KYLIGENCE ENTERPRISE 快速入门

1 Enterprise Cloud 基本功能

3.1 Kyligence Enterprise 在华为云上正确安装；

3.2 公网 web 访问 KE；

3.3 通过 web 构建 cube 并执行查询。

2 云上安装部署

3.1 操作步骤预览：

序号		步骤	目的	备注
1	物理环境准备	配置 VPC	KE 使用的云上环境逻辑隔离	
2		配置子网段	逻辑网络内部使用	
3		创建安全组	针对特定地址段开放端口	打开 KE 指定端口
4		创建镜像+虚机	将打包好的 KE 安装到虚机上	通过镜像申请虚机；HDP 环境打包在镜像中
5		绑定 EIP		
6	HE 环境准备	调整 YARN 参数	Memory 分配值调整至 12G 左右	调整完毕后重启
7		启动 Hbase master 启动 region server	为启动 KE 准备	操作完成就可启动 KE

如果您觉得操作太麻烦，您也可以请联系我们，我们手把手为您服务”到家” ！

联系电话：021-61060928

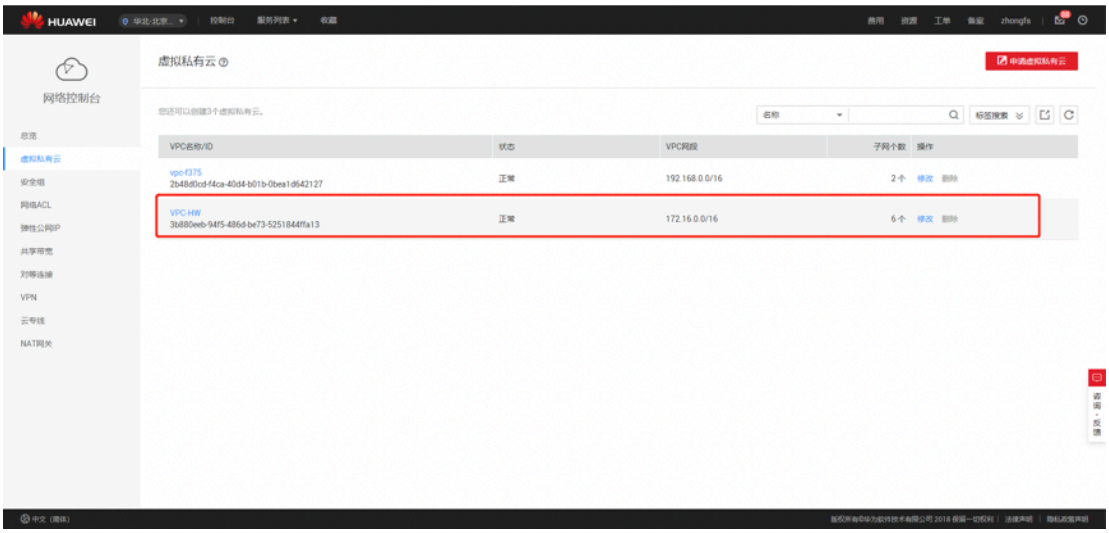
E-mail: info@kyligence.io

2.1 云环境设置

2.1.1 配置 VPC

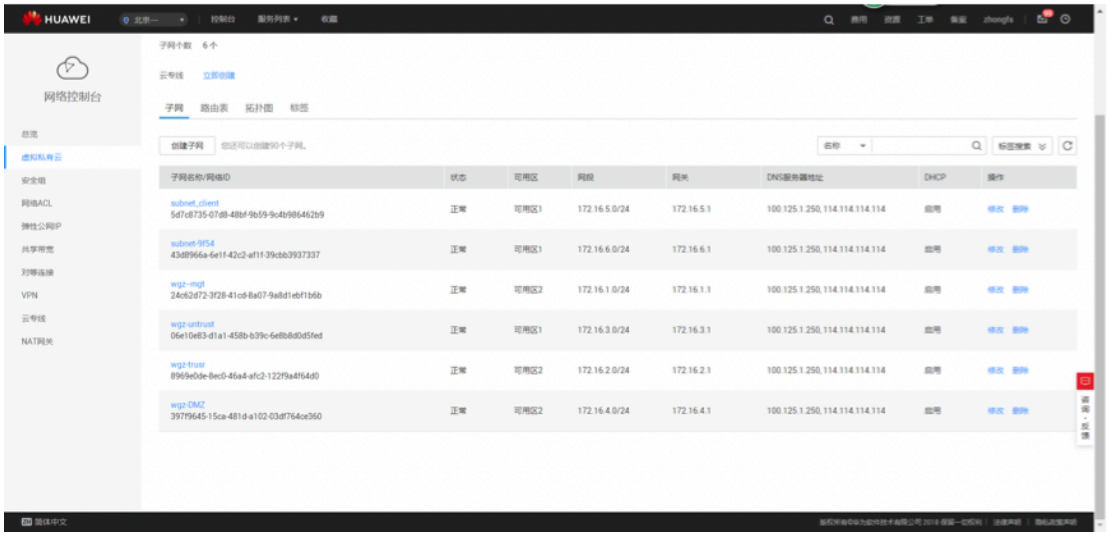
在主页依次点击 “服务列表” →“网络”→“虚拟私有云 VPC”

