

emr快速部署文档-1.2.0

EMR-1.2.0安装

一,环境准备

- 1.主机规划
- 2.部署包准备

二,安装步骤

1.环境初始化

- 1.1 上传安装包到指定目录
- 1.2 配置主机免密登陆
- 1.3 增加本地解析，并同步到所有节点
- 1.3 配置主机名
- 1.5 执行系统检测/修复
- 1.7 chrony时间同步(可选)

2.starry依赖安装

- 2.1. 配置EMR安装源
- 2.2 各节点安装JDK
- 2.3 mysql安装

3.starry安装启动

- 3.1starry-server安装配置
- 3.3 starry-agent安装配置
- 3.4 页面安装Hadoop集群

EMR-1.2.0安装

一,环境准备

1.主机规划

IP	主机名	配置	角色
192.168.90.226	master1.emr.lcoal	8c 16g 50+200G	Starry-server Starry-agent NN RM HS
192.168.90.229	master2.emr.lcoal	8c 16g 50+200G	Starry-agent NN RM DN NM
192.168.90.223	node1.emr.lcoal	8c 16g 50+200G	Starry-agent DN NM

2.部署包准备

▼

Shell | 复制代码

```

1  #解压后目录如下
2  /data/EMR/
3  |— EMR
4  |— EMR-UTILS-1.1.0.22
5  |— soft
6  |— starry

```

二,安装步骤

1.环境初始化

1.1 上传安装包到指定目录

▼

Shell | 复制代码

```

1  #上传emr-1.1.0.tgz/yum-1.1.0.tgz两个安装包到/data/tools并解压
2  tar zxf /data/tools/EMR-1.2.0.tgz -C /data

```

1.2 配置主机免密登陆

Shell | 复制代码

```
1 cat > /data/EMR/soft/ip.list << EOF
2 192.168.90.226
3 192.168.90.229
4 192.168.90.223
5 EOF
6
7 #创建ssh密钥串
8 ssh-keygen -f ~/.ssh/id_rsa -P "" -t rsa
9 #开通到所有节点的ssh免密通道（按需输入yes和root密码）
10 for i in `cat /data/EMR/soft/ip.list`;do ssh-copy-id $i;done
```

1.3 增加本地解析，并同步到所有节点

Shell | 复制代码

```
1 cat > /data/EMR/soft/hosts << EOF
2 #shuqi_emr
3 192.168.90.226 master1.emr.local master1 yum.emr.local
4 192.168.90.229 master2.emr.local master2
5 192.168.90.223 node1.emr.local node1
6 EOF
7
8 #同步hosts文件到所有节点
9 for i in `cat /data/EMR/soft/ip.list`;do rsync -av /data/EMR/soft/hosts $i
: /etc/;echo -----$i-----;done
10 for i in `cat /data/EMR/soft/ip.list`;do ssh $i "cat /etc/hosts";echo ----
-----$i-----;done
```

1.3 配置主机名

Shell | 复制代码

```
1 ssh master1 "hostnamectl --static set-hostname master1.emr.local"
2 ssh master2 "hostnamectl --static set-hostname master2.emr.local"
3 ssh node1 "hostnamectl --static set-hostname node1.emr.local"
```

1.5 执行系统检测/修复

Shell

复制代码

```
1 #脚本自动进行检测，按需输入。如系统检测有异常项，执行修复，全部通过即可
2 cd /data/EMR/soft
3 #检测
4 sh opstools.sh checklist
5 #修复
6 sh opstools.sh repair
```

1.7 chrony时间同步(可选)

```
1 #所有节点安装chrony服务
2 for i in `cat /data/EMR/soft/ip.list`;do ssh $i "yum -y install chrony";ec
ho -----$i-----;done
3
4 #mastser1作为服务端配置如下
5 vim /etc/chrony.conf
6 server ntp.aliyun.com iburst    #有网络可以写阿里云的ntp 订正时间。没有就写自己 手
动对正
7 driftfile /var/lib/chrony/drift
8 makestep 1.0 3
9 rtcsync
10 allow 192.168.90.0/24
11 local stratum 10
12 logdir /var/log/chrony
13
14 #master2和node1作为客户端配置如下
15 server master1.emr.local iburst
16 driftfile /var/lib/chrony/drift
17 makestep 1.0 3
18 rtcsync
19 logdir /var/log/chrony
20
21
22 #启动所有节点chrony服务并enable
23 for i in `cat /data/EMR/soft/ip.list`;do ssh $i "systemctl start chronyd &
& systemctl enable chronyd";echo -----$i-----;done
24
25 #查看是否同步
26 for i in `cat /data/EMR/soft/ip.list`;do ssh $i "chronyc sources -v";echo
-----$i-----;done
27 ^* 203.107.6.88
28 带星即可
```

