` 文件中是否存在 `nproc` 的配置，若存在需同步修改该文件中 `nproc` 的值。

31.2 查看配置

退出当前会话，重新登录。执行以下命令，查看配置是否生效。

```
ulimit -a
```

输出如下。

```
core file size (blocks, -c) unlimited
data seg size (kbytes, -d) unlimited
scheduling priority (-e) 0
file size (blocks, -f) unlimited
pending signals (-i) 252876
max locked memory (kbytes, -l) 64
max memory size (kbytes, -m) unlimited
open files (-n) 655350
pipe size (512 bytes, -p) 8
POSIX message queues (bytes, -q) 819200
real-time priority (-r) 0
stack size (kbytes, -s) unlimited
cpu time (seconds, -t) unlimited
max user processes (-u) 655360
virtual memory (kbytes, -v) unlimited
file locks (-x) unlimited
```

从结果中可以看出：

- core file size 表示核心文件的最大阈值（以块为单位），对应 `limits.conf` 配置文件中的 `core` 参数，需查看值是否为 `unlimited`。
- open files 表示最大打开文件描述符数，对应 `limits.conf` 配置文件中的 `nofile` 参数，需查看值是否为 `655350`。
- stack size 表示堆栈大小（以千字节为单位），对应 `limits.conf` 配置文件中的 `stack` 参数，需查看值是否为 `unlimited`。
- max user processes 表示最大用户进程数，对应 `limits.conf` 配置文件中的 `nproc` 参数，需查看值是否为 `655360`。

32（可选）配置 sysctl.conf

本文介绍如何通过修改机器的 `sysctl.conf` 配置，以提高 Linux 系统的性能。

如果您是个人用户，可以跳过此步骤；如果您是 enterprise 用户，建议配置 `sysctl.conf`。

32.0.0.1 说明

您必须在每台机器上分别执行此操作。

32.1 更改配置

执行以下命令，打开 `/etc/sysctl.conf` 配置文件：

```
sudo vim /etc/sysctl.conf
```

在 `/etc/sysctl.conf` 配置文件中添加以下内容：

```
# for oceanbase
## 修改内核异步 I/O 限制
fs.aio-max-nr=1048576

## 网络优化
net.core.somaxconn = 2048
net.core.netdev_max_backlog = 10000
net.core.rmem_default = 16777216
net.core.wmem_default = 16777216
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216

net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.tcp_syncookies = 1
```

```
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.ipv4.tcp_max_syn_backlog = 16384
net.ipv4.tcp_fin_timeout = 15
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_slow_start_after_idle=0

vm.swappiness = 0
vm.min_free_kbytes = 2097152
vm.overcommit_memory = 0

fs.file-max = 6573688

# 修改进程可以拥有的虚拟内存区域数量
vm.max_map_count = 655360

# 此处为 OceanBase 数据库的 data 目录
kernel.core_pattern = /data/core-%e-%p-%t
```

其中，`kernel.core_pattern` 中的 `/data` 为 OceanBase 数据库的 `data` 目录。

32.1.0.1 注意

`max_map_count` 配置不合理的情况下，可能会导致严重的内存泄露。

32.2 加载配置

更改配置完成后，执行以下命令，加载配置，使配置生效。

```
sysctl -p
```

32.3 ARM 环境部署建议

- bios/uefi 及内核启动参数开启 NUMA 支持。

- 关闭 NUMA Balancing，避免平衡过程中性能发生抖动，命令如下。

```
sysctl kernel.numa_balancing=0
```

- 禁用内存回收和重新分配功能，命令如下。

```
sysctl vm.zone_reclaim_mode=0
```

33 （可选）关闭防火墙和 SELinux

本文介绍如何关闭防火墙和 SELinux。如果您是个人用户，可以跳过此步骤。如果您是 enterprise 用户，建议您参考本文关闭防火墙和 SELinux。

33.0.0.1 说明

您必须在每台机器上分别执行此操作。

33.1 关闭防火墙

依次执行以下命令，关闭防火墙：

```
systemctl disable firewalld
systemctl stop firewalld
systemctl status firewalld
```

33.2 关闭 SELinux

执行以下命令，打开 `/etc/selinux/config` 配置文件：

```
vi /etc/selinux/config
```

在 `/etc/selinux/config` 配置文件中修改对应配置项为以下内容：

```
SELINUX=disabled
```

执行以下命令或重启服务器，使更改生效：

```
setenforce 0
```

执行以下命令，查看更改是否生效：

```
sestatus
```

34（可选）创建用户

本文介绍如何创建普通用户并为普通用户设置 sudo 权限。

如果您是个人用户，您可以直接使用 root 用户部署集群并跳过此步骤。如果您是 enterprise 用户，建议您在机器上创建普通用户以确保安全。您可以根据实际需要，创建您的常用账户。

34.0.0.1 说明

您必须在每台机器上分别执行此操作。

按照以下步骤创建用户，此处以创建 admin 用户为例进行说明。

1. 执行以下命令，创建账户 admin。

```
[root@test001 ~]# useradd -U admin -d /home/admin -s /bin/bash
[root@test001 ~]# mkdir -p /home/admin
[root@test001 ~]# chown -R admin:admin /home/admin
```

2. 执行以下命令，为账户 admin 设置密码。

```
[root@test001 ~]# passwd admin
```

3. （可选）为账户 admin 设置 sudo 权限。

执行以下命令，打开 `/etc/sudoers` 文件，在 `/etc/sudoers` 文件添加以下内容：

```
[root@test001 ~]# vim /etc/sudoers
# 添加如下内容：
## Same thing without a password
# %wheel ALL=(ALL) NOPASSWD: ALL
admin ALL=(ALL) NOPASSWD: ALL
```

4. 修改集群相关文件所属用户

执行如下命令：

```
[root@test001 ~]# chown -R admin:admin /data
[root@test001 ~]# chown -R admin:admin /redo
[root@test001 ~]# chown -R admin:admin /home/admin
```

此处填写您真实的挂载目录。此处 `/data`、`/redo`、`/home/admin` 为示例挂载目录。

执行以下命令，查看目录权限。

```
[root@test001 data]# ls -al
```

返回如下结果：

```
drwxr-xr-x 2 admin admin 4096 2 月 9 18:43 .  
drwxr-xr-x 2 admin admin 4096 2 月 9 18:43 log1
```

35 通过 OBD 白屏部署 OceanBase 集群

本文以 x86 架构的 CentOS Linux 7.9 镜像作为环境介绍如何使用 OBD 白屏部署 OceanBase 数据库。

35.1 前提条件

- 您的机器满足软硬件要求。详细信息，参考 [软硬件要求](#)。
- 生产环境下，您需要进行环境和配置检查，具体操作请参考 [部署前配置](#) 章节。
- 仅部署 OceanBase 数据库，至少需要 2vCPU、8 GB 内存、19 GB 磁盘的可用资源。
- 部署 OceanBase 数据库及全部组件，至少需要 4vCPU、10 GB 内存、25 GB 磁盘的可用资源，推荐内存存在 16 GB 以上。
- 部署 OCP Express 组件需先安装配置 Java 环境，目前仅支持 JDK1.8 版本。详细操作可参考 [常见问题](#) 中 [部署 OCP Express 前如何配置 Java 环境](#)。

35.1.0.1 注意

OBD 是通过 SSH 远程执行安装部署，所以您需通过 SSH 验证 Java 环境是否可用，详细操作请参考 [Java 环境验证](#)。

35.2 准备软件

使用 OBD 白屏部署 OceanBase 数据库时，可以选择在线部署或离线部署两种部署方式。

- 在线部署：OBD 所在机器需保证能访问外部网络，无需提前配置部署所需安装包，部署过程中 OBD 会从远程镜像仓库获取部署所需安装包。
- 离线部署：部署过程中无需访问外部网络，您需提前将部署所需安装包上传至 OBD 本地镜像库。选择离线部署时推荐直接下载所需版本的 all-in-one 安装包。

根据不同的部署方式有不同的准备软件方法，您可根据实际情况选择合适的方法准备软件。

35.2.1 在线部署

当您选择在线部署时，可以参考本节命令在中控机上安装 OBD。

```
[admin@test001 ~]$ sudo yum install -y yum-utils
[admin@test001 ~]$ sudo yum-config-manager --add-repo https://mirrors.aliyun.com/oceanbase/OceanBase.repo
[admin@test001 ~]$ sudo yum install -y ob-deploy
[admin@test001 ~]$ source /etc/profile.d/obd.sh
```

35.2.2 离线部署

当您选择离线部署时，可参考本节命令下载并安装 all-in-one 安装包。

您可从 [OceanBase 软件下载中心](#) 下载最新的 all-in-one 安装包，并将其复制到中控机中。执行如下命令解压并安装：

```
[admin@test001 ~]$ tar -xzf oceanbase-all-in-one-*.tar.gz
[admin@test001 ~]$ cd oceanbase-all-in-one/bin/
[admin@test001 bin]$ ./install.sh
[admin@test001 bin]$ source ~/.oceanbase-all-in-one/bin/env.sh
```

35.3 操作步骤

1. 启动白屏界面

命令行执行 `obd web` 命令启动白屏界面，单击输出的地址访问白屏界面，在白屏界面中单击 **开启体验之旅** 即可进入到 **欢迎使用 OceanBase 部署向导** 界面。在 **欢迎使用 OceanBase 部署向导** 界面选择 **OceanBase 及配套工具** 模块，单击 **确定** 后进入 OceanBase 数据库部署界面。

```
[admin@test001 ~]$ obd web
start OBD WEB in 0.0.0.0:8680
please open http://10.10.10.1:8680
```

