

ai-deployer 部署文档

1. 下载 ai-deployer

1. 安装 git

```
#centos
yum install wget -y
#ubuntu
apt-get install wget -y
```

2. 下载ai-deployer

```
#确定工作目录（此文中为/opt），并下载部署工具
cd /opt && wget https://minio.mopaas.com/ai-tool/origin/dev-xjzhu/ai-deployer.tar && tar xvf ai-deployer.tar
```

2. 修改部署配置文件

1. 修改 ai.config

```
#按照规划好的部署方案和环境实际信息修改 ai.config
cd /opt/ai-deployer && vi ai.config
```

#配置文件说明如下

变量名	可选值	说明	备注
EDITION	basic/professional	人工智能平台版本	basic 为标准版； professional 为专业版
VERSION	例： v3.1.4	人工智能平台版本号	需要是实际代码版本号
BASE	例： /opt/ai-deployer	部署工具工作目录	需要是实际部署工具目录， 如上文下载至/opt
MGR_NODE_IP	例： 127.0.0.1	管理节点 IP	实际规划的管理节点 IP
COMPUTE_NODE_IP[1-n]	例： 127.0.0.1 127.0.0.2 127.0.0.3	计算节点 IP	实际规划的计算节点 IP； 多个计算节点依次添加； 多余变量删除而非注释； 管理节点也支持作为计算 节点
GPUSHARE	false/true/vgpu	显卡使用方式	false 为 GPU 独享； true 为 GPU 共享； vgpu 为虚拟 GPU； 标准版只支持 GPU 独享

GPU_TYPE	例: RTX-3070	显卡型号	实际显卡型号 目前支持的显卡型号参考附 1
SERVER_PASSWORD	例: 123456	主机密码	所有主机密码应相同
NODE_PORT_RANGE	例: 30000-38000	K8s 端口范围	可以不更改
STORAGE_TYPE	longhorn/glusterfs	存储节点类型	支持 longhorn/glusterfs 两种存储工具
STORAGE_NODE_IP[1-n]	例: 127.0.0.1 127.0.0.2 127.0.0.3	存储节点 IP	实际规划的存储节点 IP; 多个存储节点依次添加; 多余变量删除而非注释; glusterfs 存储类型节点数需为 1 个或 3 个及以上
RAW_DISK	例: /dev/sdb	存储数据盘	实际数据盘盘符; 数据盘需未经格式化
HARBOR_NODE_IP	例: 127.0.0.1	镜像仓库 IP	实际规划的镜像仓库 IP
HARBOR_NODE_PORT	例: 38080	镜像仓库端口	可以不更改
HARBOR_ADMIN_PASSWORD	例: Harbor12345	镜像仓库管理员密码	自定义管理员密码
MYSQL_NODE_IP	例: 127.0.0.1	数据库 IP	需为部署节点的 IP
MYSQL_NODE_PASSWORD	例: 123456	数据库密码	自定义数据库密码
MINIO_ACCESS_KEY	例: minioadmin	对象存储用户名	自定义对象存储用户名
MINIO_SECRET_KEY	例: minioadmin	对象存储密码	自定义对象存储密码
EFK_NODE_IP	例: 127.0.0.1	日志节点 IP	实际规划的日志节点 IP; 标准版不提供日志服务
PROMETHEUS_IP	例: 127.0.0.1	监控节点 IP	实际规划的监控节点 IP
AI_SERVICE_IP	例: 127.0.0.1	AI 平台服务 IP	如需提供外网访问则配置为部署节点的外网 IP; 如仅提供内网访问则配置为部署节点的内网 IP
USER_PORT	例: 38084	AI 平台用户端端口	可以不更改
MGR_PORT	例: 38085	AI 平台管理端端口	可以不更改

附 1

RTX 系列显卡	GTX 系列显卡	Tesla 系列显卡
RTX-3090	GTX-1080	A100
RTX-3080	GTX-1070	P100
RTX-3070	GTX-1060	V100
RTX-3060		P40
RTX-2080		A10
RTX-2070		T4

RTX-2060		
----------	--	--

3. 检测环境是否符合部署要求

1. 执行 environmentcheck.sh

#根据输出结果判断该环境是否满足部署要求

```
./environmentcheck.sh
```

4. 部署

1. 执行 install.sh

```
./install.sh all
```

2. 部署完成

#查看节点状态

```
kubectl get node
```

#查看 pod 状态，等待所有容器运行

```
kubectl get pod -A
```

#查看后台推送平台框架镜像进度，此步骤耗时很长请耐心等待

#当看到最后输出结果为“已拉取镜像至本地仓库并设置标签”则表明完成

```
cat nohup.out
```

5. 使用

1. 用户端

#访问用户端，初次访问时在页面上获取平台序列号，提供给供应商以获得授权码

#输入授权码后即可正常使用平台

```
http://AI_SERVICE_IP:USER_PORT
```

2. 管理端

#访问管理端，初次访问时在页面上获取平台序列号，提供给供应商以获得授权码

#输入授权码后即可正常使用平台

#管理端默认用户名密码为 admin/123456，登录后请在页面右上角修改密码

http://AI_SERVICE_IP:MGR_PORT

6. 操作

1. 新增节点

#修改 ai.config 配置文件，在 COMPUTE_NODE_IP[1-n]参数下依次添加新节点的信息

#执行环境检测，根据输出结果判断新节点是否满足部署要求

```
./environmentcheck.sh
```

#执行新增节点

```
./operate.sh add_node ai <NEW_NODE_IP>
```

2. 更新AI平台web应用

#由 CI-CD 工具完成最新代码的编译和镜像构建（由供应商完成）

#修改 ai.config 配置文件，修改 VERSION 为最新版本号（如版本号没有变更则不需修改）

#更新 web 应用

```
./operate.sh update_web
```

3. 升级标准版至专业版

#修改 ai.config 配置文件，修改 EDITION 为 professional

#升级至专业版

```
./operate.sh update_edition
```

4. 更新ai-deployer

#拉取最新 ai-deployer 部署工具代码

```
./operate.sh update_ai-deployer <git-username> <git-password>
```