2.starry依赖安装

2.1. 配置EMR安装源

```

1  #配置emr目录为yum源 (master1执行)
2  sh /data/EMR/soft/install_nginx.sh
3  #页面访问验证
4  http://master1.emr.local/
5
6
7  #更改repo模版并分发给所有主机
8  for i in `cat /data/EMR/soft/ip.list`;do rsync -av /data/EMR/soft/emr.repo $i:/etc/yum.repos.d;/echo -----$i-----;done
9
10 #验证yum可用性。是否有starry包
11 for i in `cat /data/EMR/soft/ip.list`;do ssh $i "yum list | grep starry";echo -----$i-----;done

```

2.2 各节点安装JDK

```

1  #安装jdk (master1执行)
2  cd /data/EMR/soft/
3  for i in `cat /data/EMR/soft/ip.list`;do echo $i ;rsync -av /data/EMR/soft/{jdk,jdk1.8.0_241} $i:/opt/third/ ;echo -----$i-----;done
4  for i in `cat /data/EMR/soft/ip.list`;do echo $i ;rsync -av /data/EMR/soft/default $i:/usr/java/;echo -----$i-----;done
5  for i in `cat /data/EMR/soft/ip.list`;do echo $i ;rsync -av /data/EMR/soft/mysql-connector-java.jar $i:/usr/share/java/;echo -----$i-----;done
6  for i in `cat /data/EMR/soft/ip.list`;do ssh $i 'echo "export JAVA_HOME=/opt/third/jdk" >> /etc/profile';echo -----$i-----;done
7  for i in `cat /data/EMR/soft/ip.list`;do ssh $i 'echo "export PATH=\$PATH:\$JAVA_HOME/bin" >> /etc/profile';echo -----$i-----;done
8  for i in `cat /data/EMR/soft/ip.list`;do ssh $i 'source /etc/profile &&java -version';echo -----$i-----;done

```

2.3 mysql安装

依赖系统yum源

```
1 #安装mysql (master1执行)
2 sh /data/EMR/soft/install_mysql.sh
```

3.starry安装启动

3.1starry-server安装配置

```
1  #starry-server安装
2  yum install -y starry-server
3
4  #配置license
5  vim /etc/profile
6  export SECURITY_LICENSE=xxxxxxxxxx
7
8  source /etc/profile
9
10 #starry-server配置
11 [root@master1 soft]# starry-server setup
12 Using python /usr/bin/python
13 Setup starry-server
14 Checking SELinux...
15 SELinux status is 'disabled'
16 Customize user account for starry-server daemon [y/n] (n)? n
17 Adjusting starry-server permissions and ownership...
18 Checking firewall status...
19 Checking JDK...
20 [1] Oracle JDK 1.8 + Java Cryptography Extension (JCE) Policy Files 8
21 [2] Custom JDK
22 =====
23 Enter choice (1): 2
24 WARNING: JDK must be installed on all hosts and JAVA_HOME must be valid on all hosts.
25 WARNING: JCE Policy files are required for configuring Kerberos security.
    If you plan to use Kerberos, please make sure JCE Unlimited Strength Jurisdiction Policy Files are valid on all hosts.
26 Path to JAVA_HOME: /opt/third/jdk
27 Validating JDK on Starry Server...done.
28 Check JDK version for Starry Server...
29 JDK version found: 8
30 Minimum JDK version is 8 for Starry. Skipping to setup different JDK for Starry Server.
31 Completing setup...
32 Configuring database...
33 Enter advanced database configuration [y/n] (n)? y
34 Configuring database...
35 =====
36 Choose one of the following options:
37 [1] - PostgreSQL (Embedded)
38 [2] - Oracle
39 [3] - MySQL / MariaDB
```

```

40 [4] - PostgreSQL
41 [5] - Microsoft SQL Server (Tech Preview)
42 [6] - SQL Anywhere
43 [7] - BDB
44
=====
====
45 Enter choice (1): 3
46 Hostname (localhost):
47 Port (3306):
48 Database name (starry):
49 Username (starry):
50 Enter Database Password (Shuq1-1298):
51 Re-enter password:
52 Configuring starry database...
53 Should starry use existing default jdbc /usr/share/java/mysql-connector-ja
va.jar [y/n] (y)? y
54 Configuring remote database connection properties...
55 WARNING: Before starting Starry Server, you must run the following DDL dir
ectly from the database shell to create the schema: /var/lib/starry-serve
r/resources/Starry-DDL-MySQL-CREATE.sql
56 Proceed with configuring remote database connection properties [y/n] (y)?
y
57 Extracting system views...
58 starry-admin-1.1.0.0.0.jar
59
60 Starry repo file doesn't contain latest json url, skipping repoinfos modif
ication
61 Adjusting starry-server permissions and ownership...
62 Starry Server 'setup' completed successfully.
63
64
65
66 #初始化 starry 数据库连接配置
67 starry-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-co
nector-java.jar
68 #初始化starry库
69 mysql -ustarry -pShuq1-1298 starry < /var/lib/starry-server/resources/Sta
rry-DDL-MySQL-CREATE.sql;
70 #启动starry-server
71 starry-server start
72 echo "source /etc/profile && starry-server restart" >> /etc/rc.d/rc.local