35.3.2.1 说明

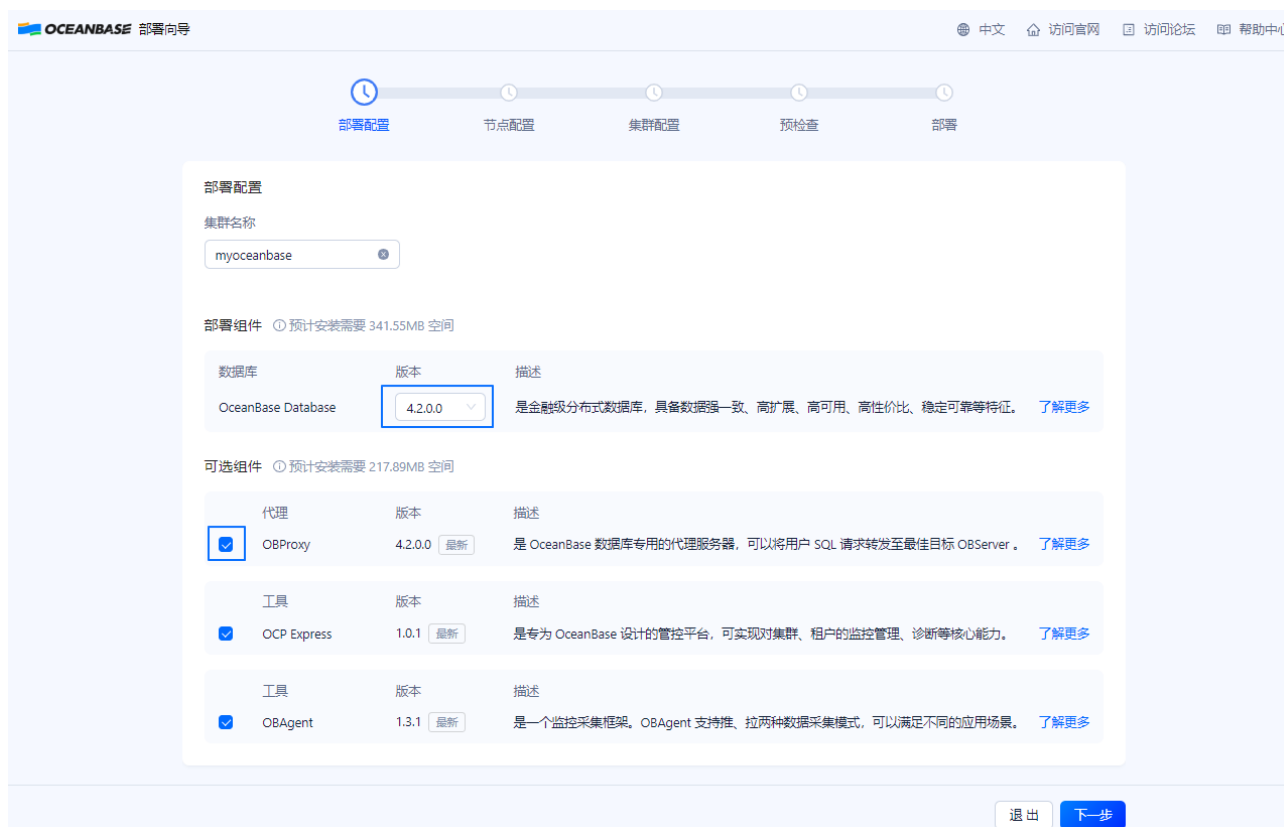
- 白屏界面默认使用 8680 端口，您可使用 `obd web -p <PORT>` 命令指定端口。
- 在阿里云或其他云环境下，可能出现程序无法获取公网 IP，从而输出内网地址的情况，此 IP 非公网地址，您需要使用正确的地址访问白屏界面。
- `obd web` 命令绑定在 0.0.0.0 上，在多网卡部署的情况下，您可通过任一个可访问的 IP 访问白屏界面。

鼠标放置到图形化操作界面右上角的 **中文** 字符处可根据显示的语种单击切换中英文界面。



2. 部署配置

部署配置 界面可以配置集群名称，部署类型和部署组件。



其中：

- **集群名称** 默认为 `myoceanbase`，支持自定义，不可和已有部署名重复。
- 默认部署全部组件，您可通过单击 **可选组件** 下对应组件前的勾选框设置是否部署对应组件。需注意，勾选 **OCP Express** 组件时会自动勾选 **OBAgent** 组件，取消勾选 **OBAgent** 组件时会自动取消勾选 **OCP Express** 组件。
- 单击对应组件后的 **了解更多** 可跳转查看对应组件的文档介绍。
- 部署所需组件时，您可单击 **版本** 下的下拉框自行选择 OceanBase 数据库的版本，其他组件版本固定为最新版本。

配置完成之后单击 **下一步** 可进入到 **节点配置** 页面。

3. 节点配置

节点配置 界面可以配置数据库和组件节点，部署用户以及软件安装路径。

其中：

- 数据库节点默认为三个 Zone，可通过单击 **+ 新增 Zone** 或尾部的删除图标新增或删除 Zone。
- **OCP Express 节点** 既可从下选框选择 OBServer 节点 IP，也可输入新的节点 IP，仅支持选择或输入一个节点。
- **OBProxy 节点** 既可通过下选框选择 OBServer 节点 IP，也可输入新的节点 IP，支持配置多个节点。
- **用户名** 默认为当前进程的启动用户，默认 SSH 端口为 22 端口，支持自定义用户和 SSH 端口。您需输入对应用户的密码，如各节点间已配置免密可免去输入密码。
- **软件路径** 默认为部署用户的家目录，可自定义修改。

配置完成之后单击 **下一步** 可进入到 **集群配置** 页面。

4. 集群配置

集群配置 界面可对集群进行配置，包括系统租户的管理员用户（root@sys）密码、数据和日志目录、数据库及各组件的端口和参数配置等。

其中：

- **模式配置** 可选择 **最大占用** 或 **最小可用**，**最大占用** 模式将最大化利用环境资源，保证集群的性能与稳定性，推荐使用；**最小可用** 模式配置满足集群正常运行的资源参数。两种模式的详细介绍可参考 [模式配置规则](#) 一文。
- **root@sys 密码** 默认为 OBD 自动生成的随机字符串，可自定义设置。支持数字、英文、特殊字符，长度 8 ~ 32 之内，特殊字符仅支持「~!@#%^&*_-+=\|(){}[]:;',./」
- 集群的数据目录和日志目录默认在 **节点配置** 页面中配置的软件路径下，需为以 **/** 开头的绝对路径，支持自定义设置，但需确保设置的目录为空。
- 数据库和各组件的端口均为默认值，可自定义设置（仅支持 1024~65535 范围），需确保设置对的端口未被占用。
- 单击打开 **更多配置** 按钮查看对应的集群或组件参数，可使用自动分配的配置，也可自定义各个参数。

全部配置完成后，单击 **下一步** 即可进入到 **预检查** 页面。

5. 预检查

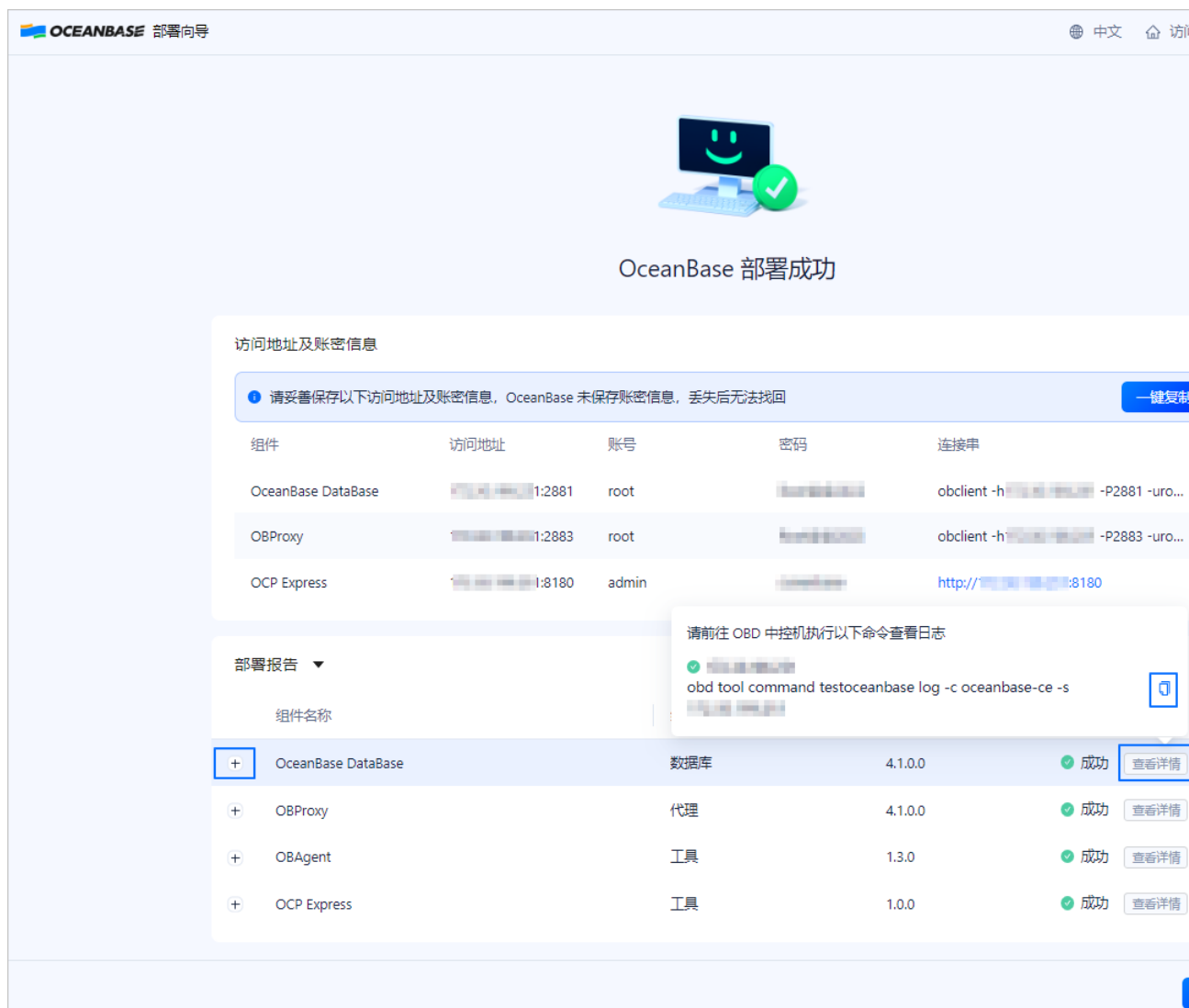
在 **预检查** 页面查看所有配置信息，若发现问题可单击 **上一步** 进行修改；确认无误后，单击 **预检查** 进行检查。

若预检查报错，您可根据页面建议选择 **自动修复** 或者单击 **了解更多方案** 跳转至错误码文档，参考文档自行修改。所有报错修改后，可单击 **重新检查** 再次进行预检查。



6. 部署

预检查通过后，单击 **部署** 即可开始 OceanBase 数据库的部署。



其中：

- 部署成功后可复制显示的连接串，在黑屏界面执行连接 OceanBase 数据库。
- 单击输出的 OCP Express 组件的连接串可跳转到 OCP Express 的登录界面，通过界面展示的账号密码登录并修改密码后可使用白屏界面管理集群。

35.3.2.2 说明

在阿里云或其他云环境下，可能出现程序无法获取公网 IP，从而输出内网地址的情况，此 IP 非公网地址，您需要使用正确的地址访问白屏界面。

- 在 **部署报告** 里单击对应组件前的展开图标即可在白屏页面查看部署的日志信息。
- 在 **部署报告** 里单击对应组件列表里的 **查看详情** 后，单击对应命令后的复制图标即可复制命令在中控机中执行查看对应组件的日志位置。

7. 单击 **完成**，结束部署

35.3.2.3 注意

如需部署多个集群，您需在白屏界面单击 **完成** 结束当前 OBD 进程后才可再次执行 `obd web` 命令进行下一集群的部署。

35.4 相关操作

35.4.3 Java 环境验证

由于 OBD 是通过远程执行脚本部署 OCP Express，所以需要通过 SSH 方式验证 Java 环境，直接在机器上执行 `java -version` 可能无效。

35.4.3.1 说明

使用终端交互会自动初始化环境变量，SSH 方式访问不会初始化环境变量，会导致使用 SSH 执行命令时出现 Java 命令不存在或使用错误 Java 版本的情况。

您可在任意一台网络与 OCP Express 所在节点连通的机器上执行如下命令进行验证。

```
# ocp_express_node_username: ocp_express 所在节点的用户名
# ocp_express_node_ip: ocp_express 所在节点 IP
[admin@test001 ~]$ ssh <ocp_express_node_username>@<ocp_express_node_ip>
'java -version'

# 输出结果
openjdk version "1.8.0_xxx"
OpenJDK Runtime Environment (build 1.8.0_362-b08)
OpenJDK 64-Bit Server VM (build 25.362-b08, mixed mode)
```