如下页面，选择 “创建虚拟私有云”，具体操作按网页指引说明执行。



2.1.2 配置子网网段

进入刚建的 “VPC”即 “ 虚拟私有云”，点击“创建子网”，并按网页说明操作。

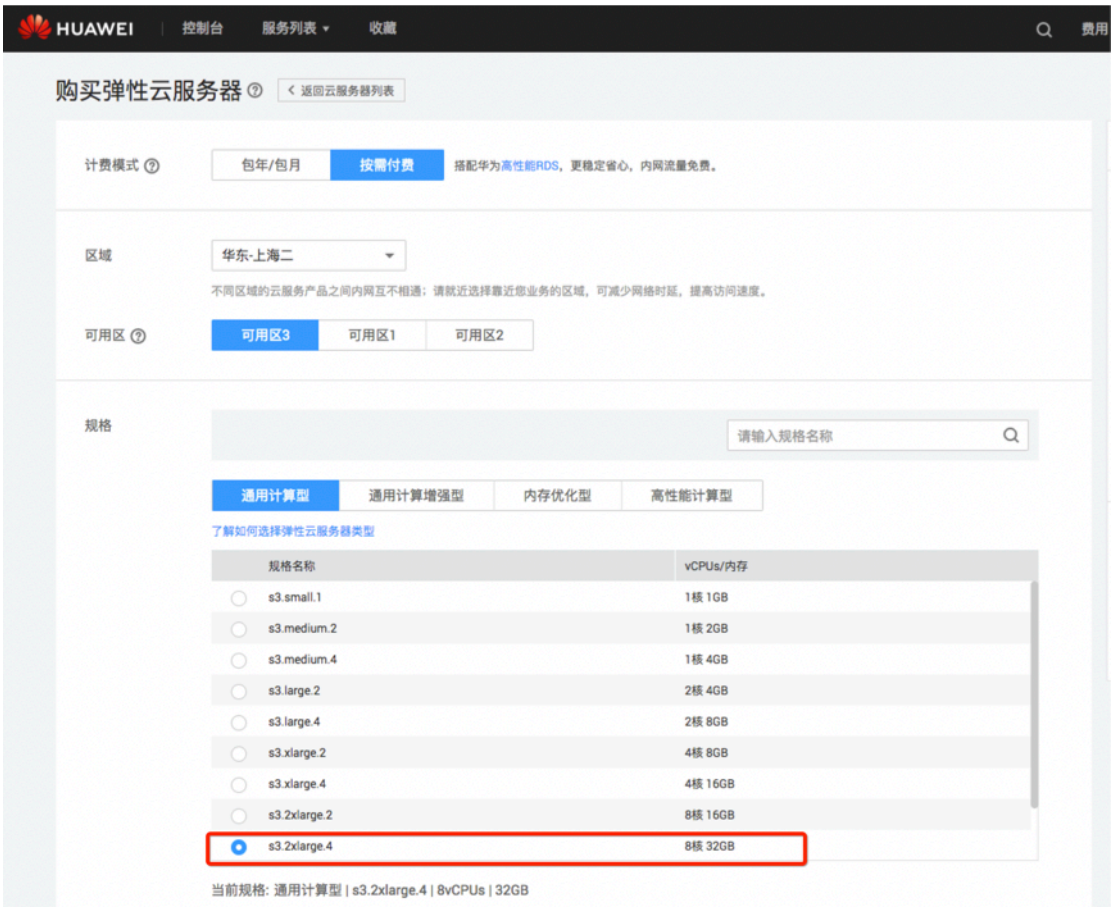


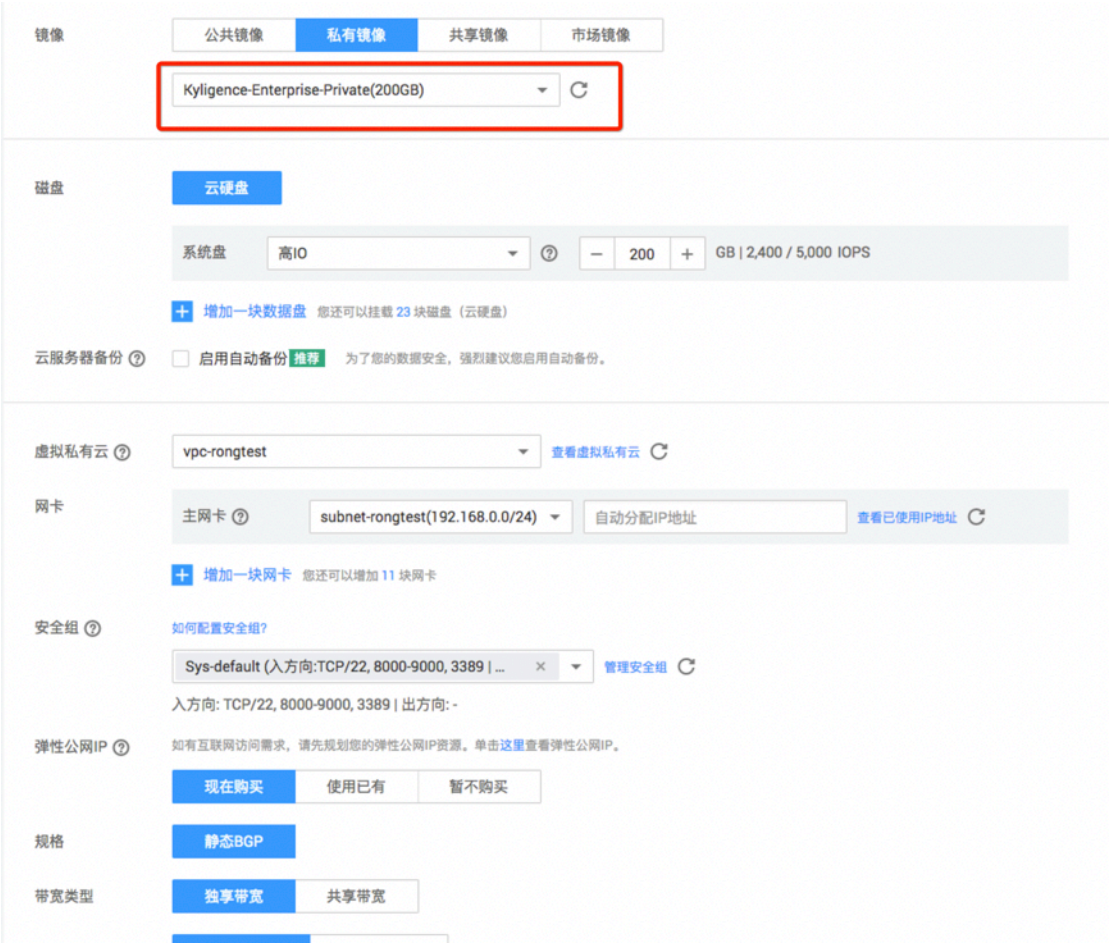
2.1.3 创建安全组

打开 22 、 7070，8080 及 8088 端口，如图：

修改入方向规则