```

```

[root@master1 soft]# starry-server setup
Using python /usr/bin/python
Setup starry-server...
Checking SELinux...
SELinux status is 'disabled'
Customize user account for starry-server daemon [y/n] (n)? n
Adjusting starry-server permissions and ownership...
Checking firewall status...
Checking JDK...
[1] Oracle JDK 1.8 + Java Cryptography Extension (JCE) Policy Files 8
[2] Custom JDK
Enter choice (1): 2
WARNING: JDK must be installed on all hosts and JAVA_HOME must be valid on all hosts.
WARNING: JCE Policy files are required for configuration. If you plan to use Kerberos, please make sure JCE Unlimited Strength Jurisdiction Policy Files are valid on all hosts.
Path to JAVA_HOME: /opt/third/jdk
Validating JDK on Starry Server...done.
Check JDK version for Starry Server...
JDK version found: 8
Minimum JDK version is 8 for Starry. Skipping to setup different JDK for Starry Server.
Completing setup...
Configuring database...
Enter advanced database configuration [y/n] (n)? y
Configuring database...
Choose one of the following options:
[1] - PostgreSQL (Embedded)
[2] - Oracle
[3] - MySQL / MariaDB
[4] - PostgreSQL
[5] - Microsoft SQL Server (Tech Preview)
[6] - SQL Anywhere
[7] - ODB
Enter choice (1): 3
Hostname (localhost):
Port (3306):
Database name (starry): starry
Username (starry): starry
Enter Database Password (bigdata):
Re-enter password:
Configuring starry database...
Should starry use existing default jdbc /usr/share/java/mysql-connector-java.jar [y/n] (y)? y
Configuring remote database connection properties...
WARNING: Before starting Starry Server, you must run the following DDL directly from the database shell to create the schema: /var/lib/starry-server/resources/Starry-DDL-MySQL-CREATE.sql
Proceed with configuring remote database connection properties [y/n] (y)? y
Extracting system views...
..starry-admin-1.2.0-1.jar
Error extracting starry-views-package-1.2.0-1.jar

Starry repo file doesn't contain latest json url, skipping repoinfos modification
Adjusting starry-server permissions and ownership...
Starry Server 'setup' completed successfully.
[root@master1 soft]#

