

PythonOCC镜像使用指南

1 商品说明

PythonOCC 是基于 Python 的开源三维建模与工程计算工具，深度集成 OpenCASCADE 技术核心。提供几何建模、装配体设计、网格生成、有限元分析等功能，适用于 CAD/CAM/CAE 场景。通过 Python 脚本化接口，开发者可快速构建复杂三维应用。

本商品通过鲲鹏服务器进行安装部署PythonOCC 7.5.1 OpenEuler 22.03 Docker镜像版本

2 商品购买

您可以在云商店搜索“PythonOCC三维建模与工程计算工具”。

其中，地域、规格、推荐配置使用默认，购买方式根据您的需求选择按需/按月/按年，短期使用推荐按需，长期使用推荐按月/按年，确认配置后点击“立即购买”。

3 商品资源配置

商品支持**ECS控制台配置**，下面对资源配置的方式进行介绍。

3.1 ECS 控制台配置

3.1.1 准备工作

在使用ECS控制台配置前，需要您提前配置好**安全组规则**。

安全组规则的配置如下：

- 入方向规则放通CloudShell连接实例使用的端口22，以便在控制台登录调试。
- 出方向规则一键放通

3.1.2 创建 ECS

前提工作准备好后，选择ECS控制台配置跳转到购买ECS页面，ECS资源的配置如下图所示：

基础配置

计费模式 

包年/包月  按需计费 竞价计费

按需计费实例不支持备案。[了解备案限制](#) 

区域 

  推荐区域 [华北-北京四](#) | [华南-广州](#) | [华东-上海一](#) |  [华北-乌兰察布一](#) |  [西南-贵阳一](#)

云服务器创建后无法更改区域；不同区域之间内网互不相通，请就近选择靠近您业务的区域，减少网络时延。[如何选择区域](#) 

可用区 

随机分配 可用区1 可用区2 可用区3 可用区7 随机至多可用区

实例

[规格类型选型](#) [业务场景选型](#)

CPU架构 

x86计算 鲲鹏计算

实例筛选 

隐藏售罄的规格

鲲鹏通用计算增强型 鲲鹏内存优化型 鲲鹏超高I/O型

CSDN @p_xcn

操作系统

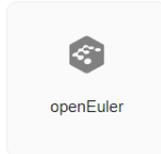
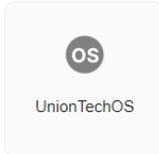
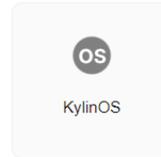
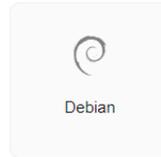
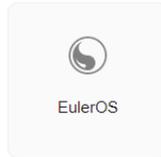
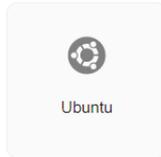
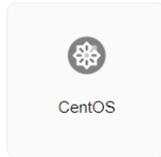
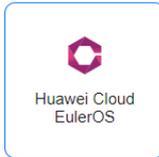
镜像 ?

公共镜像

私有镜像

共享镜像

市场镜像



Huawei Cloud EulerOS 2.0 64bit for kAi2p with HDK 23.0.1 and CANN ...

存储与备份

系统盘 ?

磁盘类型

系统盘大小(GiB)

通用型SSD

40

IOPS上限2,280, IOPS突发上限8,000 [高级设置](#)

⊕ 增加一块数据盘

您还可以挂载 23 块磁盘 (云硬盘)

 开启备份

CSDN @p_xcn

云备份名称: eck-kettle 允许覆盖

描述:

登录凭证:

密钥对:

云备份:

云备份策略 (可选):

高级选项: 现在配置

实例自定义数据脚本:

```
#!/bin/bash
echo 'root:xox' | chpasswd
bash home/mnt.sh
```

购买量: 台 配置费用 ¥0.3988/小时 弹性公网IP配置费用 ¥0.80/GB

值得注意的是:

- VPC您可以自行创建
- 安全组选择3.1.1章节中配置的安全组
- 弹性公网IP选择现在购买, 推荐选择“按流量计费”, 带宽大小可设置为5Mbit/s
- 高级配置需要在高级选项支持注入自定义数据, 所以登录凭证不能选择“密码”, 选

择创建后设置

- 其余默认或按规则填写即可。

4 商品使用

4.1.1 PythonOCC 三维建模与工程计算工具

4.1.2 测试脚本：

1. 测试PythonOCC与occwl安装正确性与版本正确性

操作示例流程

```
> docker exec -it pythonocc /bin/bash
```

```
> cd /home
```

```
> conda activate pythonocc
```

```
> python -c "import OCC; print(OCC.VERSION)"
```

返回7.5.1，并且没有报错

```
> pip show occwl
```

返回中有3.0.0，并且没有报错

2. 测试PythonOCC功能可用

操作示例流程

```
> docker exec -it pythonocc /bin/bash
```

```
> cd /home
```

```
> conda activate pythonocc
```

```
> python test.py
```

执行成功没有报错

4.2 参考文档

- [PythonOCC github官网](#)