2.1.4 创建 KE 镜像（HDP 沙盒环境打包了 KE），选择申请虚拟服务器



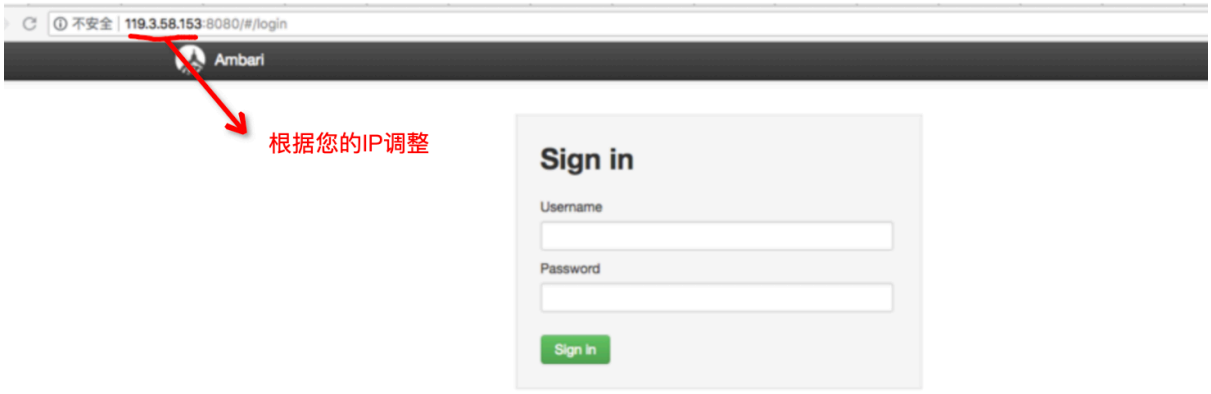


2.1.5 绑定弹性公网 IP

至此，云上物理环境准备完毕。

补充说明：以上操作都在 “控制台” 执行；

以下是 KE 软环境准备要领，即 ambari 环境调整。通过公网 ip 打开 ambari 管理界面(确保 7070，8080，8088 端口已经打开，后续需要 访问) 通过 WEB 访问

