

EEE Openstack Machine Setup

EEE Openstack Machine Setup

申请实例
创建实例

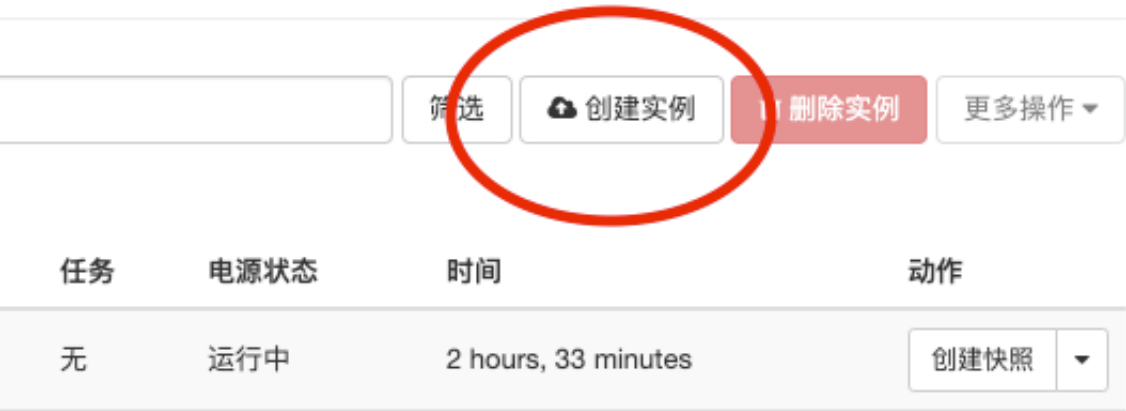
环境配置
首次登陆
设置密码
开启 ssh 密码登陆
GPU 设置
查看设备
cuDNN
Anaconda

申请实例

登陆 <http://portal.eeeos/> 后

创建实例

第一步：点击创建实例



第二步：在弹出的窗口中输入实例名称，任意均可。

创建实例

详情

源

实例类型

网络

网络接口

安全组

密钥对

配置

服务器组

scheduler hint

元数据

请提供实例的主机名，欲部署的可用区域和数量。增大数量以创建多个同样配置的实例。

实例名称

EEE

描述

可用域

nova

数量

1

实例总计 (4 Max)

50%

1 当前用量

1 已添加

2 剩余量

取消

返回

下一步

创建实例

第三步：上一个页面中的“下一步”，输入分配的空间大小。其中“删除实例时删除卷”默认为“不”，建议选“是”。最后，在下方的可用配额中选择 Ubuntu 的箭头，选择后，Ubuntu 会显示已分配。

创建实例

详情

源

实例类型

网络

网络接口

安全组

密钥对

配置

服务器组

scheduler hint

元数据

实例的源是用来创建实例的模板。可以使用一个镜像、一个实例的快照（镜像快照）、一个卷或一个卷快照（如果启用这个功能）。您也可以通过创建一个新卷来选择使用具有持久性的存储。

选择源

镜像

创建新卷

是

不

卷大小 (GB)

500

删除实例时删除卷

是

不

已分配

名称	已更新	大小	类型	可见性	
Ubuntu 18.04 (20200218) (amd64)	3/12/20 1:28 PM	313.61 MB	vmdk	公有	↓

可用配额 3

选择一个

Q 点击这里进行过滤或者全文搜索

名称	已更新	大小	类型	可见性	
CentOS 7 (1907) (x86_64)	2/21/20 2:31 PM	397.77 MB	qcow2	公有	↑
Fedora Cloud 31 (1.9) (x86_64)	3/13/20 2:20 AM	338.89 MB	qcow2	公有	↑
openSUSE Leap 15.1 JeOS (8.10.26) (x86_64)	2/21/20 2:37 PM	215.90 MB	qcow2	公有	↑