如您已安装符合条件的 Java 但是验证却没有通过，可以通过以下任一方法解决：

- 方法一：通过组件页面 **更多配置** 配置 `java_bin` 路径

如下图，在配置项 `java_bin` 中配置 Java 的真实路径，如 `/jdk8/bin/java`。

OCEANBASE 部署向导

中文 访问官网 访问论坛 帮助中心

部署配置 节点配置 集群配置 预检查 部署

OBProxy 服务端口 2883 OBProxy Exporter 端口 2884

OBAgent 监控服务端口 8088 OBAgent 管理服务端口 8089

OCP Express 端口 8180

更多配置 ☒

OBProxy 参数名称	参数值	介绍
proxy_mem_limited	自动分配 2G	ODP 运行时内存上限。超过上限 ODP 即自动退出。请输入带容量带单...
obproxy_sys_password	自定义 请输入	password pf obproxy sys user

OCP Express 参数名称	参数值	介绍
log_dir	自动分配 请输入	OCP express server日志目录，默认为工作目录下的log
java_bin	自定义 /jdk8/bin/	OCP express 使用的java可执行文件的路径
memory_size	自动分配 请输入	OCP express server进程内存大小。请输入带容量带单位的整数，如2G
logging_file_max_size	自定义 100MB	单个日志文件大小
logging_file_total_size_cap	自定义 1GB	日志文件总大小
logging_file_max_history	自动分配 请输入	最多保留的归档日志文件的天数，默认不限制

退出 上一步 下一步

- 方法二：将 Java 可执行文件软链接到 `/usr/bin/java` 中

```
[admin@test001 bin]$ pwd
/jdk8/bin
[admin@test001 bin]$ ln -s /jdk8/bin/java /usr/bin/java
```

35.4.4 管理部署后的集群

您可执行如下命令对 OBD 部署的集群进行管理。更多操作详见 [集群命令组](#)。

查看集群列表

```
[admin@test001 ~]$ obd cluster list
```

查看集群状态，以部署名为 myoceanbase 为例

```
[admin@test001 ~]$ obd cluster display myoceanbase
```

```
# 停止运行中的集群，以部署名为 myoceanbase 为例
```

```
[admin@test001 ~]$ obd cluster stop myoceanbase
```

```
# 销毁已部署的集群，以部署名为 myoceanbase 为例
```

```
[admin@test001 ~]$ obd cluster destroy myoceanbase
```

35.4.5 部署特定版本组件

使用 all-in-one 安装包部署时，all-in-one 的包是基于 OceanBase 版本进行迭代，若包中有其他组件存在更新版本，您可从 [OceanBase 软件下载中心](#) 下载最新版本的组件，参考如下步骤将其上传至本地镜像库，OBD 部署时会自动获取本地镜像库中的最新版本。

1. 进到组件安装包所在目录下，将安装包添加至本地镜像库

```
[admin@test001 rpm]$ obd mirror clone *.rpm
```

2. 查看本地镜像中安装包列表

```
[admin@test001 rpm]$ obd mirror list local
```

36 通过 OCP 部署 OceanBase 集群

本文以 x86 架构的 CentOS Linux 7.9 镜像作为环境介绍如何使用 OCP 白屏部署 OceanBase 数据库。

36.1 前提条件

- 您已部署 OCP，详细操作请参考 [OceanBase 云平台/部署指南](#)。
- 当前登录 OCP 的用户拥有 **主机管理** 和 **Create Cluster** 的权限。
- 已上传 oceanbase-ce、oceanbase-ce-libs 和 oceanbase-ce-utils 软件包至 OCP。
- OceanBase 宿主机已安装好 MySQL 客户端。
- OceanBase 宿主机已安装 python2.7。
- （可选）若您创建的是多副本 OceanBase 集群，则需当前 OCP 中已存在可用的 ODP 集群。如何新增 ODP 集群可参考 [OceanBase 云平台/用户指南/管理 OBProxy/创建 OBProxy 集群](#)。

36.2 操作步骤

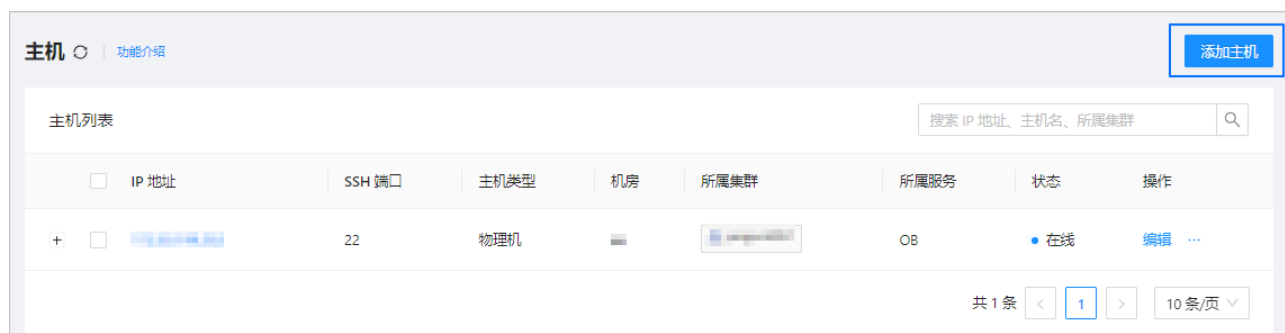
36.2.0.1 说明

本文以 OCP V4.0.3 社区版本为例提供操作指导。不同 OCP 版本的操作界面可能不同，请以实际界面为准。

36.2.1 步骤一：添加机器到 OCP 资源池

部署 OceanBase 集群前，您需先参考本节内容将机器添加到 OCP 资源池。

1. 使用 admin 账号登录 OCP。
2. 在左侧导航栏中单击 **主机**，进入 **主机** 页面。
3. 在 **主机** 页面右上角单击 **添加主机**。



4. 在弹出的对话框中填写机器信息。

主机

主机列表

IP 地址

SSH 端口

22

添加主机

IP 地址

请输入

SSH 端口

22

机型

请选择

机房

请选择

主机类型

物理机 容器

凭证

请选择

主机别名 (可选)

请输入

说明 (可选)

新增机房

机房

请输入

区域

请选择

说明 (可选)

新增区域

区域

请输入

说明 (可选)

取消

确定

各字段填写说明如下表所示。

字段	描述
IP 地址	待添加机器的 IP，请输入机器的 IPv4 地址。
SSH 端口	默认为 22。
机型	为配置相同的主机指定的一个标签，便于 OCP 更好地管理主机。如果没有对应的机型，可在下拉菜单中单击 新增机型 ，添加新的机型。
机房	选择机器所处机房。如果没有对应的机房，可在下拉菜单中单击 新增机房 在 OCP 中新增机房信息。机房信息包括 机房 与 区域。 • 机房是 OceanBase 集群需要记录的一种主机属性，这个属性会作为 OceanBase 负载均衡和 SQL 语句路由策略的参考项，请按照实际情况填写。 • 区域用于表示主机所处的地理区域，是 OceanBase 集群需要记录的主机属性之一，对于 OceanBase 集群的负载均衡和 SQL 语句路由策略会有一定的影响，请按照实际情况填写。 36.2.1.2 说明 OCP V3.1.1 版本开始支持多 AZone 模式，新增机房时，将在当前 OCP 所在的 AZone 中创建新的机房。
主机类型	有 物理机 和 容器 两种选项。
凭据	选择远程登录物理机使用的凭据，您可以在下拉菜单中单击 新增凭据 来新建凭据。 新增凭据时需注意： • 凭据名：以英文字母开头、英文或数字结尾，可包含英文、数字和下划线，且长度为 2 ~ 32。 • 授权类型：包含用户名/密码和 SSH 私钥两种方式。 • 用户类型：支持拥有 root 权限的 SSH 用户，拥有 root 权限的普通用户须先配置免密执行 sudo 指令。OCP 目前仅支持配置所有指令免密执行 sudo，即在 <code>/etc/sudoers</code> 文件中，添加 <code>username ALL=(ALL) NOPASSWD:ALL</code>。
主机别名（可选）	主机别名是您为配置相同的主机指定的一个标签，建议您指定有意义的名称，便于更好地管理主机。
说明（可选）	主机的注释，便于更好地管理主机。

5. 填写完成后，单击 **确定**。

36.2.2 步骤二：创建 OceanBase 集群

您可参考本节介绍通过 OCP 创建 OceanBase 集群。

1. 在 OCP 白屏界面中根据实际业务场景，找到新建集群的入口。

- 如果您没有可管理的集群，系统会在 **集群** 页面提示您新建集群，可直接在提示信息中单击 **创建集群**。
- 如果您已有可管理的集群，可单击 **集群** 后在 **集群** 页面右上角单击 **创建集群**。

2. 在 **创建集群** 页面，设置集群的基础信息。

基本设置

集群类型

☒ 主集群 ☐ 备集群

集群名

test_cluster

root@sys 密码

.....

随机生成

OB 版本

oceanbase-4.0.0.0-102000012023010515-el7

关联 OBProxy 集群 ☒

仅可选择与此 OB 集群网络一致的 OBProxy 集群

选择关联 OBProxy 集群

obproxy_cu2 4.0.0 x

仅可选择 4.0 及以上版本的 OBProxy 集群

基础信息相关说明如下表所示。

配置	描述
集群类型	选择 主集群 。
集群名称	自定义待管理的集群的名称。集群名称必须以英文字母开头，可支持大小写字母、数字和下划线，长度为 2 ~ 48 个字符。
root@sys 密码	支持自定义或随机生成。密码需要满足以下复杂度条件： ● 长度：8 ~ 32 字符 ● 支持字母、数字和特殊字符，且至少包含大、小写字母、数字和特殊字符各 2 个 ● 支持的特殊字符如下：~!@#%^&* _-+={} []:;,./? 输入密码后，您可以单击 复制密码，将自定义或随机生成的密码复制到剪贴板。
OB 版本	可以从列表中选择已有的 OceanBase 数据库版本，也可以在下拉菜单中单击 添加版本 ，上传 OceanBase RPM 包。
关联 OBProxy 集群	该选项用于关联已有的 ODP 集群，如您创建的是多副本的 OceanBase 集群，建议您打开 关联 OBProxy 集群 开关，为该 OceanBase 集群关联 ODP 集群。关联后您的业务涉及的 SQL 请求将会被精准转发到相应的副本，使您对 OceanBase 数据库的访问效果能够媲美访问单机数据库。 ● 默认使用 proxyro 用户关联，您无需填写用户名和密码。 ● 在下拉框中选择要关联的 ODP 集群。如下拉框中没有可关联的 ODP 集群，请参考本文 前提条件 中的描述添加 ODP 集群。 注意 当 OB 版本 中选择的集群为 V4.0 及以上版本时，仅支持关联 V4.0.0 及以上版本的 OBProxy 集群。

3. 设置集群的部署模式信息。

默认添加 3 个 Zone 的信息，如果您希望部署的集群 Zone 的个数大于 3 个，您可以在下方单击 **+ 新增** 按钮，增加 Zone 信息。如果部署的集群 Zone 的个数小于 3 个，您可以单击 Zone 后面的删除图标。

部署模式

Zone 名称	机房	机型 (可选)	CPU 架构	主机	Root Server 位置
zone1	请选择	请选择	x86_64	请选择	请选择
zone2	请选择	请选择	x86_64	请选择	请选择
zone3	请选择	请选择	x86_64	请选择	请选择
+ 新增 Zone					