```

默认n

自定义jdk地址

填入jdk绝对路径

自定义数据库

数据库相关信息

使用获取到的mysql驱动jar

确认写入配置文件

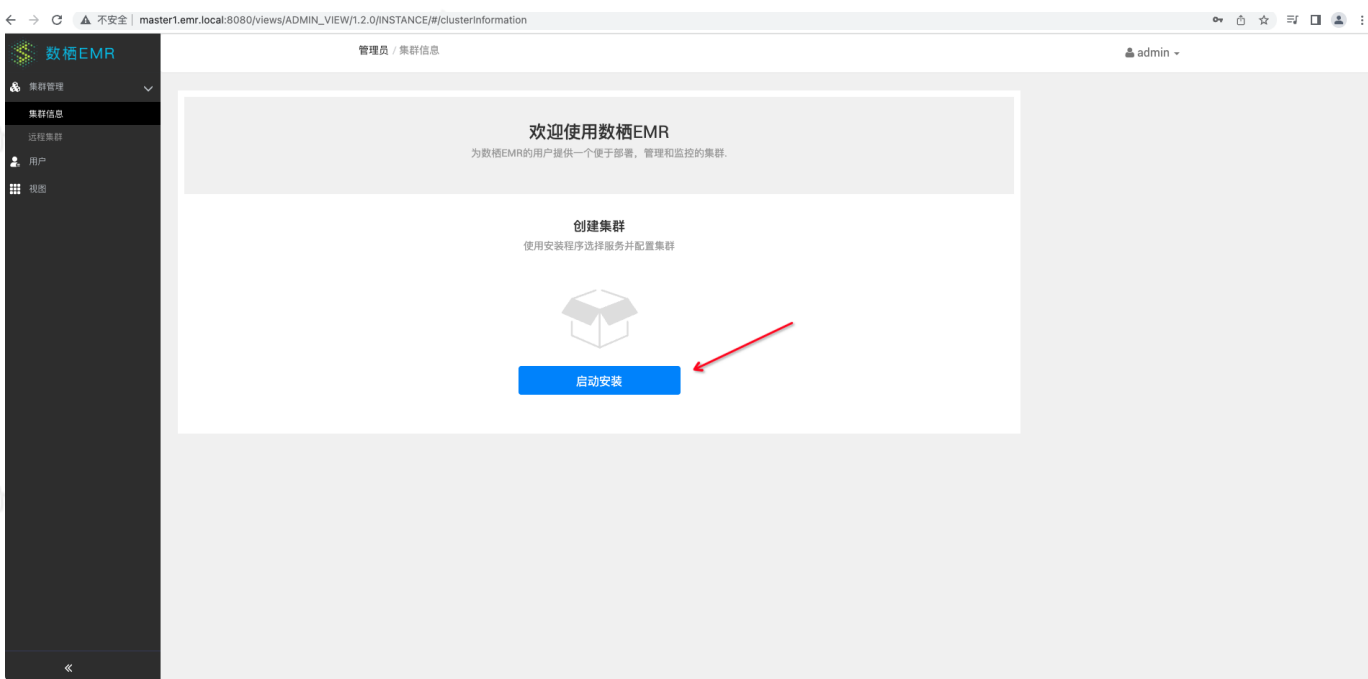
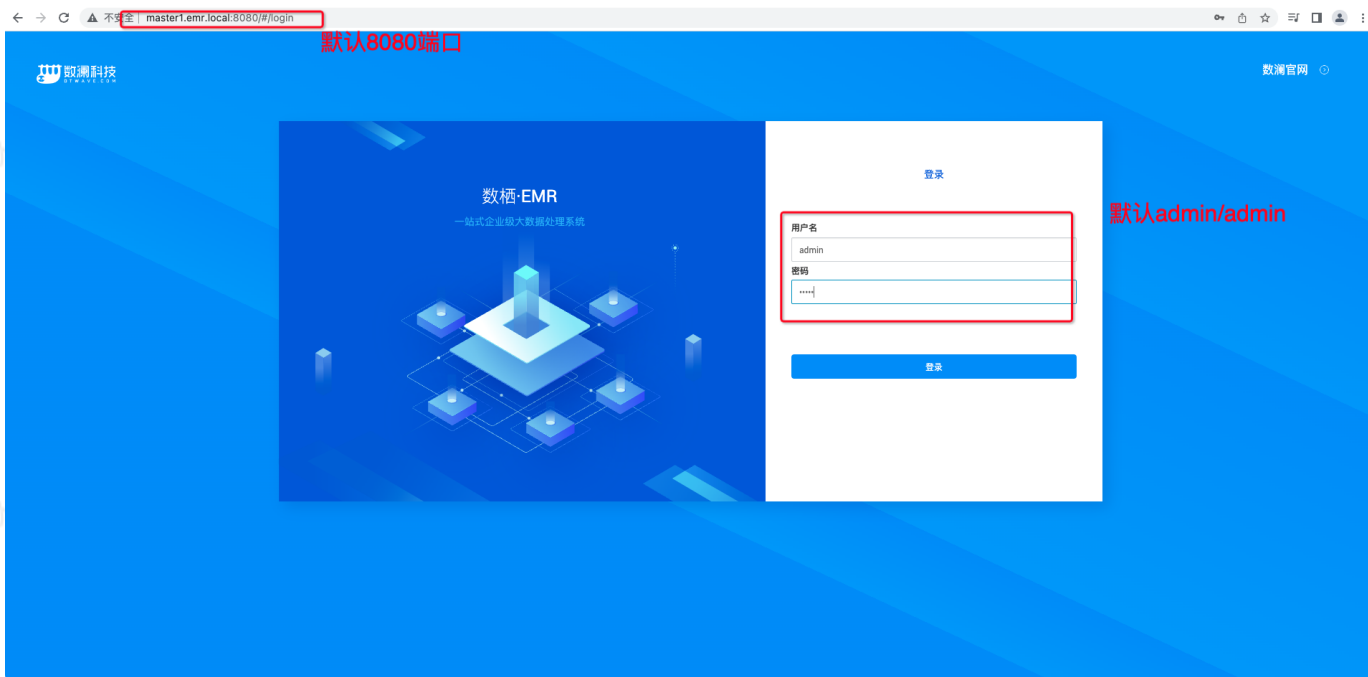
3.3 starry-agent安装配置

```

1 #starry-agent安装
2 for i in `cat /data/EMR/soft/ip.list`;do ssh $i 'yum install -y starry-agent';echo -----$i-----;done
3
4 #更改客户端配置
5 for i in `cat /data/EMR/soft/ip.list`;do ssh $i "sed -i '/^hostname=/c hostname=master1' /etc/starry-agent/conf/starry-agent.ini";echo -----$i-----;done
6 for i in `cat /data/EMR/soft/ip.list`;do ssh $i 'starry-agent start';echo -----$i-----;done