取消

返回

下一步

创建实例

第四步：根据需求选择实例配置，点击向上的箭头。

详情

类型管理实例的计算、内存和存储容量的大小。

源

已分配

名称	虚拟内核	内存	磁盘总计	根磁盘	临时磁盘	公有
----	------	----	------	-----	------	----

实例类型 *

从以下可选项中选择一项

网络

▼ 可用配额 7

选择一个

网络接口

Q 点击这里进行过滤或者全文搜索 ✕

安全组

名称	虚拟内核	内存	磁盘总计	根磁盘	临时磁盘	公有
----	------	----	------	-----	------	----

密钥对

> c1-medium	1	12 GB	0 GB	0 GB	0 GB	是	↑
-------------	---	-------	------	------	------	---	---

配置

> c1-large	2	24 GB	0 GB	0 GB	0 GB	是	↑
------------	---	-------	------	------	------	---	---

服务器组

> ag1-xlarge	4	48 GB	0 GB	0 GB	0 GB	是	↑
--------------	---	-------	------	------	------	---	---

scheduler hint

> c1-xlarge	4	48 GB	0 GB	0 GB	0 GB	是	↑
-------------	---	-------	------	------	------	---	---

元数据

> ag1-2xlarge	8	96 GB	0 GB	0 GB	0 GB	是	↑
---------------	---	-------	------	------	------	---	---

> c1-xxlarge	8	96 GB	0 GB	0 GB	0 GB	是	↑
--------------	---	-------	------	------	------	---	---

> ag1-4xlarge	⚠ 16	⚠ 192 GB	0 GB	0 GB	0 GB	是	↑
---------------	------	----------	------	------	------	---	---

✕ 取消

< 返回

下一步 >

创建实例

第五步：直接跳到“密钥对”，如果在 openstack 主页中已经创建了密钥对，那么直接在可用配额中选择即可。因为有些是个人的密钥，建议创建一个项目的密钥对。

创建实例

详情

源

实例类型

网络

网络接口

安全组

密钥对

配置

服务器组

scheduler hint

元数据

密钥对允许您SSH到您新创建的实例。您可以选择一个已存在的密钥对、导入一个密钥对或生成一个新的密钥对。

+ 创建密钥对

↓ 导入密钥对

已分配

正在显示 0 项

名称	类型	指纹
从以下可用的密钥对选择一个		
正在显示 0 项		

▼ 可用配额 5

选择一个

Q 点击这里进行过滤或者全文搜索

正在显示 5 项

名称	类型	指纹	
liuzhengzhe	ssh	d3:fd:c5:bb:d1:2c:8d:55:a2:ff:b9:a6:0f:f2:5b:bf	↑
monomono	ssh	ea:9b:4f:03:8b:55:ec:eb:b6:ff:dd:c5:b5:ed:30:d9	↑
nwong1	ssh	4e:64:3d:4c:11:35:63:76:2d:7c:ee:12:dc:65:6b:86	↑
qihao	ssh	51:42:2a:87:54:5e:6a:af:6d:3e:96:54:cc:98:0c:f6	↑
rfhe	ssh	59:23:22:2d:43:80:46:84:9d:7b:e3:c9:d2:e3:a9:e6	↑