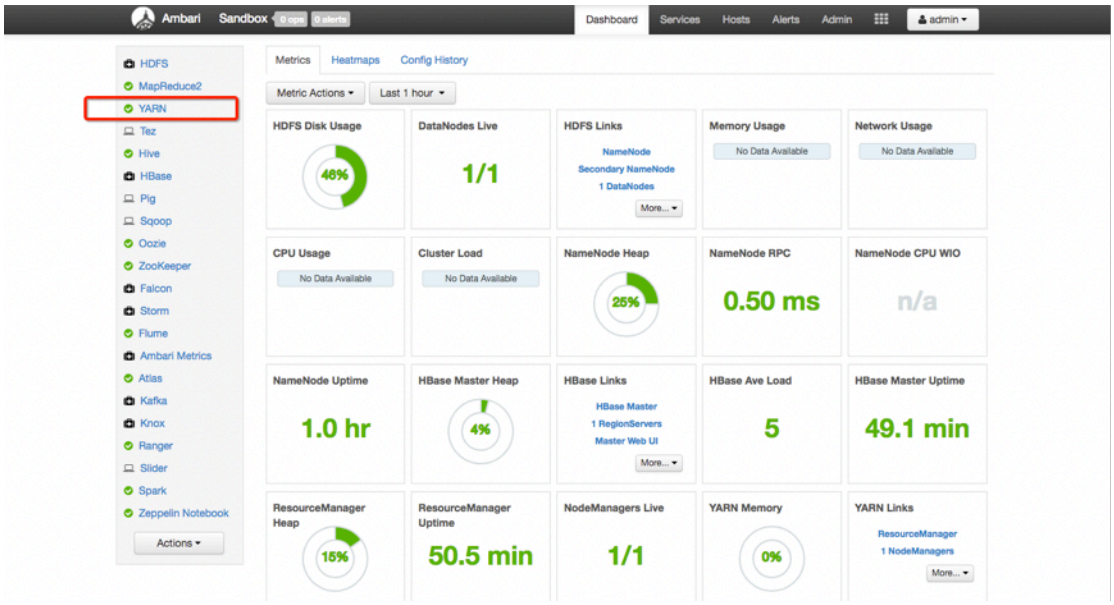


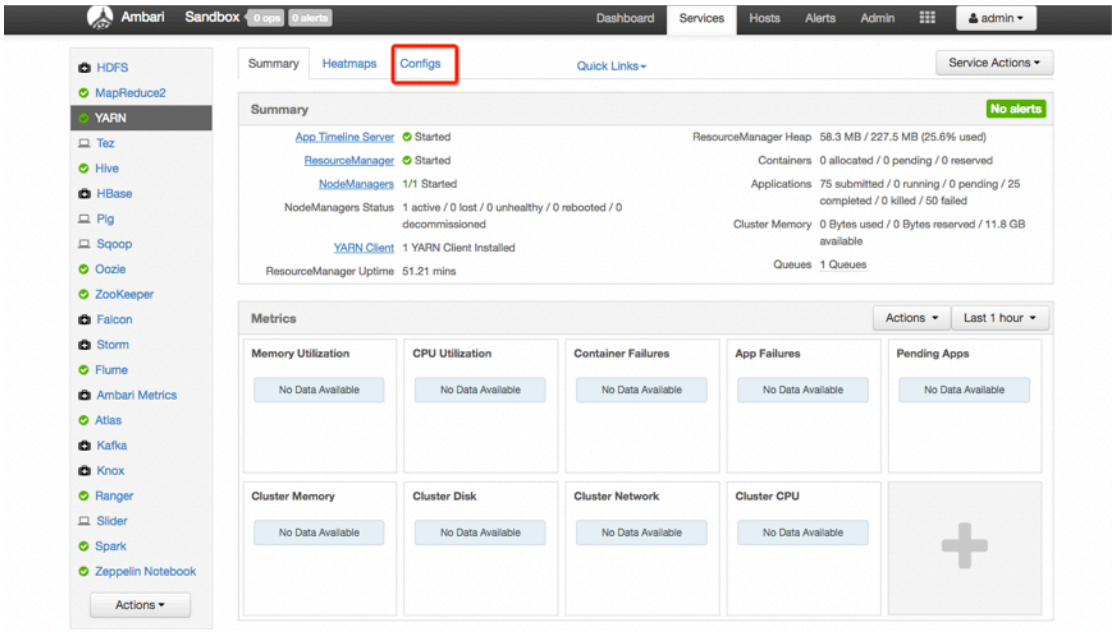
默认用户名 admin 默认密码 admin 可以 ssh 进入虚拟机修改 admin 密码，不想修改密码可以跳过这步(生产环境不建议把 ambari 服务暴露在公网并且不修改默认密码)修改密码命令如下：

ambari-admin-password-reset
ambari-agent restart

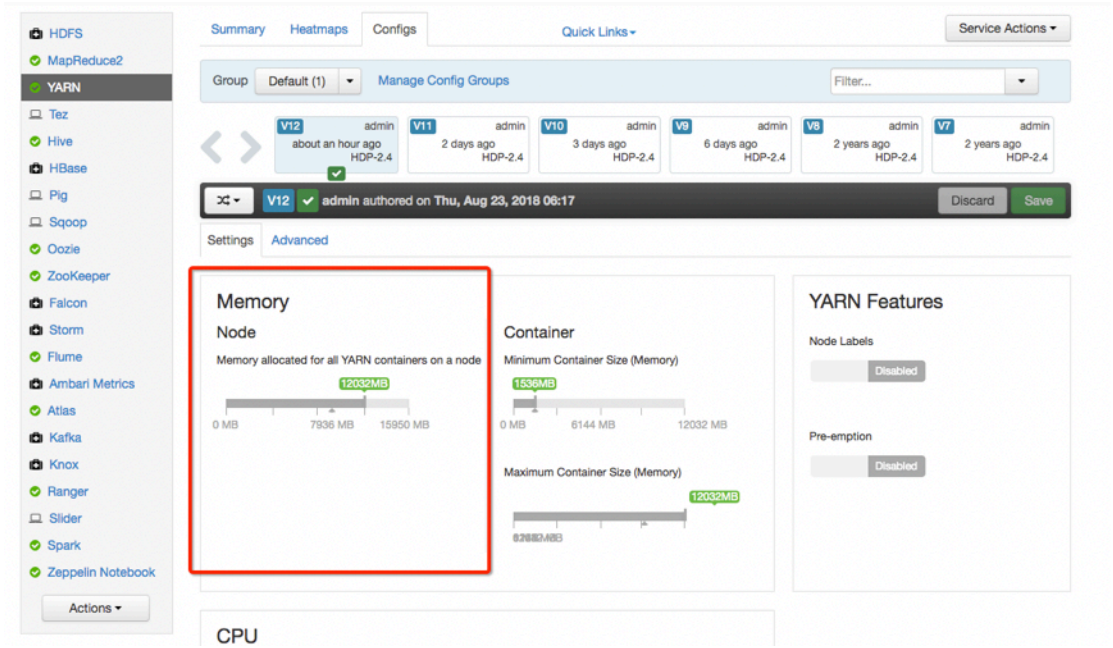
2.2 软环境设置（通过 ambari）

2.2.1 调整分配给 yarn 的内存





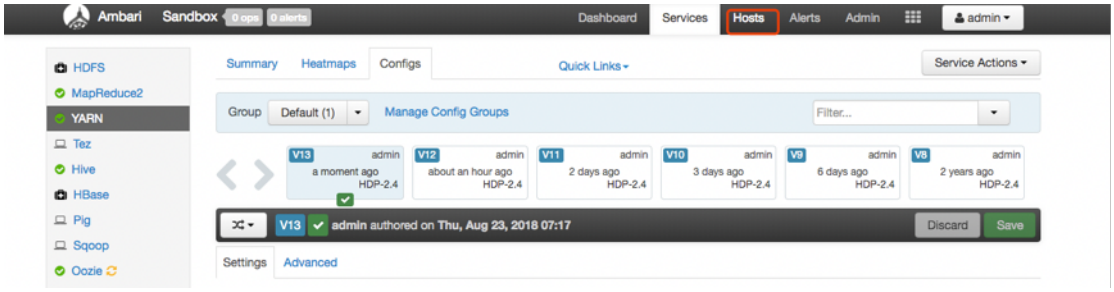
将下图中 Memory 分配值调整至 12G 左右(32G 的虚拟机可以酌情调整的更大)12G 是正 常运行需要的最小安全值，内存太小会导致各项服务运行很慢或者卡死，不建议分配过小内 存并保存。