Zone 优先级排序

Zone

同时选择多个 Zone 穿梭到右侧后，可设置为同一优先级

zone1

zone2

zone3

>

优先级排序

优先级从上到下代表从高到低，可拖拽排序

恢复为默认

> 产品版本：V4.2.1

207

每个 Zone 需要设置的信息及其说明如下表所示。

配置	描述
Zone 名称	一般会有一个默认名称，您可以根据需要自定义名称。Zone 名称必须以英文字母开头，可支持大小写字母、数字和下划线，长度为 2 ~ 48 个字符。部署多 Zone 的 OceanBase 集群必须有一个 Zone 与主 Zone 的 Region 相同。
机房	Zone 所在的机房，一个 Zone 只能存在于一个机房。
机型（可选）	如果选择了机型，后面主机列表会根据机型进行过滤。
CPU 架构	当选择的 OB 版本 中存在多个不同架构的安装包时，需选择匹配此软件包的主机硬件架构。 36.2.2.3 注意 - 默认架构类型为第一个安装包的架构类型，支持切换。切换指定架构后，主机 列表中即展示该架构的主机。 - 为不同的 Zone 设置不同的架构时，混合架构部署模式可能存在稳定性与性能问题，请谨慎操作。
主机	您可以选择多个主机，或进行添加主机的操作。
RootServer 位置	您需选择一个 IP 作为 Root Service 所在的机器。多副本的 OceanBase 集群，其每个 Zone 都需指定一个 Root Server。
Zone 优先级排序	Zone 的优先级排序。最高优先级的 Zone 会被视为 Primary Zone，且只能有一个。排序方法如下： - 勾选左侧列表框中的一个或多个 Zone。左侧列表框中显示了当前集群中所有可选的 Zone。 - 单击中间的 > 按钮。此时勾选的 Zone 就会被移动到 优先级排序 列表中。同时勾选的多个 Zone 具有相同的优先级。 - 重复 1、2 动作，添加低一优先级的 Zone。 - 如需调整优先级，可在 优先级排序 列表中拖拽以调整顺序。 列表中从上到下，优先级依次递减。

4. 打开 **参数配置** 模块，并配置集群参数。

- 您可通过如图 ① 处，逐个添加启动参数项并为其配置值。
- 您也可通过如图 ② 处，单击 **选择参数模板** 选择一个参数模板，系统会将模板中的参数连同配置自动填充到此处。

36.2.2.4 说明

在未创建集群参数模板的情况下，您可单击 **新建集群模板**，为集群创建参数模板，详情可参见 OceanBase 云平台文档 [集群参数模板管理](#)。



5. 打开 **自定义配置**，对路径及端口进行配置。

- 配置路径：
 - a. 在 **路径配置** 模块，根据需要自定义 **软件安装路径**、**数据盘路径** 和 **日志盘路径**。
 - b. 配置完成后，单击 **测试**，校验路径是否可用。若不可用，您可根据测试结果进行排查，或将路径修改为其它可用路径。

路径配置

软件安装路径 

/home/admin

/oceanbase

 该路径已赋权可用

数据盘路径 

/data

 admin 用户没有主机 172.17.0.1 中该路径的 写 权限

日志盘路径 

/data/log1

 该路径已赋权可用

测试

恢复默认

- 配置端口：
 - a. 在 **端口配置** 模块，自定义 **SQL 端口** 和 **RPC 端口**。
 - b. 配置完成后，单击 **测试**，校验端口是否已被占用。若被占用，您需重新配置可用端口。

端口配置

SQL 端口

2881	^
	v

✓ 此端口未被占用，检查通过

RPC 端口

2882	^
	v

✓ 此端口未被占用，检查通过

测试

恢复默认

6. 完成测试后，单击 **提交**。

7. 在弹出的 **确认提交信息** 对话框中，确认信息无误后，单击 **确定**。

36.2.2.5 说明

您可在 **系统管理 > 任务** 中查看该任务的执行进度。

36.3 相关文档

集群部署完成后，需创建业务租户，详细信息请参见 [OcenaBase 云平台/用户指南/管理租户/租户基础操作/新建租户](#)。

37 使用命令行部署 OceanBase 数据库生产环境

本文介绍如何通过 OBD 黑屏命令行方式部署 OceanBase 数据库生产环境。

37.0.0.1 说明

OBD 除了支持通过命令行方式部署 OceanBase 数据库生产环境外，也支持通过白屏部署 OceanBase 集群环境。推荐您通过 OBD 白屏方式进行部署，详细操作步骤请参考 [通过 OBD 白屏部署 OceanBase 集群](#)。

37.1 名词解释

- 中控机器
安装 OBD 的机器，用于管理 OceanBase 集群，存储 OceanBase 集群安装包和集群配置信息。
- 目标机器
安装 OceanBase 数据库的机器。
- OBD
OceanBase Deployer, OceanBase 安装部署工具，简称为 OBD。详细信息请参考 [OBD 文档](#)。
- ODP
OceanBase Database Proxy, OceanBase 数据库代理，是 OceanBase 数据库专用的代理服务器，简称为 ODP（又称为 OBProxy）。详细信息请参考 [ODP 文档](#)。
- OCP
OceanBase Cloud Platform, OceanBase 云平台，简称为 OCP。详细信息请参考 [OCP 文档](#)
- OCP Express
基于 Web 的 OceanBase 数据库 4.x 管理工具，融合在 OceanBase 数据库集群中，支持对数据库集群关键性能及基本数据库管理功能。详细信息请参考 [OceanBase 云平台 Express \(OCP Express\)](#)。

37.2 前提条件

在部署 OceanBase 数据库之前，请您确认以下信息：

- 您的机器满足软硬件要求。详细信息，参考 [软硬件要求](#)。
- 生产环境下，您需要进行环境和配置检查，具体操作请参考 [部署前配置](#) 章节。

- 中控机器中已安装 OBD，建议安装最新版本，详细信息请参考 [安装并配置 OBD](#)。
- 您的机器中已安装 OBClient 客户端，详细信息请参考 [OBClient 文档](#)。

37.3 部署模式

本文采用三副本部署模式，推荐使用四台机器，您可以根据自己实际情况选择合适的部署方案。本文中四台机器的使用情况如下：

角色	机器	备注
OBD	10.10.10.4	安装在中控机上的自动化部署软件
OBServer 节点	10.10.10.1	OceanBase 数据库 Zone1
OBServer 节点	10.10.10.2	OceanBase 数据库 Zone2
OBServer 节点	10.10.10.3	OceanBase 数据库 Zone3
OBAgent	10.10.10.1、10.10.10.2、10.10.10.3	OceanBase 数据库监控采集框架
ODP	10.10.10.4	OceanBase 数据库专用的反向代理软件
OCP Express	10.10.10.4	基于 Web 的 OceanBase 数据库 4.x 管理工具

37.3.0.1 说明

- 在生产环境下，推荐将 ODP 和应用部署在一台机器上，以节省 ODP 访问应用的时间。您可以在每个应用的服务器上部署一个 ODP 服务。在本例中，为了方便介绍，将 ODP 单独部署。
- 部署 ODP 的机器使用的配置可以和部署 OceanBase 数据库的机器配置不一致。部署 ODP 最小只需 1C/1G 内存。
- 单机部署操作步骤可参考 OBD 文档中 [单机部署 OceanBase 数据库](#) 一文。

37.4 操作步骤

37.4.0.1 注意

- 以下内容以 x86 架构的 CentOS Linux 7.9 镜像作为环境，其他环境可能略有不同。
- 部署 OceanBase 集群之前，为了数据安全，建议您切换到非 root 用户。

37.4.1 步骤一：（可选）下载并安装 all-in-one 安装包

从 V4.0.0 开始，OceanBase 提供统一的安装包 all-in-one package。您可以通过这个统一的安装包一次性完成 OBD、OceanBase 数据库、ODP、OBAgent 等组件的安装。

您也可根据实际需求从 [OceanBase 软件下载中心](#) 选择部分组件下载安装或者指定组件的版本。

37.4.1.2 在线安装

若您的机器可以连接网络，可执行如下命令在线安装。

```
[admin@test001 ~]$ bash -c "$(curl -s https://obbusiness-private.oss-cn-shanghai.aliyuncs.com/download-center/opensource/oceanbase-all-in-one/installer.sh)"
[admin@test001 ~]$ source ~/.oceanbase-all-in-one/bin/env.sh
```

37.4.1.3 离线安装

若您的机器无法连接网络，可参考如下步骤离线安装。

1. 从 [OceanBase 软件下载中心](#) 下载最新的 all-in-one 安装包，并将其复制到中控机任意目录下。
2. 在安装包所在目录下执行如下命令解压安装包并安装。

```
[admin@test001 ~]$ tar -xzf oceanbase-all-in-one-*.tar.gz
[admin@test001 ~]$ cd oceanbase-all-in-one/bin/
[admin@test001 bin]$ ./install.sh
[admin@test001 bin]$ source ~/.oceanbase-all-in-one/bin/env.sh
```

37.4.1.4 说明

您可通过执行 `which obd` 和 `which obclient` 命令检测是否安装成功，如果可以找到 `oceanbase-all-in-one` 目录下的 `obd` 和 `obclient` 路径，则表示安装成功。

37.4.2 步骤二：配置 OBD

37.4.2.5 说明

本文以 OBD V2.2.0 为例提供操作指导。不同 OBD 版本的配置文件可能不同，请以实际配置文件为准。更多 OBD 相关内容请参见 [OBD 文档](#)。

如果是离线部署 OceanBase 集群，可参考 **步骤一** 在中控机上下载并安装 all-in-one 安装包，all-in-one 安装包提供的组件已经经过相互适配测试，为官方推荐版本。

如果对部署所需组件版本有特别要求，可从 [OceanBase 软件下载中心](#) 自行下载组件对应版本安装包，复制到中控机任一目录，在该目录下参考以下步骤配置 OBD。

37.4.2.6 说明

如果是在线部署 OceanBase 集群或通过 all-in-one 安装包部署 OceanBase 集群，则跳过步骤 1~3。

1. 禁用远程仓库

```
[admin@test001 rpm]$ obd mirror disable remote
```

37.4.2.7 说明

安装 all-in-one 安装包后默认关闭远程仓库，您可通过 `obd mirror list` 命令进行确认，查看 Type=remote 对应的 Enabled 变成了 False，说明已关闭远程镜像源。

2. 将安装包添加至本地镜像库

```
[admin@test001 rpm]$ obd mirror clone *.rpm
```

3. 查看本地镜像库中安装包列表

```
[admin@test001 rpm]$ obd mirror list local
```

4. 选择配置文件

若您机器中的 OBD 是通过直接下载的方式安装，则可在 `/usr/obd/example` 目录下查看 OBD 提供的配置文件示例。

若您机器中的 OBD 是通过解压 all-in-one 安装包的方式安装，则可在 `~/.oceanbase-all-in-one/obd/usr/obd/example` 目录下查看 OBD 提供的配置文件示例。请根据您的资源条件选择相应的配置文件。

37.4.2.8 说明

配置文件分为小规格开发和专业开发两种模式，两种模式的配置文件中配置项基本相同，配置的规格略有不同，您可根据实际资源条件进行选择。

- 小规格开发模式：适用于个人设备（内存不低于 8 GB），配置文件名中带有 `mini` 或 `min` 标识。
- 专业开发模式：适用于高配置 ECS 或物理服务器（可用资源不低于 16 核 64 GB）。如果您选择单机部署，即目标机器只有一台，可使用单机部署配置文件。
- 本地单机部署配置样例：`mini-local-example.yaml`、`local-example.yaml`
- 单机部署配置样例：`mini-single-example.yaml`、`single-example.yaml`
- 单机部署 + ODP 配置样例：`mini-single-with-obproxy-example.yaml`、`single-with-obproxy-example.yaml`