```

3.4 页面安装Hadoop集群

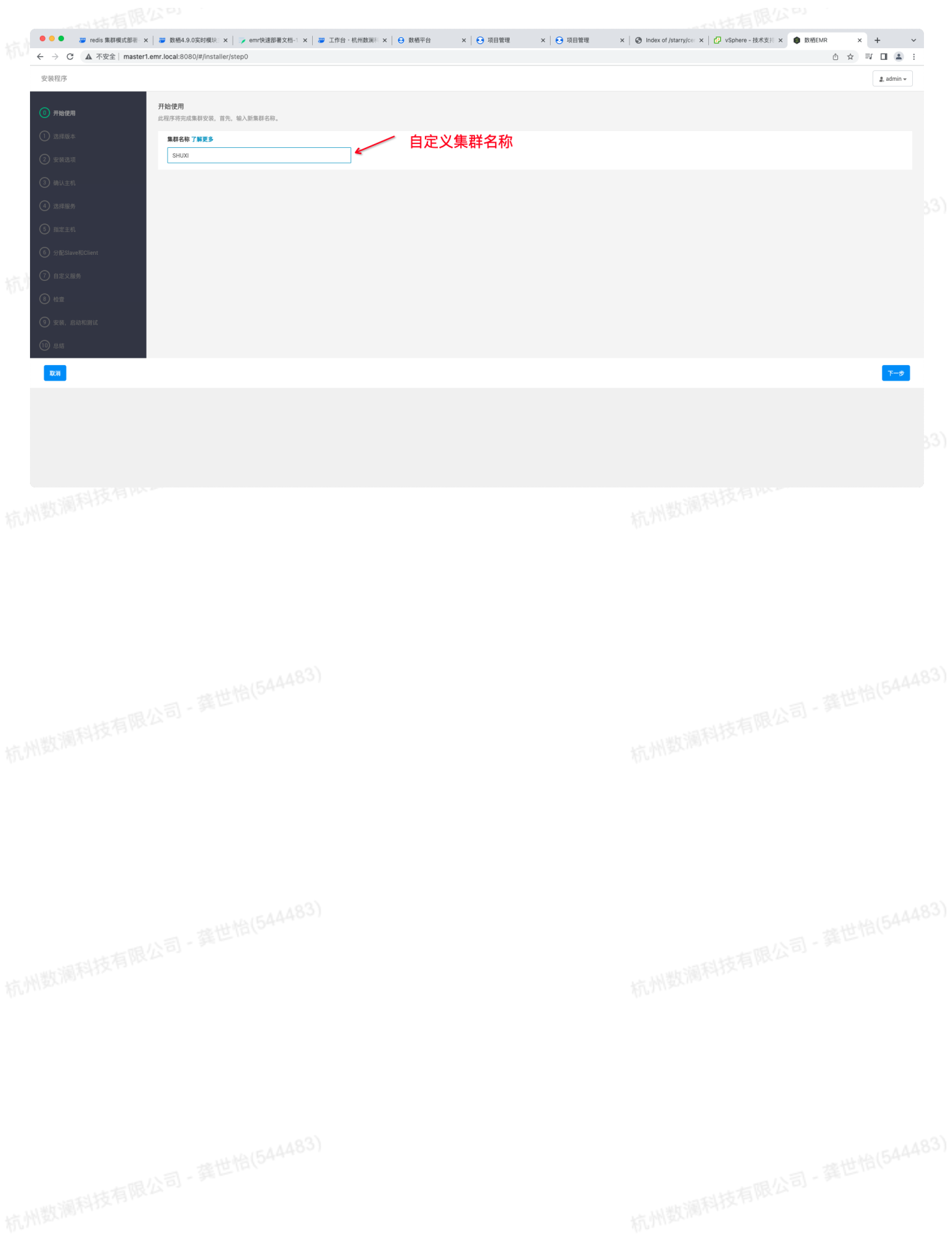


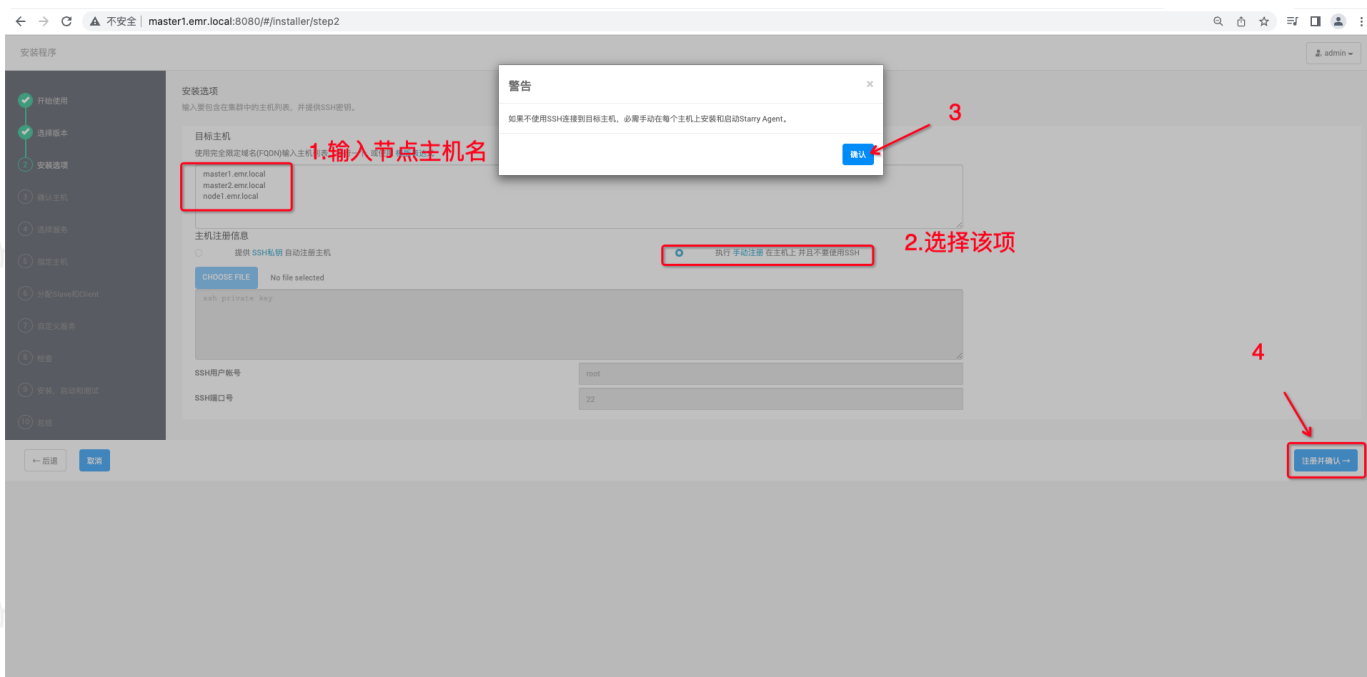