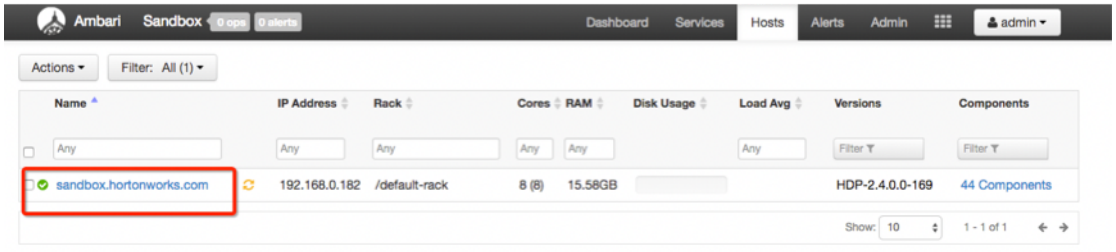
重启收到配置调整影响到的服务。

2.2.2 启动 hbase master 服务和 region server 服务

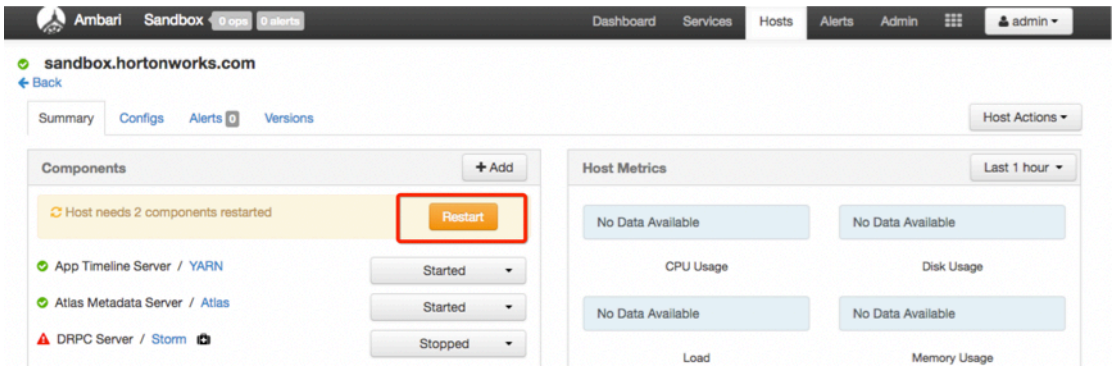
点击红色框内的 hosts



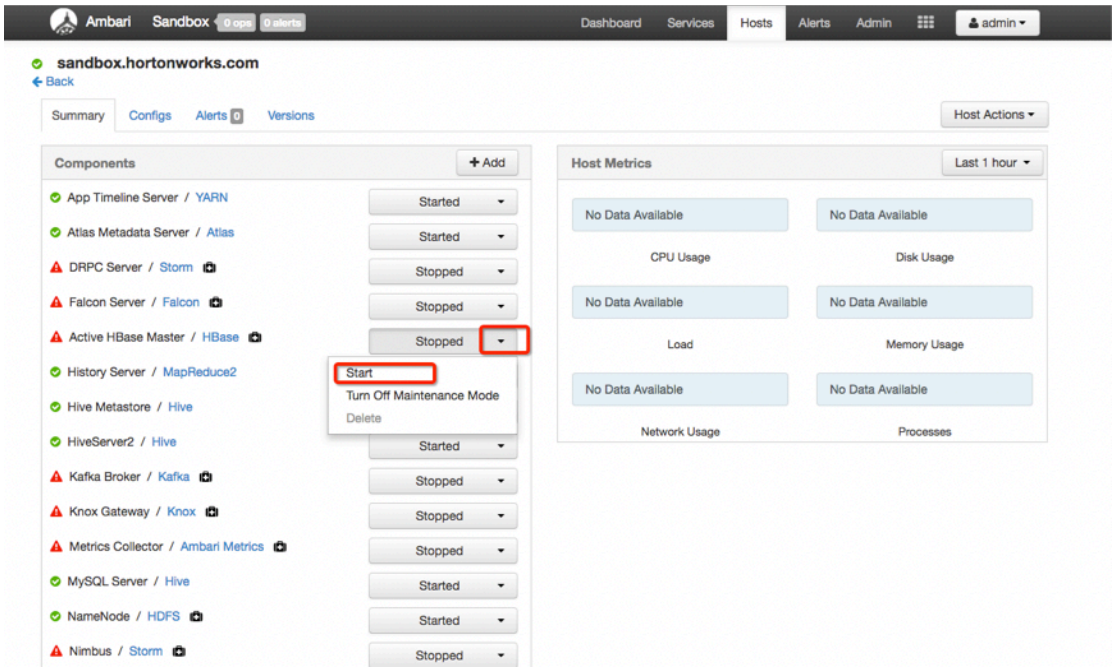
点击红色框内的内容



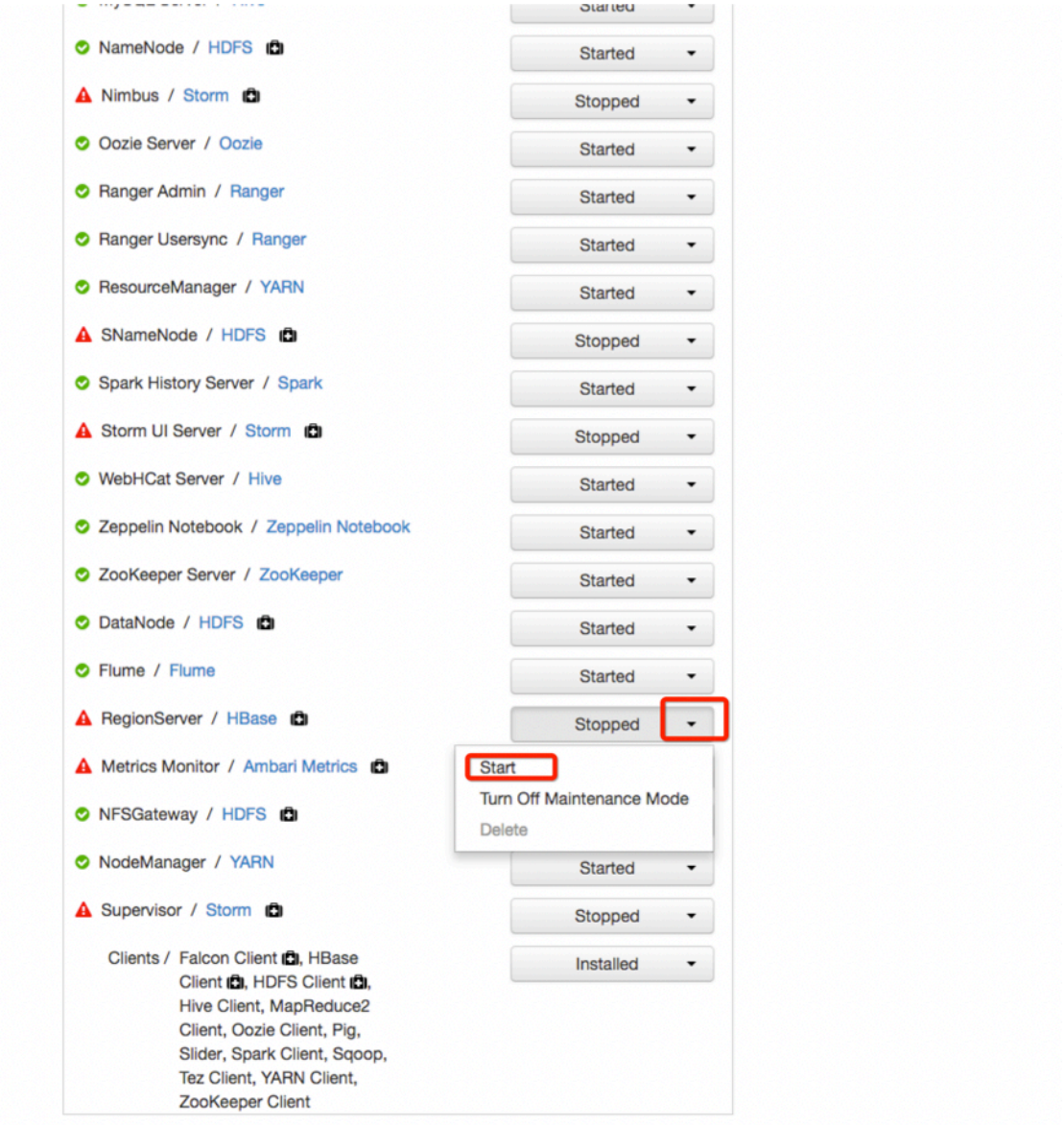
如果出现下图中的情况，请点击 restart 按钮



然后继续启动 hbase master 在 Components 栏里面找到 Hbase master 服务并启动



找到 Region Server 并启动



配置并启动 Kyligence Enterprise ssh 登录到虚拟机
username:root
password: kylin@123

```
Last login: Thu Aug 23 11:20:52 on ttys004
GuoJings-MacBook-Pro-2:~ jing.guo$ ssh root@119.3.58.153
root@119.3.58.153's password:
Last login: Thu Aug 23 03:18:36 2018 from 180.168.34.194
[root@sandbox ~]#
```

进入工作目录:cd /usr/local/kap 执行环境检查脚本：bin/check-env.sh，顺利的话应该会输出如下图所示

```
[root@sandbox bin]# ./check-env.sh

KAP is checking installation environment, log is at /usr/local/kap/logs/check-env.out

Checking Kerberos
.................................................[PASS]
Checking Hadoop Configuration
.................................................[PASS]
Checking Permission of HBase's Table
.................................................[PASS]
Checking Permission of HBase's Root Dir
.................................................[PASS]
Checking Permission of HDFS Working Dir
.................................................[PASS]
Checking Hive Classpath
.................................................[PASS]
Checking Hive Usages
.................................................[PASS]
Checking Java Version
.................................................[PASS]
Checking JDBC Usages
.................................................[PASS]
Checking Legacy Sample Cubes
.................................................[PASS]
Checking ACL Migration Status
.................................................[PASS]
Checking OS Commands
.................................................[PASS]
Checking Ports Availability
.................................................[PASS]
Checking Snappy Availability
.................................................[PASS]
Checking Spark Availability
.................................................[PASS]
Checking Metadata Accessibility
.................................................[PASS]

>   The available yarn RM cores: 7
>   The available yarn RM memory: 10496M
>   The max executor instances can be 2
>   The current executor instances is 1
Checking environment was successful and is now suppressed. To check again, run 'bin/check-env.sh' manually.
```