如果您选择分布式部署，即目标机器有多台，可使用分布式安装配置文件。

- 分布式部署配置样例：`mini-distributed-example.yaml`、`distributed-example.yaml`
- 分布式部署 + ODP 配置样例：`mini-distributed-with-obproxy-example.yaml`、`distributed-with-obproxy-example.yaml`
- 分布式部署 + ODP + OCP Express 配置样例：`default-components.yaml`、`default-components-min.yaml`
- 分布式部署全部组件：`all-components-min.yaml`、`all-components.yaml`

37.4.2.9 说明

若您选择部署 OCP Express 或已部署 OCP，则无需部署 Prometheus 和 Grafana。

5. 修改配置文件。

此处以使用 `default-components.yaml` 部署分布式 OceanBase 集群为例，介绍如何修改配置文件。

37.4.2.10 说明

您需根据您的真实情况修改下述参数。

a. 修改用户名和密码。

```
## Only need to configure when remote login is required
user:
  username: admin
  # password: your password if need
  key_file: /home/admin/.ssh/id_rsa
```

```
# port: your ssh port, default 22
# timeout: ssh connection timeout (second), default 30
```

`username` 为登录到目标机器的用户名，确保您的用户名有 `home_path` 的写权限。`password` 和 `key_file` 均用于验证用户，通常情况下只需要填写一个。

37.4.2.11 注意

在配置密钥路径后，如果您的密钥不需要口令，请注释或者删除 `password`，以免 `password` 被视为密钥口令用于登录，导致校验失败。

- b. 修改机器的 IP、端口和相关目录，并配置内存相关参数及密码。

```
oceanbase-ce:
servers:
# Please don't use hostname, only IP can be supported
- name: server1
ip: 10.10.10.1
- name: server2
ip: 10.10.10.2
- name: server3
ip: 10.10.10.3
global:
devname: eth0
cluster_id: 1
# please set memory limit to a suitable value which is matching resource.
memory_limit: 64G # The maximum running memory for an observer
system_memory: 30G # The reserved system memory. system_memory is
reserved for general tenants.
datafile_size: 192G # Size of the data file.
datafile_next: 200G
datafile_maxsize: 1T
log_disk_size: 192G # The size of disk space used by the clog files.
```

```
enable_syslog_wf: false # Print system logs whose levels are higher than
WARNING to a separate log file. The default value is true.
max_syslog_file_count: 4 # The maximum number of reserved log files before
enabling auto recycling. The default value is 0.
# observer cluster name, consistent with obproxy's cluster_name
appname: obdemo
ocp_meta_tenant: # The config for ocp express meta tenant
tenant_name: ocp
max_cpu: 1
memory_size: 2G
log_disk_size: 7680M
mysql_port: 2881 # External port for OceanBase Database. The default value
is 2881. DO NOT change this value after the cluster is started.
rpc_port: 2882 # Internal port for OceanBase Database. The default value is
2882. DO NOT change this value after the cluster is started.
# The working directory for OceanBase Database. OceanBase Database is
started under this directory. This is a required field.
home_path: /home/admin/observer
# The directory for data storage. The default value is $home_path/store.
data_dir: /data
# The directory for clog, ilog, and slog. The default value is the same as the
data_dir value.
redo_dir: /redo
root_password: ***** # root user password, can be empty
proxyro_password: ***** # proxyro user password, consistent with obproxy's
observer_sys_password, can be empty
server1:
zone: zone1
server2:
```

```
zone: zone2
server3:
zone: zone3
```

若机器中存在不一致的配置项，可将相关配置项移到对应 server 里进行配置，配置文件中对应 server 中的配置优先级高于 `global`。以其中两台机器中配置端口不同为例：

```
server2:
mysql_port: 3881
rpc_port: 3882
zone: zone2
server3:
mysql_port: 2881
rpc_port: 2882
zone: zone3
```

配置项	是否必选	默认值	说明
servers	必选	无	每台机器需要用 <code>- name: 机器标识名 (换行)ip: 机器 IP 指定</code> ，多个机器就指定多次，机器标识名不能重复。在机器 IP 不重复的情况下，也可以使用 <code>- <ip> (换行) - <ip></code> 的格式指定，此时 <code>- <ip></code> 的格式相当于 <code>- name: 机器 IP (换行) ip: 机器 IP</code> 。
devname	可选	无	和 <code>servers</code> 里指定的 IP 对应的网卡，通过 <code>ip addr</code> 命令可以查看 IP 和网卡对应关系。

memory_limit	可选	0	observer 进程能从环境中获取的最大内存，未配置的情况下以 <code>memory_limit_percentage</code> 配置项为准，配置项详细介绍请参考 memory_limit 和 memory_limit_percentage 。
system_memory	可选	0M	保留的系统内存，该参数值会占用 <code>memory_limit</code> 的内存，未配置的情况下 OceanBase 数据库会自适应。
datafile_size	可选	0	指定对应节点数据文件（block_file）大小，未配置的情况下以 <code>datafile_disk_percentage</code> 配置项为准，详细介绍请参见 datafile_size 和 datafile_disk_percentage 。
datafile_next	可选	0	控制磁盘空间的增长步长，用于设置自动扩容，未配置的情况下若要开启自动扩容功能可参见 配置磁盘数据文件的动态扩容 。
datafile_maxsize	可选	0	限制磁盘空间的最大可用上限，用于设置自动扩容，未配置的情况下若要开启自动扩容功能可参见 配置磁盘数据文件的动态扩容 。

log_disk_size	可选	0	用于设置 Redo 日志磁盘的大小，未配置的情况下以 <code>log_disk_percentage</code> 配置项为准，详细介绍请参考 log_disk_size 和 log_disk_percentage 。
enable_syslog_wf	可选	true	设置是否把 WARN 以上级别的系统日志打印到一个单独的日志文件中。
max_syslog_file_count	可选	0	设置在回收日志文件之前可以容纳的日志文件数量。值为 0 时表示不自动清理。
appname	可选	obcluster	设置 OceanBase 集群名。
ocp_meta_tenant	可选	tenant_name: ocp max_cpu: 1 memory_size: 2147483648	配置 OCP Express 所需的 meta 租户规格，ocp_meta_tenant 下的配置都会作为创建租户时的参数传入。 示例中仅列举了几个重要配置，具体可配置参数可参考创建租户命令支持的参数，详情请参见 OBD 文档 集群命令组 中 obd cluster tenant create 命令介绍。
mysql_port	必选	2881	设置 SQL 服务协议端口号，默认为 2881。
rpc_port	必选	2882	设置远程访问的协议端口号，是 observer 进程跟其他节点进程之间的 RPC 通信端口，默认为 2882。
home_path	必选	无	OceanBase 数据库安装路径，建议在普通用户 admin 下。

data_dir	可选	\$home_path/store	设置存储 SSTable 等数据的目录，建议配置为独立的磁盘。
redo_dir	可选	默认与 data_dir 相同	clog、ilog 和 slog 的目录，默认与 data_dir 值相同，建议配置为独立的磁盘。
root_password	可选	OB V2.1.0 之前版本默认为空 OB V2.1.0 及之后版本默认为随机字符串	OceanBase 集群的超级管理员（root@sys）的密码，建议设置复杂的密码。使用 OB V2.1.0 及之后版本时，未设置的情况下会自动生成随机字符串。
proxyro_password	可选	OB V2.1.0 之前版本默认为空 OB V2.1.0 及之后版本默认为随机字符串	ODP 连接 OceanBase 集群使用的账户名（proxyro@sys）的密码。使用 OB V2.1.0 及之后版本时，未设置的情况下会自动生成随机字符串。

c. 配置 obproxy-ce 组件并修改 IP 和 home_path 。

```
obproxy-ce:
depends:
- oceanbase-ce
servers:
- 10.10.10.4
global:
listen_port: 2883 # External port. The default value is 2883.
prometheus_listen_port: 2884 # The Prometheus port. The default value is
2884.
home_path: /home/admin/obproxy
enable_cluster_checkout: false
skip_proxy_sys_private_check: true
enable_strict_kernel_release: false
obproxy_sys_password: ***** # obproxy sys user password, can be empty.
When a depends exists, OBD gets this value from the oceanbase-ce of the
depends.
observer_sys_password: ***** # proxyro user password, consistent with
oceanbase-ce's proxyro_password, can be empty. When a depends exists,
OBD gets this value from the oceanbase-ce of the depends.
```

配置项	是否必选	默认值	说明
servers	必选	无	每台机器需要用 <code>- name: 机器标识名 (换行)ip: 机器 IP 指定</code> ，多个机器就指定多次，机器标识名不能重复。在机器 IP 不重复的情况下，也可以使用 <code>- <ip> (换行) - <ip></code> 的格式指定，此时 <code>- <ip></code> 的格式相当于 <code>- name: 机器 IP (换行) ip: 机器 IP</code> 。
listen_port	必选	2883	ODP 监听端口，默认 2883。
prometheus_listen_port	必选	2884	ODP prometheus 监听端口，默认 2884。
home_path	必选	无	ODP 安装路径，建议在普通用户 admin 下。
obproxy_sys_password	可选	OB V2.1.0 之前版本默认为空 OB V2.1.0 及之后版本默认为随机字符串	ODP 管理员账户 (root@proxysys) 的密码。使用 OB V2.1.0 及之后版本时，未设置的情况下会自动生成随机字符串。
observer_sys_password	可选	OB V2.1.0 之前版本默认为空 OB V2.1.0 及之后版本默认为和 <code>proxyro_password</code> 相同的随机字符串	ODP 连接 OceanBase 集群使用的账户名 (proxyro@sys) 的密码，配置密码需和 <code>proxyro_password</code> 的值相同。使用 OB V2.1.0 及之后版本时，未设置的情况下会自动生成和 <code>proxyro_password</code> 相同的随机字符串。

d. 修改 obagent 和 ocp-express 组件的 IP 和 `home_path`。

37.4.2.12 注意

OBD 自 V2.0.0 起支持部署 OCP Express, 且 OBD V2.0.0 仅支持部署 OCP Express V1.0.0, OCP Express V1.0.1 及之后版本需使用 OBD V2.1.0 及之后版本部署。

```
obagent:
depends:
- oceanbase-ce
servers:
- name: server1
# Please don't use hostname, only IP can be supported
ip: 10.10.10.1
- name: server2
ip: 10.10.10.2
- name: server3
ip: 10.10.10.3
global:
home_path: /home/admin/obagent
ocp-express:
depends:
- oceanbase-ce
- obproxy-ce
- obagent
servers:
- name: server1
ip: 10.10.10.4
global:
# The working directory for prometheus. prometheus is started under this
directory. This is a required field.
home_path: /home/oceanbase/ocp-server
memory_size: 1G # The memory size of ocp-express server. The recommend
value is 512MB * (expect node num + expect tenant num) * 60MB.
```

```
admin_passwd: *****
```

```
logging_file_total_size_cap: 10GB # The total log file size of ocp-express server
# logging_file_max_history: 1 # The maximum of retention days the log
archive log files to keep. The default value is unlimited
```