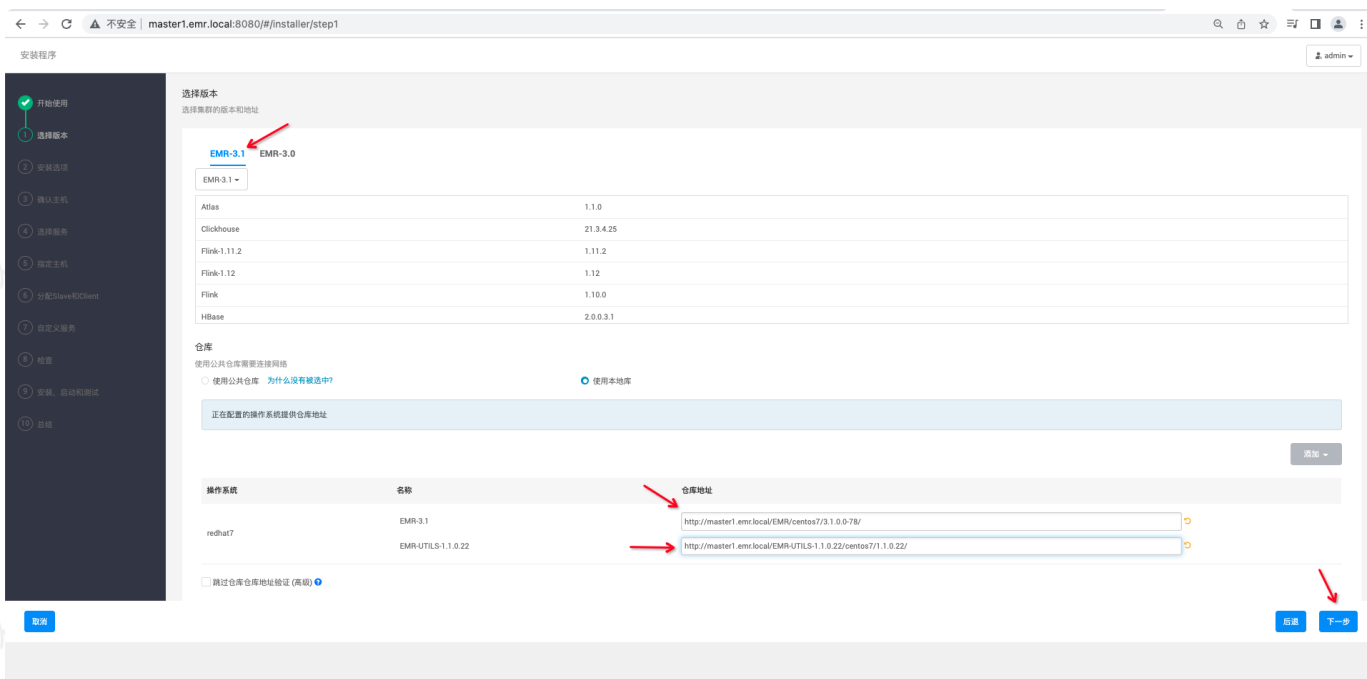
杭州数澜科技有限公司 - 龚世怡(544483)