需要大概 5-10 分钟

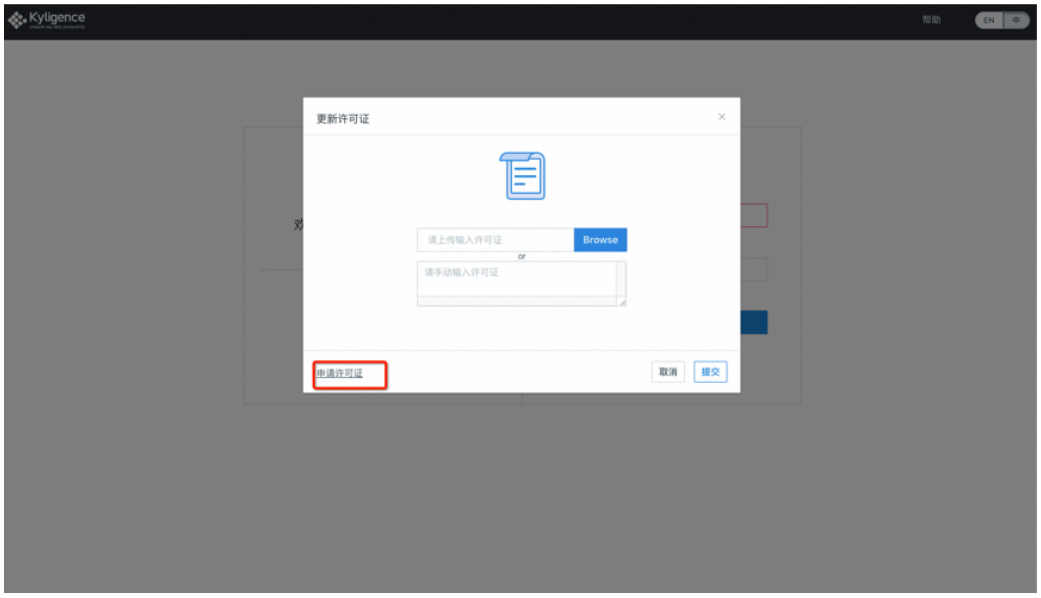
(2)创建 sample cube

```
[root@sandbox kap]# bin/sample.sh
Loading sample data into HDFS tmp path: /tmp/kylin/sample_cube/data
Going to create sample tables in hive to database DEFAULT by cli
WARNING: Use "yarn jar" to launch YARN applications.

Logging initialized using configuration in file:/etc/hive/2.4.0.0-169/0/hive-log4j.properties
OK
Time taken: 1.425 seconds
WARNING: Use "yarn jar" to launch YARN applications.

Logging initialized using configuration in file:/etc/hive/2.4.0.0-169/0/hive-log4j.properties


Time taken: 0.979 seconds
Sample hive tables are created successfully; Going to create sample cube...
18/08/23 04:51:09 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 360 minutes, Emptier interval =
0 minutes.
Moved: 'hdfs://sandbox.hortonworks.com:8020/tmp/kylin/sample_cube' to trash at: hdfs://sandbox.hortonworks.com:8020/user/root/.T
rash/Current
kylin version is 3.0.0.1
mv DEFAULT.KYLIN_ACCOUNT--learn_kylin.json DEFAULT.KYLIN_ACCOUNT--learn_kylin.json
mv: `DEFAULT.KYLIN_ACCOUNT--learn_kylin.json' and `DEFAULT.KYLIN_ACCOUNT--learn_kylin.json' are the same file
mv DEFAULT.KYLIN_CAL_DT--learn_kylin.json DEFAULT.KYLIN_CAL_DT--learn_kylin.json
mv: `DEFAULT.KYLIN_CAL_DT--learn_kylin.json' and `DEFAULT.KYLIN_CAL_DT--learn_kylin.json' are the same file
mv DEFAULT.KYLIN_CATEGORY_GROUPINGS--learn_kylin.json DEFAULT.KYLIN_CATEGORY_GROUPINGS--learn_kylin.json
mv: `DEFAULT.KYLIN_CATEGORY_GROUPINGS--learn_kylin.json' and `DEFAULT.KYLIN_CATEGORY_GROUPINGS--learn_kylin.json' are the same f
ile
mv DEFAULT.KYLIN_COUNTRY--learn_kylin.json DEFAULT.KYLIN_COUNTRY--learn_kylin.json
mv: `DEFAULT.KYLIN_COUNTRY--learn_kylin.json' and `DEFAULT.KYLIN_COUNTRY--learn_kylin.json' are the same file
mv DEFAULT.KYLIN_SALES--learn_kylin.json DEFAULT.KYLIN_SALES--learn_kylin.json
mv: `DEFAULT.KYLIN_SALES--learn_kylin.json' and `DEFAULT.KYLIN_SALES--learn_kylin.json' are the same file
mv DEFAULT.KYLIN_STREAMING_TABLE--learn_kylin_streaming.json DEFAULT.KYLIN_STREAMING_TABLE--learn_kylin_streaming.json
mv: `DEFAULT.KYLIN_STREAMING_TABLE--learn_kylin_streaming.json' and `DEFAULT.KYLIN_STREAMING_TABLE--learn_kylin_streaming.json'
are the same file
Retrieving hive dependency...
Retrieving hbase dependency...
Sample cube is created successfully in project 'learn_kylin'.
Restart Kylin Server or click Web UI => System Tab => Reload Metadata to take effect
```

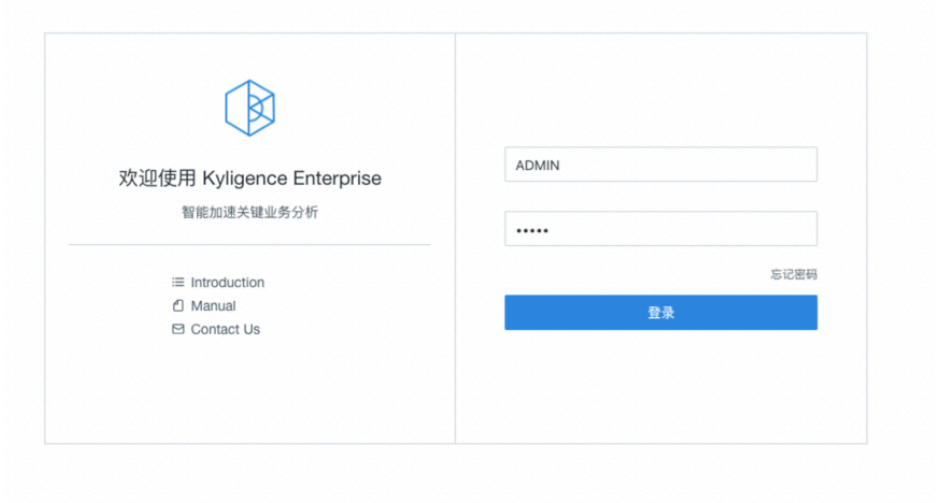
(3)启动 Kyligence Enterprise

```
[root@sandbox kap]# bin/kylin.sh start
Retrieving hive dependency...
Retrieving hbase dependency...
Retrieving kafka dependency...
SPARK_HOME is set to /usr/local/kap/spark/
Retrieving hive dependency...

Use project dict: true.
Start to upgrade the cube storage directory.

Project isolation enabled: true
start to check hdfs
check upgrade flag
cube storage had been upgrade, upgrade had been done! For re-upgrade,delete ${kylin.env.hdfs-working-dir}/success
upgrade cube storage directory successfully

A new KAP server is started by root. To stop it, run 'kylin.sh stop'
Check the log at /usr/local/kap/logs/kylin.log
Web UI is at http://sandbox.hortonworks.com:7070/kylin
[root@sandbox kap]#
```