正在显示 5 项

取消

返回

下一步

创建实例

第六步：点击“创建密钥对”，再点击“把私钥复制到剪贴板”。把私钥保存到本地文件。

创建实例

详情

源

实例类型

网络

网络接口

安全组

创建密钥对

密钥对是您实例创建后登陆进去的一种方式。选择一个您易于识别的密钥对名字，名字只能由半角的字母、数字、空格、减号组成。

密钥对名称

EEE

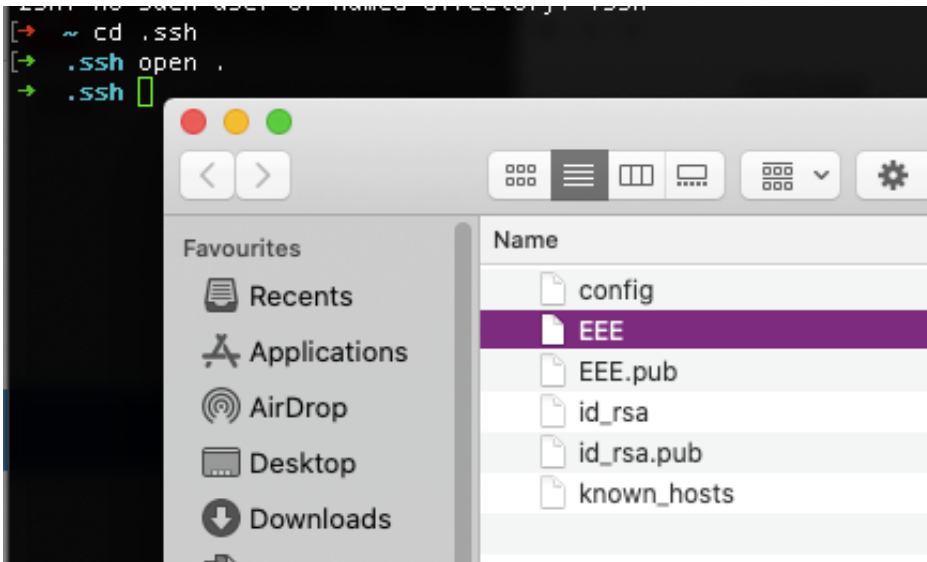
密钥类型

SSH 密钥

创建密钥对

把私钥复制到剪贴板

完成



第七步：点击创建实例，等待若干分钟。如果状态显示为“运行”，即分配实例成功，拷贝其 IP 地址。

正在显示 2 项

	实例名称	镜像名称	IP 地址	实例类型	密钥对	状态	可用域	任务	电源状态	时间	动作	
<input type="checkbox"/>	EEE	-	192.168.149.56	ag1-xlarge	EEE	创建	us-east-1	nova	块设备映射	无状态	0 minutes	编辑实例
<input type="checkbox"/>	mono	Ubuntu 18.04 (20200218) (amd64)	192.168.149.98	ag1-2xlarge	monomono	运行	us-east-1	nova	无	运行中	2 hours, 47 minutes	创建快照

正在显示 2 项

环境配置

首次登陆

```
1  chmod 700 ~/.ssh
2  cd ~/.ssh
3  chmod 600 private_key_name
4
5  ssh -i private_key_name ubuntu@192.168.xxx.xxx
```

设置密码

```
1  sudo su
2  sudo passwd ubuntu
3  # 设置机器密码，注意此时 ubuntu 是 admin 权限
4  exit
5
6  # 之后再次登陆，输入 IP 和 密码即可，不再需要 private_key
7  # 注意保管好 private_key。public_key 可以在 openstack 页面查看
8  ssh ubuntu@192.168.xxx.xxx
```

开启 ssh 密码登陆

```
1 # sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config_bak
2 # 可以选择备份一下
3 sudo vim /etc/ssh/sshd_config
4 # change "PasswordAuthentication no" to "PasswordAuthentication yes"
5
6 # 退出 vim 后
7 systemctl restart sshd
8
9 # 可以本地测试一下
10 ssh ubuntu@192.168.xxx.xxx
```

GPU 设置

```
1 sudo apt-get update
2 sudo apt-get install -y gcc
3 sudo apt-get install -y build-essential
4
5 cd ~
6
7 # 这里以 CUDA 10.2 为例
8 curl -O
  "http://developer.download.nvidia.com/compute/cuda/10.2/Prod/local_install
  ers/cuda_10.2.89_440.33.01_linux.run"
9
10 sudo chmod +x ~/cuda_10.2.89_440.33.01_linux.run
11
12 sudo tee /etc/modprobe.d/nouveau.conf <<!
13 blacklist nouveau
14 options nouveau modeset=0
15 !
16
17 sudo mkdir /opt/NVIDIA
18
19 sudo ~/cuda_10.2.89_440.33.01_linux.run --silent --driver --toolkit --
  toolkitpath=/opt/NVIDIA/cuda-10.2
20
21 # 在 ~/.bashrc 中添加
22 export PATH=/usr/local/cuda/bin:$PATH
23 export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
24
25 # 此时 nvcc 命令可以调用
26 # nvidia-smi 命令也可以显示 GPU 信息
27
28 systemctl set-default graphical.target
29
30 # cuDNN加速 根据自己的需求安装
31 # anaconda 根据自己的需求安装
```