配置项	是否必选	默认值	说明
servers	必选	无	每台机器需要用 - name: 机器标识名 (换行)ip: 机器 IP 指定, 多个机器就指定多次, 机器标识名不能重复。在机器 IP 不重复的情况下, 也可以使用 - <ip> (换行) - <ip> 的格式指定, 此时 - <ip> 的格式相当于 - name: 机器 IP (换行) ip: 机器 IP。
home_path	必选	无	组件的安装路径, 建议在普通用户 admin 下。
memory_size	必选	无	OCP Express 服务器的内存容量, 推荐算法为: $\text{memory_size} = 512\text{MB} + (\text{期望的节点数} * \text{期望的租户数}) * 60\text{MB}$, 期望的租户数需要包含 sys 和 ocp meta 租户本身。
logging_file_total_size_cap	必选	1GB	日志文件总大小, 默认为 1GB。 37.4.2.13 注意 该参数的单位需使用 GB 或 MB, 若使用 G 或 M 单位会报错, 无法成功部署 OCP Express。

admin_passwd	必选	无	OCP Express 登录页面的 admin 账户密码，密码长度 8~32 位，至少包含 2 位数字、2 位大写字母、2 位小写字母和 2 位特殊字符（~!@#%^&*_-+= (){}[]:;,./）。使用 OBD V2.1.0 及之后版本时，未设置的情况下会自动生成随机字符串。
--------------	----	---	--

37.4.3 步骤三：部署 OceanBase 集群

37.4.3.14 说明

本节中所使用命令的详细使用方法可参考 OBD 文档中的 [集群命令组](#)。

1. 部署 OceanBase 集群

```
[admin@test001 ~]$ obd cluster deploy obtest -c default-components.yaml
```

联网情况下，在您执行了 `obd cluster deploy` 命令之后，OBD 将检查您的目标机器是否有部署所需安装包。如果没有安装包，OBD 将自动从 YUM 源获取。

2. 启动 OceanBase 集群

```
[admin@test001 ~]$ obd cluster start obtest
```

当安装了 OCP Express 时，会输出 OCP Express 的访问地址。在阿里云或其他云环境下，安装程序可能因无法获取公网 IP 而输出内网地址，此 IP 非公网地址，您需要使用正确的地址。

3. 查看 OceanBase 集群状态

```
# 查看 OBD 管理的集群列表
```

```
[admin@test001 ~]$ obd cluster list
```

```
# 查看 obtest 集群状态
```

```
[admin@test001 ~]$ obd cluster display obtest
```

4. （可选）修改集群配置

OceanBase 数据库有数百个配置项，有些配置是耦合的，在您熟悉 OceanBase 数据库之前，不建议您修改示例配件文件中的配置。此处示例用来说明如何修改配置，并使之生效。

使用 `obd cluster edit-config` 命令进入编辑模式，修改集群配置。

```
[admin@test001 ~]$ obd cluster edit-config obtest
```

修改配置并保存退出后，OBD 会告知如何使得此次修改生效，保存修改后输入如下。

```
Search param plugin and load ok
Search param plugin and load ok
Parameter check ok
Save deploy "obtest" configuration
Use `obd cluster reload obtest` to make changes take effect.
```

复制 OBD 输出的命令执行即可。

```
[admin@test001 ~]$ obd cluster reload obtest
```

37.4.4 步骤四：连接 OceanBase 集群

此处以使用 OBClient 客户端连接 OceanBase 集群为例：

```
obclient -h<IP> -P<PORT> -u<user_name>@<tenant_name>#<cluster_name> -p -c -A
# example
obclient -h10.10.10.4 -P2883 -uroot@sys#obdemo -p -c -A
```

参数说明：

- `-h`：提供 OceanBase 数据库连接 IP，直连时为 OBServer 节点地址，通过 ODP 连接时为 ODP 地址。
- `-u`：提供 OceanBase 数据库的连接账户，格式有：`用户名@租户名#集群名`、`集群名:租户名:用户名`、`集群名-租户名-用户名` 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`。

37.4.4.15 注意

连接时所用的集群名为配置文件中 `appname` 配置名称，而非部署时的 `deploy name`。

- -P: 提供 OceanBase 数据库连接端口，直连时为 `mysql_port` 配置项的值，通过 ODP 连接时为 `listen_port` 配置项的值。
- -p: 提供 OceanBase 数据库连接密码。
- -c: 表示在 OBClient 运行环境中不要忽略注释。

37.4.4.16 说明

Hint 是特殊的注释，不受 -c 影响。

- -A: 表示在 OBClient 连接数据库时不自动获取统计信息。

更多连接 OceanBase 集群的详细操作可参见 [连接 OceanBase 数据库](#) 章节。

37.4.5 步骤五：创建用户租户

部署 OceanBase 集群之后，建议创建用户租户进行业务操作。sys 租户仅做集群管理使用，不适合在业务场景中使用。

您可以使用如下两种方法创建用户租户：

- 方法一：使用 OBD 创建用户租户。

```
obd cluster tenant create <deploy name> [-n <tenant name>] [flags]
# example
obd cluster tenant create test -n obmysql --max-cpu=2 --memory-size=2G --log-disk-size=3G --max-iops=10000 --iops-weight=2 --unit-num=1 --charset=utf8
```

该命令默认根据集群剩余全部可用资源创建租户，您可通过配置各个参数控制租户占用资源的多少，命令详细使用方法请参考 OBD 手册 [集群命令组](#) 中 `obd cluster tenant create` 命令。

- 方法二：通过命令行创建用户租户，详细操作可参考 [创建租户](#)。

37.5 相关操作

37.5.6 管理集群

您可执行如下命令对 OBD 部署的集群进行管理，详细命令参见 OBD 文档 [集群命令组](#) 一文。

```
# 查看集群列表
obd cluster list
```

```
# 查看集群状态，以部署名为 obtest 为例
obd cluster display obtest

# 停止运行中的集群，以部署名为 obtest 为例
obd cluster stop obtest

# 销毁已部署的集群，以部署名为 obtest 为例
obd cluster destroy obtest
```

37.5.7 卸载 all-in-one

您可参考如下步骤卸载已安装的 all-in-one。

1. 卸载安装包

```
[admin@test001 ~]$ cd oceanbase-all-in-one/bin/
[admin@test001 bin]$ ./uninstall.sh
```

2. 删除环境变量

```
[admin@test001 bin]$ vim ~/.bash_profile
# 删除该文件中的 source /home/admin/.oceanbase-all-in-one/bin/env.sh
```

3. 使修改生效

删除环境变量后，您需重新登录终端，或者执行 `source` 命令使修改生效，可参考如下命令。

```
[admin@test001 ~]$ source ~/.bash_profile
```

38 在 Kubernetes 环境中部署 OceanBase 集群

本文结合示例介绍如何通过 ob-operator 在 Kubernetes 环境中部署 OceanBase 集群。

38.1 前提条件

在开始之前，请确保您已满足以下条件：

- 您有可用的 Kubernetes 集群且至少有 9 个可用 CPU，33 GB 可用内存 和 360 GB 的可用存储空间。
- ob-operator 依赖 cert-manager，请确保您已安装 cert-manager。cert-manager 的安装方法请参考对应的[安装文档](#)。
- 连接 OceanBase 集群时，您需已安装 MySQL 客户端或 OBClient。

38.1.0.1 说明

本文以 ob-operator V2.2.0 为例介绍操作步骤，其他版本可能略有不同。不同版本的操作步骤可参见 [ob-operator 手册](#) 中对应版本内容。

38.2 部署 ob-operator

使用 ob-operator 可以简化 OceanBase 数据库在 Kubernetes 中的部署和运维，您可通过以下两种方式部署 ob-operator。

38.2.1 使用 Helm 部署

您可执行以下命令部署 ob-operator。

```
helm repo add ob-operator https://oceanbase.github.io/ob-operator/  
helm install ob-operator ob-operator/ob-operator --namespace=oceanbase-  
system --create-namespace --version=2.2.0
```

参数说明：

- `--namespace`：命名空间，可根据需要自定义，建议使用 `oceanbase-system`。
- `--version`：ob-operator 版本号，建议使用最新的版本。

38.2.2 使用配置文件部署

您可执行如下命令使用配置文件部署 ob-operator。

```
kubectl apply -f https://raw.githubusercontent.com/oceanbase/ob-operator/2.2.0_release/deploy/operator.yaml
```

38.2.2.1 ob-operator 自定义

如果您需要对 ob-operator 进行自定义修改，可执行如下命令下载配置文件。

```
wget https://raw.githubusercontent.com/oceanbase/ob-operator/2.2.0_release/deploy/operator.yaml
```

根据自身需求修改配置文件后，执行如下命令进行自定义部署。配置文件中的配置项的详细含义可参见 [配置 ob-operator](#)。

```
kubectl apply -f operator.yaml
```

38.3 部署 OceanBase 集群

1. 创建 PVC

ob-operator 在部署 OceanBase 集群时需要创建 PVC 作为 OceanBase 集群的存储。本文中使用 local-path-provisioner 来管理 PVC。部署命令如下：

```
kubectl apply -f https://raw.githubusercontent.com/rancher/local-path-provisioner/v0.0.24/deploy/local-path-storage.yaml
```

更多信息可以参考 [local-path-provisioner](#)。

2. 创建 Namespace

执行如下命令创建部署 OceanBase 集群使用的 Namespace。

```
kubectl create namespace oceanbase
```

创建后可执行 `kubectl get namespace oceanbase` 命令查看是否创建成功（状态为 Active）。

3. 创建默认用户的 Secret

创建 OceanBase 集群之前，您需要先执行如下命令创建好若干 Secret 来存储 OceanBase 集群中的特定用户。

```
kubectl create secret -n oceanbase generic root-password --from-  
literal=password='<root_password>'  
kubectl create secret -n oceanbase generic proxyro-password --from-  
literal=password='<proxyro_password>'
```

`<root_password>` 和 `<proxyro_password>` 分别表示 `root@sys` 用户和 `proxyro@sys` 用户的密码，您需修改为想要配置的密码。创建后可执行 `kubectl get secret -n oceanbase` 命令查看 Secret 资源是否创建成功。

4. 定义 OceanBase 集群

OceanBase 集群通过 yaml 配置文件进行定义，您可执行如下命令创建配置文件，此处以配置文件名为 `obcluster.yaml` 为例。

```
vi obcluster.yaml
```

配置文件内容示例如下。

```
apiVersion: oceanbase.oceanbase.com/v1alpha1  
kind: OBCluster  
metadata:  
  name: obcluster  
  namespace: oceanbase  
spec:  
  clusterName: obcluster  
  clusterId: 1  
  userSecrets:  
    root: root-password  
    proxyro: proxyro-password  
  topology:  
    - zone: zone1  
  replica: 1  
    - zone: zone2
```

```
replica: 1
- zone: zone3
replica: 1
observer:
image: oceanbase/oceanbase-cloud-native:4.2.1.1-101010012023111012
resource:
cpu: 2
memory: 10Gi
storage:
dataStorage:
storageClass: local-path
size: 50Gi
redoLogStorage:
storageClass: local-path
size: 50Gi
logStorage:
storageClass: local-path
size: 20Gi
monitor:
image: oceanbase/obagent:4.2.1-100000092023101717
resource:
cpu: 1
memory: 1Gi
```