(4)打开 KE 管理界面



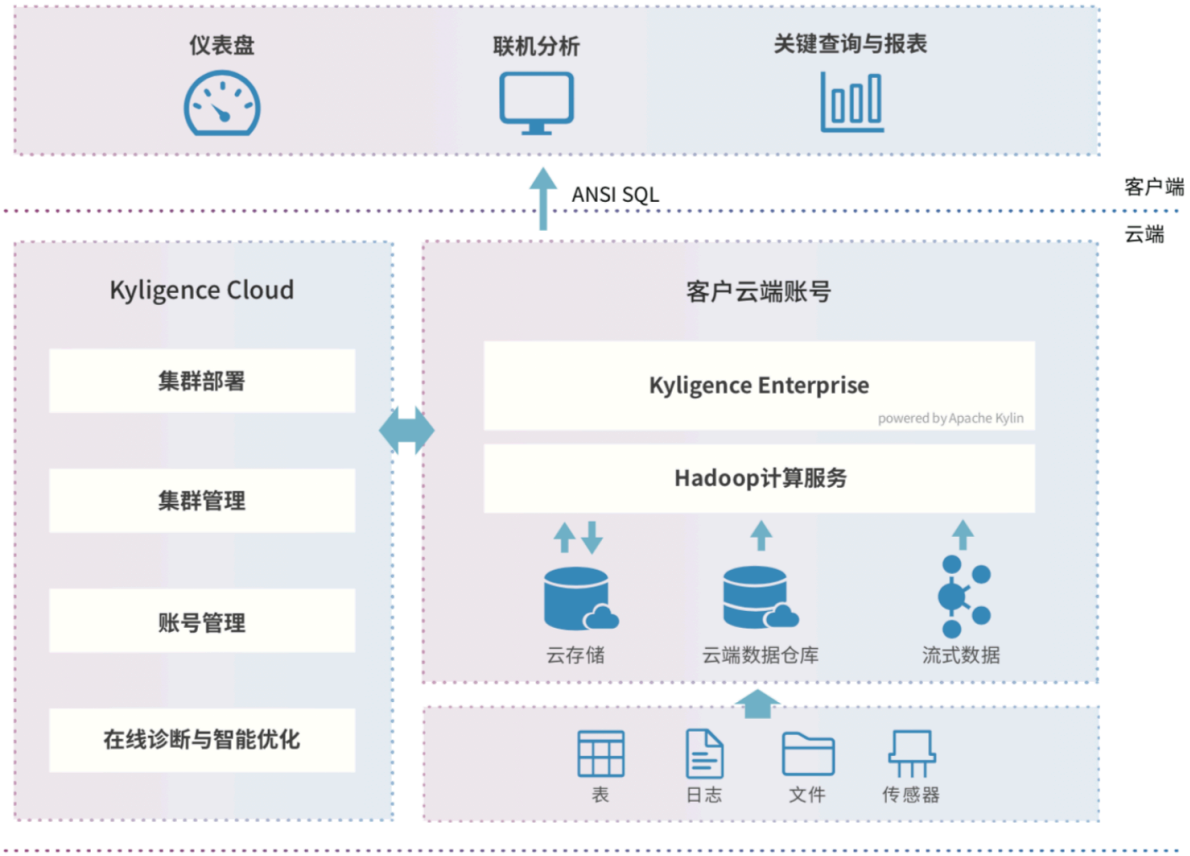
点击申请许可证 向 kyligence 申请好许可证之后就可以进入 KE 管理界面使用 KE 了，我们直接使用试用的 license，跳过申请步骤



3 KE 使用

3.1 KE 与 KC 简介

本章将介绍 Kyligence Enterprise 使用的核心环节：数据导入，设计模型，设计 Cube，优化 Cube，构建流式 Cube，以及高级度量设计等。从下图 KC（Kyligence Cloud）架构图可以看出，KE（Kyligence Enterprise 是 KC 产品的核心，以下会对 KE 的使用做详细说明。



Kyligence Cloud 的架构图

Kyligence Cloud 以您的身份与 华为云进行通信； Kyligence Cloud 使用您提供的帐户信息，在您的 华为云帐户下创建云资源。您使用 Kyligence Enterprise 进行数据建模和分析，所有数据交互均发生在您的虚拟私有云（VPC）内。Kyligence Cloud 不会触碰您的数据。借助于计算与存储分离的架构，您可以方便地对集群进行扩容和缩容，在不需要计算的时候，甚至可以安全地停止集群而不用担心丢失数据。表结构、Cube 模型等元数据存储在您帐户下的 RDS 或 SQL Server 实例中，Cube 数据会保存在华为云存储中。

在该图中，Hadoop / Spark 集群、Kyligence Enterprise、KyAnalyzer 是随需启动的服务。云存储和元数据则具有更长的持久性。