查看设备

```
1 # 查看 硬盘空间
2 df -hl
3
4 # 查看 CPU
5 cat /proc/cpuinfo
6
7 # 查看 内存
8 free -m
```

cuDNN

```
1 # 从 Nvidia 下载 CUDA 对应版本的 cuDNN
2 tar -zxvf cudnn-10.2-linux-x64-v8.0.3.33.tgz
3 # sudo cp cuda/include/cudnn.h /usr/local/cuda/include
4 sudo cp cuda/include/* /usr/local/cuda/include
5
6 sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
7 sudo chmod a+r /usr/local/cuda/include/cudnn.h
  /usr/local/cuda/lib64/libcudnn*
8 cd /usr/local/cuda/lib64/
9
10 ls -lha libcudnn*
11
12 # 创建软链接
13 sudo rm -rf libcudnn.so libcudnn.so.8
14 sudo ln -s libcudnn.so.8.0.3 libcudnn.so.8
15 sudo ln -s libcudnn.so.8 libcudnn.so
16
17 sudo rm -rf libcudnn_adv_infer.so libcudnn_adv_infer.so.8
18 sudo ln -s libcudnn_adv_infer.so.8.0.3 libcudnn_adv_infer.so.8
19 sudo ln -s libcudnn_adv_infer.so.8 libcudnn_adv_infer.so
20
21 sudo rm -rf libcudnn_adv_train.so libcudnn_adv_train.so.8
22 sudo ln -s libcudnn_adv_train.so.8.0.3 libcudnn_adv_train.so.8
23 sudo ln -s libcudnn_adv_train.so.8 libcudnn_adv_train.so
24
25 sudo rm -rf libcudnn_cnn_infer.so libcudnn_cnn_infer.so.8
26 sudo ln -s libcudnn_cnn_infer.so.8.0.3 libcudnn_cnn_infer.so.8
27 sudo ln -s libcudnn_cnn_infer.so.8 libcudnn_cnn_infer.so
28
29 sudo rm -rf libcudnn_cnn_train.so libcudnn_cnn_train.so.8
30 sudo ln -s libcudnn_cnn_train.so.8.0.3 libcudnn_cnn_train.so.8
31 sudo ln -s libcudnn_cnn_train.so.8 libcudnn_cnn_train.so
32
33 sudo rm -rf libcudnn_ops_infer.so libcudnn_ops_infer.so.8
34 sudo ln -s libcudnn_ops_infer.so.8.0.3 libcudnn_ops_infer.so.8
```

```

35 sudo ln -s libcudnn_ops_infer.so.8 libcudnn_ops_infer.so
36
37 sudo rm -rf libcudnn_ops_train.so libcudnn_ops_train.so.8
38 sudo ln -s libcudnn_ops_train.so.8.0.3 libcudnn_ops_train.so.8
39 sudo ln -s libcudnn_ops_train.so.8 libcudnn_ops_train.so
40
41 ls -lha libcudnn*
42 sudo ldconfig
43
44 # 从 Nvidia 下载下面三个文件, 和 cuDNN 文件一起下载
45 sudo dpkg -i libcudnn8_8.0.3.33-1+cuda10.2_amd64.deb
46 sudo dpkg -i libcudnn8-dev_8.0.3.33-1+cuda10.2_amd64.deb
47 sudo dpkg -i libcudnn8-samples_8.0.3.33-1+cuda10.2_amd64.deb
48
49 # 测试 cuDNN 安装
50 cp -r /usr/src/cudnn_samples_v8/ $HOME
51 cd ~/cudnn_samples_v8/mnistCUDNN
52 make clean && make
53 ./mnistCUDNN
54 # 显示 Test passed 即成功

```

Anaconda

```

1 # ./Anaconda3-2020.07-Linux-x86_64.sh
2 # 常规安装步骤, 并选择 conda_init 为 yes
3 # 在 ~/.bashrc 中添加
4 export PATH=/home/ubuntu/anaconda3/bin:$PATH

```

NOTE: 最后在 openstack 管理界面重启一下实例, 重新登陆后一切正常。