主要配置介绍如下，全面的配置项介绍可参见 [集群创建](#)。

配置项	是否必填	说明
metadata.name	必填	集群名。K8s 中资源的名字。
metadata.namespace	必填	集群所在的命名空间。
spec.clusterName	必填	OceanBase 集群名。
spec.clusterId	必填	OceanBase 集群 ID。

spec.userSecrets	必填	OceanBase 集群默认用户的 Secret。
spec.userSecrets.root	必填	OceanBase 集群 root@sys 用户的 Secret 名称, Secret 中需要包含 password 字段。
spec.userSecrets.proxyro	选填	OceanBase 集群 proxyro@sys 用户的 Secret 名称, Secret 中需要包含 password 字段。
spec.topology	必填	OceanBase 集群部署 topo 的定义, 包含对各个 Zone 的定义。
spec.topology[i].zone	必填	OceanBase 集群中 Zone 的名字。
spec.topology[i].replica	必填	OceanBase 集群中 Zone 下包含的 OBSERVER 节点数。
spec.observer.image	必填	所用的 OceanBase 数据库镜像。
spec.observer.resource	必填	OceanBase 集群中 OBSERVER 节点的资源配置。
spec.observer.resource.cpu	必填	OceanBase 集群中 OBSERVER 节点的 CPU 资源配置。建议配置为大于 2 的整数, 小于 2 会引发系统异常。
spec.observer.resource.memory	必填	OceanBase 集群中 OBSERVER 节点的 Memory 资源配置。建议配置值大于 10Gi, 小于 10Gi 会引发系统异常。
spec.observer.storage	必填	OceanBase 集群中 OBSERVER 节点的存储配置。
spec.observer.storage.dataStorage	必填	OceanBase 集群中 OBSERVER 节点的数据存储配置, 建议至少配置为 memory 大小的 3 倍以上。
spec.observer.storage.redoLogStorage	必填	OceanBase 集群中 OBSERVER 节点的 Clog 存储配置, 建议至少配置为 memory 大小的 3 倍以上。
spec.observer.storage.logStorage	必填	OceanBase 集群中 OBSERVER 节点的运行日志存储配置, 建议大小至少配置为 10Gi 以上。
spec.observer.storage.*.storageClass	必填	对于存储配置生效, 用于定义创建 pvc 使用的 storageClass。

spec.observer.storage*.size	必填	对于存储配置生效，用于定义创建 pvc 的容量。
spec.monitor	选填	监控配置，建议开启。ob-operator 会使用 OBAgent 来做监控数据采集，通过对接 Prometheus 可以实现对 OceanBase 集群的状态监控。
spec.monitor.image	必填	OBAgent 所使用的镜像。
spec.monitor.resource	必填	监控容器使用的资源。
spec.monitor.resource.cpu	必填	监控容器使用的 CPU 资源。
spec.monitor.resource.memory	必填	监控容器使用的 memory 资源。

5. 部署 OceanBase 集群

```
kubectl apply -f obcluster.yaml
```

执行如下命令查看 OceanBase 集群状态，当状态变成 `running` 时即表示 OceanBase 集群已成功部署并完成初始化。该过程一般需要几分钟时间，主要耗时过程为镜像拉取和集群初始化。

```
kubectl get obclusters.oceanbase.oceanbase.com obcluster -n oceanbase
```

38.3.3 直连 OceanBase 集群

部署 OceanBase 集群后，您可参考本节步骤直连 OceanBase 集群。推荐部署 ODP 后通过 ODP 连接 OceanBase 集群，部署 ODP 的操作可参见下文 [部署 ODP](#)。

1. 获取 OceanBase 集群 Pod 地址

```
kubectl get pods -n oceanbase -l ref-obcluster=obcluster -o wide
```

示例中的 `oceanbase` 对应 `metadata.namespace` 的值，`obcluster` 对应 `metadata.name` 的值，您需根据实际配置进行替换。

输出如下：

```
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
obcluster-1-zone2-c76d303299a9 2/2 Running 0 4m 10.10.10.1 node-x <none>
obcluster-1-zone3-2cdf3cd8a05e 2/2 Running 0 4m 10.10.10.2 node-x <none>
```

```
<none>
obcluster-1-zone1-94904330202f 2/2 Running 0 4m 10.10.10.3 node-x <none>
<none>
```

2. 连接 OceanBase 集群

您可以使用任一节点的 IP 连接到 OceanBase 集群，对应的连接命令如下。

```
mysql -h10.10.10.1 -P2881 -uroot@sys -p oceanbase -A -c
```

38.4 部署 ODP

ODP 通过 yaml 配置文件进行定义，详细部署步骤如下。

1. 创建 ODP 的配置文件

```
vi obproxy.yaml
```

此处以配置文件名为 `obproxy.yaml` 为例，文件内容如下：

```
apiVersion: v1
kind: Service
metadata:
  name: svc-obproxy
  namespace: oceanbase
spec:
  type: ClusterIP
  selector:
    app: obproxy
  ports:
    - name: "sql"
      port: 2883
      targetPort: 2883
    - name: "prometheus"
      port: 2884
      targetPort: 2884
```

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: obproxy
  namespace: oceanbase
spec:
  selector:
    matchLabels:
      app: obproxy
  replicas: 2
  template:
    metadata:
      labels:
        app: obproxy
    spec:
      containers:
        - name: obproxy
          image: oceanbase/obproxy-ce:4.2.1.0-11
          ports:
            - containerPort: 2883
            - containerPort: 2884
          name: "sql"
          name: "prometheus"
          env:
            - name: APP_NAME
              value: helloworld
            - name: OB_CLUSTER
```

```
value: obcluster
- name: RS_LIST
value: ${RS_LIST}
- name: PROXYRO_PASSWORD
valueFrom:
secretKeyRef:
name: proxyro-password
key: password
resources:
limits:
memory: 2Gi
cpu: "1"
requests:
memory: 200Mi
cpu: 200m
```

主要环境变量配置说明：

- APP_NAME：ODP 的应用名称。
- OB_CLUSTER：ODP 连接的 OceanBase 集群名。
- RS_LIST：OceanBase 集群的 rs_list，格式为 `{ip1}:{sql_port1};{ip2}:{sql_port2};{ip3}:{sql_port3}`，需要根据实际部署情况进行替换。您可直连到 OceanBase 集群中执行 `SELECT GROUP_CONCAT(CONCAT(SVR_IP, ':', SQL_PORT) SEPARATOR ';') AS RS_LIST FROM oceanbase.DBA_OB_SERVERS;` 命令查看，直连 OceanBase 集群的操作可参见 [直连 OceanBase 集群](#)。
- PROXYRO_PASSWORD：需要配置 name 为上文创建的 proxyro@sys 用户的 Secret 名称，Secret 中需要包含 password 字段。

2. 部署 ODP

```
kubectl apply -f obproxy.yaml
```

3. 验证是否部署成功

执行如下命令查看 ODP 的 pod 状态。

```
kubectl get pod -A | grep obproxy
```

输出如下，有两个 obproxy 的 pod。

```
oceanbase obproxy-5cb8f4d975-pmr59 1/1 Running 0 21s
oceanbase obproxy-5cb8f4d975-xlvjp 1/1 Running 0 21s
```

执行如下命令查看 ODP 的服务。

```
kubectl get svc svc-obproxy -n oceanbase
```

输出如下：

```
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
svc-obproxy ClusterIP 10.10.10.1 <none> 2883/TCP,2884/TCP 2m26s
```

38.4.4 通过 ODP 连接 OceanBase 集群

推荐通过 ODP 来连接 OceanBase 集群，在部署好 OceanBase 数据库和 ODP 后，您可通过以下步骤连接 OceanBase 集群。

1. 获取 ODP 的服务连接地址

```
kubectl get svc ${servicename} -n ${namespace}
```

for example

```
kubectl get svc svc-obproxy -n oceanbase
```

输出如下：

```
# output
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
svc-obproxy ClusterIP 10.10.10.1 <none> 2883/TCP,2884/TCP 2m26s
```

2. 连接 OceanBase 集群

您可以通过 CLUSTER-IP 和 PORT 的方式进行连接，对应的连接命令如下。

```
mysql -h10.10.10.1 -P2883 -uroot@sys#obcluster -p oceanbase -A -c
```

38.5 监控 OceanBase 集群

您可使用 OceanBase Dashboard 工具监控 OceanBase 集群。OceanBase Dashboard 是与 ob-operator 配套的白屏运维工具，目前最新版本为 V0.2.1，具有集群管理、租户管理、备份管理、性能监控和终端连接等功能，是在 K8s 环境中配合 ob-operator 监控 OceanBase 集群性能指标的首选工具。

38.5.5 部署 OceanBase Dashboard

推荐使用 Kubernetes 包管理器 Helm 安装 OceanBase Dashboard，确认已经安装了 Helm 之后，执行下面三条命令即可在默认命名空间安装 OceanBase Dashboard。

```
helm repo add ob-operator https://oceanbase.github.io/ob-operator/  
helm repo update ob-operator  
helm install oceanbase-dashboard ob-operator/oceanbase-dashboard --version=0.  
2.1
```

若要在其他命名空间安装 OceanBase Dashboard，可将最后一条安装命令替换为如下命令。

```
helm install oceanbase-dashboard ob-operator/oceanbase-dashboard --version=0.  
2.1 -n <namespace> --create-namespace
```

其中，<namespace> 需设置为待安装 OceanBase Dashboard 的目标命名空间，若目标命名空间不存在，可配置 `--create-namespace` 进行创建。若您的集群支持 LoadBalancer 服务，也可在安装时配置 `--set service.type=LoadBalancer`，指定创建 LoadBalancer 类型的服务。

执行命令后输出如下，则表示 OceanBase Dashboard 部署成功。

```
NAME: oceanbase-dashboard  
LAST DEPLOYED: Wed May 8 11:04:49 2024  
NAMESPACE: default  
STATUS: deployed  
REVISION: 1  
TEST SUITE: None
```

NOTES:

Welcome to OceanBase dashboard

1. After installing the dashboard chart, you can use `port-forward` to expose the dashboard outside like:

```
> kubectl port-forward -n default services/oceanbase-dashboard-oceanbase-dashboard 18081:80 --address 0.0.0.0
```

then you can visit the dashboard on `http://$YOUR_SERVER_IP:18081`

2. Use the following command to get password for default admin user

```
> echo $(kubectl get -n default secret oceanbase-dashboard-user-credentials -o jsonpath='{.data.admin}' | base64 -d)
```

Log in as default account:

Username: admin

Password: <Get from the above command>

部署 OceanBase Dashboard 后 K8s 集群需要一段时间用于拉取所需镜像，可通过如下命令查询 OceanBase Dashboard 是否完成安装。

```
kubectl get deployment oceanbase-dashboard-oceanbase-dashboard
```

输出如下，`READY` 列显示 `1/1` 即表示已经安装完成，可进行后续操作。

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
------	-------	------------	-----------	-----

oceanbase-dashboard-oceanbase-dashboard	1/1	1	1	2m10s
---	-----	---	---	-------

38.5.6 访问 OceanBase Dashboard

OceanBase Dashboard 默认创建的登录账号为 `admin`，通过输出中的第二条命令可获取默认账号的密码，命令如下。

```
echo $(kubectl get -n default secret oceanbase-dashboard-user-credentials -o
jsonpath='{.data.admin}' | base64 -d)
```

您有如下三种方法访问 OceanBase Dashboard，可根据实际情况选择合适的访问方法。

- 通过 NodePort 访问：OceanBase Dashboard 默认创建 NodePort 类型的服务，您可以通过 NodePort 访问 OceanBase Dashboard。
- 通过 LoadBalancer 访问：若您的集群支持 LoadBalancer 服务，可通过 LoadBalancer 访问 OceanBase Dashboard。
- 通过 Port Forward 临时访问：若您的集群节点端口不可访问，无法使用上述两种服务，可通过 Port Forward 临时访问 OceanBase Dashboard。

通过 NodePort 访问

通过 LoadBalancer 访问

通过 Port Forward 临时访问

OceanBase Dashboard 默认创建 NodePort 类型的服务，您可以通过执行如下命令获取服务在节点上暴露的端口。需要注意的是，服务的名称会根据您指定的 Helm Chart 名称变化而变化，具体可参考安装 OceanBase Dashboard 时输出中的第一条指令中的服务名称，本文示例中为 `oceanbase-dashboard-oceanbase-dashboard`。

```
kubectl get svc oceanbase-dashboard-oceanbase-dashboard
```

输出如下：

```
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
oceanbase-dashboard-oceanbase-dashboard NodePort 10.10.10.1 <none> 80:
30176/TCP 13m
```

通过浏览器访问该 K8s 节点的 `PORT` 端口即可打开 OceanBase Dashboard 登录界面。服务的端口号由 K8s 动态分配，节点 IP 可通过执行 `kubectl get nodes -o wide` 命令查看。

您可在安装 OceanBase Dashboard 指定创建 LoadBalancer 类型的服务，也可在安装 OceanBase Dashboard 后，执行如下命令将服务类型修改为 LoadBalancer。

```
kubectl patch -n oceanbase-dashboard svc oceanbase-dashboard-oceanbase-
dashboard --type=merge --patch='{"spec": {"type": "LoadBalancer"}}'
```

修改完成后集群将为 OceanBase Dashboard 服务分配外部 IP，通过该外部 IP 即可访问 OceanBase Dashboard 的页面。修改后等待一段时间再次查看服务信息会看到 External IP 字段已被赋值。

```
kubectl get svc oceanbase-dashboard-oceanbase-dashboard
```

输出如下：

```
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
oceanbase-dashboard-oceanbase-dashboard LoadBalancer 10.10.10.1 10.10.10.2
80:18082/TCP 1d5h
```

在浏览器中输入 `http://10.10.10.2:18082` 即可访问 OceanBase Dashboard，这里仅以 EXTERNAL-IP 为 `10.10.10.2`，PORT 为 `18082` 为例，您需根据实际输出进行替换。

若您集群节点的端口不可访问，无法使用 NodePort 类型的服务暴露 OceanBase Dashboard，也不支持 LoadBalancer 的服务，可以使用 `kubectl port-forward` 命令将 OceanBase Dashboard 暴露到当前机器指定端口进行临时访问，例如执行下面的命令可将 OceanBase Dashboard 暴露在您当前机器（执行这条命令的机器）的 `18081` 端口。

```
kubectl port-forward -n default services/oceanbase-dashboard-oceanbase-
dashboard 18081:80 --address 0.0.0.0
```

通过其他电脑的浏览器访问该机器的 `18081` 端口，可以打开登录界面。若执行上述命令的机器是您的个人电脑，打开浏览器访问 `http://127.0.0.1:18081` 即可访问到服务。

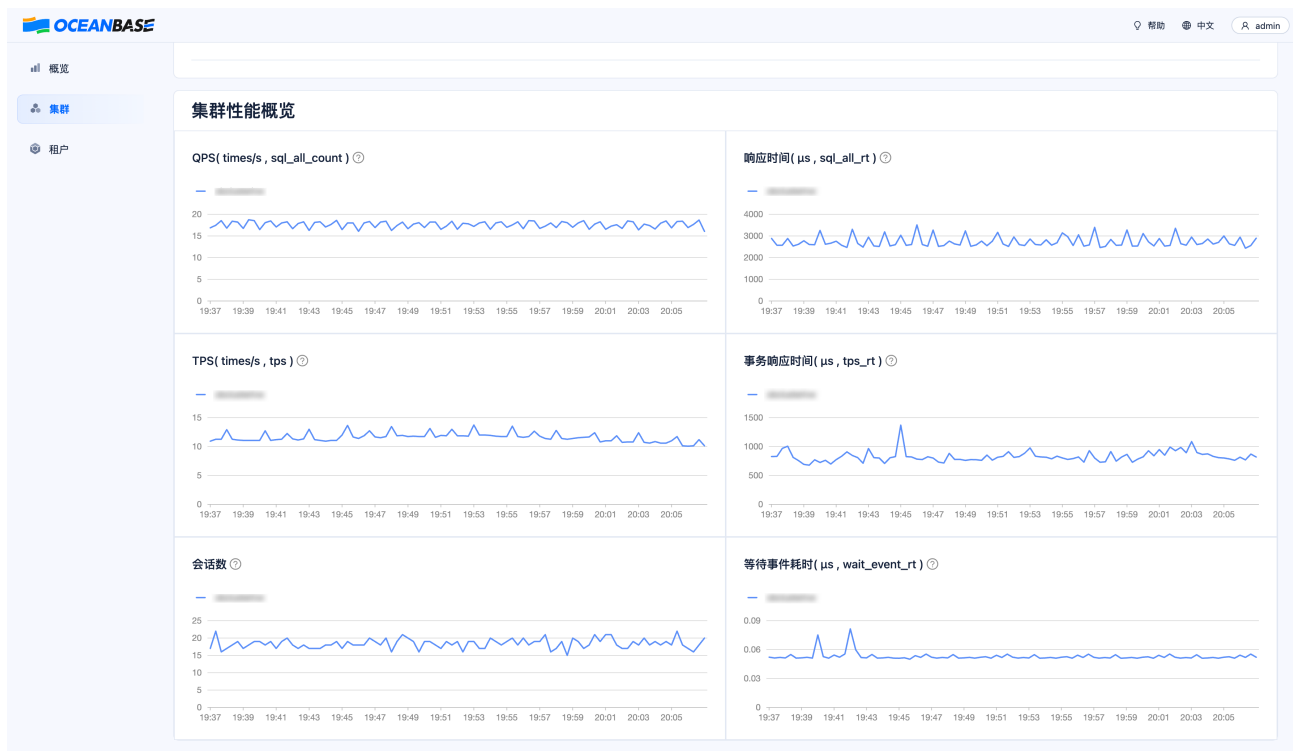
38.5.7 查看监控指标

访问到 OceanBase Dashboard 页面后，您可单击左侧的 **集群** 和 **租户**，选择查看集群或租户的监控信息。页面展示示例如下。

38.5.7.1 说明

除了针对集群和租户的性能指标监控功能外，OceanBase Dashboard 还提供了集群管理、租户管理、备份管理和终端连接等方便用户运维 OceanBase 集群的功能。

集群监控指标如下：



租户监控指标如下：



38.6 相关文档

- [创建租户](#)
- [GitHub 文档](#)
- [GitHub 代码仓库](#)

39 清理旧集群

如果你的机器上曾经部署过 OceanBase 集群，并且现在想重新部署一个新的 OceanBase 集群，请在部署新集群之前先清理旧的集群。

本文分两种场景进行介绍，您可根据自身集群情况选择合适的场景参考。

39.1 清理旧集群建议

在清理旧的集群环境之前，建议您先进行数据备份。这样可以确保数据的完整性和安全性，确保您在将来可能需要时拥有数据的副本，以便进行还原或迁移。

有关 OceanBase 集群数据备份相关信息，请参见 [备份前准备](#)。

39.2 场景一：使用命令行部署的 OceanBase 集群

若您机器中的集群是通过命令行手动部署，可参考本节内容进行清理。

步骤一：停止和卸载服务

停止旧 observer 进程，并确保它们不再运行。然后，卸载相关的数据库软件。步骤如下：

注意

在停止 observer 进程之后，将会导致 OceanBase 数据库无法使用。因此，在执行以下操作之前，请再次确认集群已经不再使用，并且需要备份的数据已经完成备份。请谨慎操作，以免造成数据丢失或系统不可用的情况。

1. 使用以下命令查看 observer 进程是否存在。

```
ps -ef|grep observer
```

2. （可选）停止 observer 进程。

如果您的机器存在 observer 进程，请使用以下命令停止旧 observer 进程：

```
kill -9 `pidof observer`
```

3. 使用以下命令查看已安装的 OceanBase 数据库 RPM 包名称。

```
rpm -qa|grep oceanbase
```

4. 使用以下命令卸载安装的 OceanBase 数据库 RPM 包。

```
sudo rpm -e $rpm_name
```

其中，`$rpm_name` 是要卸载的软件包的名称。

可以使用以下命令查看 RPM 包是否卸载完成。

```
rpm -ql $rpm_name
```

5. 重复步骤 1~4，直至停止和卸载所有 OBServer 节点中的服务。

步骤二：数据库文件清理

删除旧的数据库文件，包括数据文件、日志文件和其他临时文件。

注意

以下删除操作是不可逆的，即删除后的文件和目录将无法恢复。因此执行以下命令前，请确认要删除的目录下数据已不再使用。

1. 使用以下命令删除 observer 进程运行日志盘文件。

```
sudo rm -rf /home/admin/oceanbase/*
```

2. 使用以下命令删除 observer 进程日志盘文件。

```
sudo rm -rf /data/log1/*
```

3. 使用以下命令删除 observer 进程数据盘文件。

```
sudo rm -rf /data/1/*
```

4. 重复步骤 1~3，直至删除所有 OBServer 节点中旧数据库文件。

39.3 场景二：使用 OBD 部署的 OceanBase 集群

若您机器中的集群是通过 OBD 命令部署，可参考本节内容进行清理。

1. 查看集群状态

```
obd cluster list
```

输出如下，详细的状态介绍可参见 [obd cluster list](#)。

```
+-----+  
| Cluster List |
```

```

+-----+-----+-----+
| Name | Configuration Path | Status (Cached) |
+-----+-----+-----+
| test | /home/admin/.obd/cluster/odp | running |
| demo | /home/admin/.obd/cluster/demo | running |
+-----+-----+-----+

```

2. 查看是否存在备租户

此处以 test 集群为例介绍进行操作，您需根据实际情况修改集群名。

```
obd cluster tenant show test -g
```

输出如下，表示不存在备租户，且集群中仅有 sys 租户。若集群中的租户存在备租户，其输出内容和字段含义的详细介绍可参见 [使用 OBD 部署主备租户](#)。

```

+-----+-----+-----+-----+-----+-----+
+
| tenant base info |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+
| tenant_name | tenant_type | compatibility_mode | primary_zone | max_cpu |
min_cpu | memory_size | max_iops | min_iops | log_disk_size | iops_weight |
tenant_role |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+
| sys | SYS | MYSQL | RANDOM | 3.0 | 3.0 | 1024M | 9223372036854775807 |
9223372036854775807 | 2048M | 3 | PRIMARY |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
+-----+-----+

```

3. （可选）主备租户解耦

若集群中的租户存在备租户，销毁集群时会报错，此时可以选择为主备租户解耦，也可以选择强制执行销毁命令，集群销毁后备租户将无法使用。您可根据实际场景选择是否执行本步骤。

主备租户解耦详细操作可参见 [主备租户解耦](#)。

4. 销毁集群

注意

destory 命令不可逆，会在停止 OBCServer 进程的同时清除数据库文件。因此执行以下命令前，请先确认集群已经不再使用，并且需要备份的数据已经完成备份。

此处以 test 集群为例介绍进行操作，您需根据实际情况修改集群名。

```
obd cluster destroy test -f [--igs]
```

-f 表示检查到工作目录下有运行中的进程时，会强制停止进程，销毁集群。

--igs 表示检查到集群中租户在其他集群中存在备租户时，强制执行 obd cluster destroy 命令。

39.4 相关文档

- 如果您正在使用 OceanBase 云平台（OceanBase Cloud Platform, OCP）管理 OceanBase 集群，那么直接在相关操作界面按提示执行删除集群即可，详细操作请参见 [删除集群](#)。
- 如果您是在 Kubernetes 环境中部署 OceanBase 数据库，可通过 ob-operator 管理 OceanBase 集群，删除集群的详细操作请参见 [删除集群](#)。