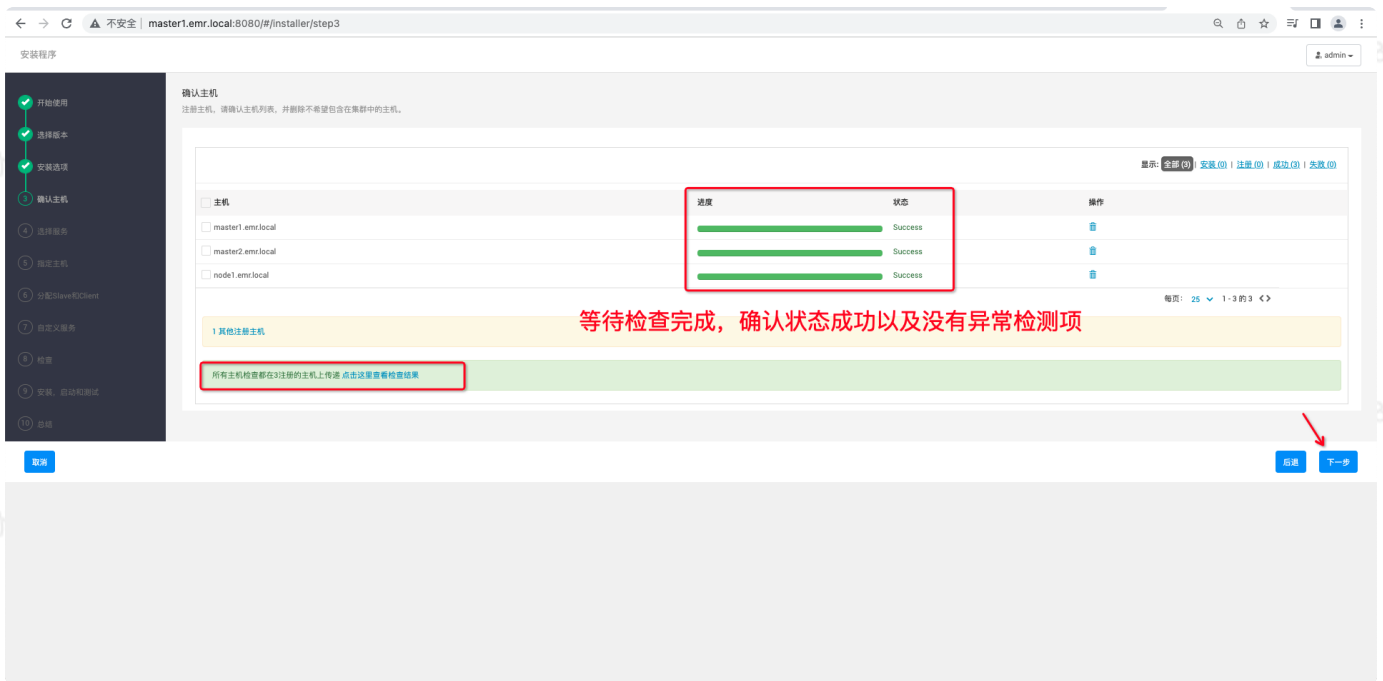
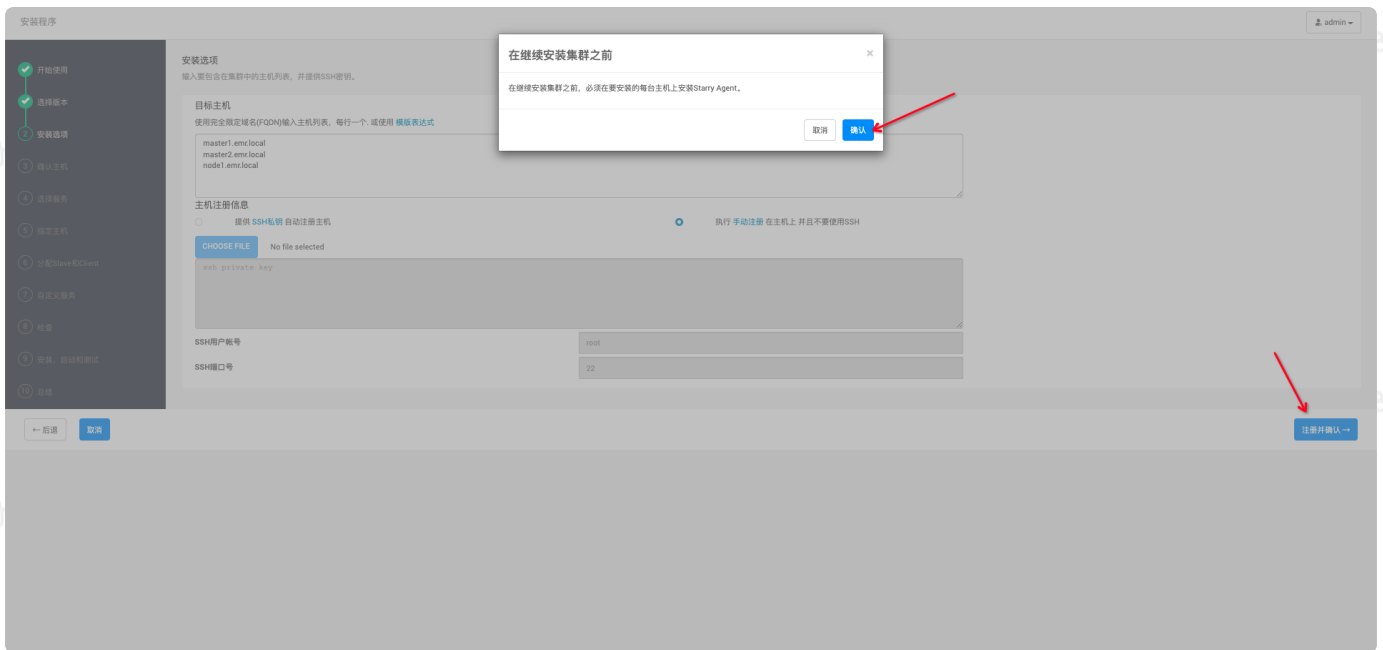
杭州数澜科技有限公司 - 龚世怡(544483)

公司 - 龚世怡(544483)

公司 - 龚世怡(544483)







第一次初始化安装集群时，spark3、flink-1.12、clickhouse不能一起部署，该服务依赖部分Hadoop集群的服务正常才能部署成功。可以在部署完成后，再增加服务。

杭州数澜科技有限公司

10 选择

按需勾选要部署的服务

☒	Hive	3.0.0.3.1	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
☒	HBase	2.0.0.3.1	Non-relational distributed database and centralized service for configuration management & synchronization
☒	ZooKeeper	3.6.3.3.1	Centralized service which provides highly reliable distributed coordination
☐	Storm	1.2.1	Apache Hadoop Stream processing framework
☒	Infra Solr	0.1.0	Core shared service used by Starry managed components.
☒	Starry Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
☒	Atlas	1.1.0	Atlas Metadata and Governance platform
☒	Kafka	1.0.0.3.1	A high-throughput distributed messaging system
☐	Log Search		
☐	Ranger	1.2.0.3.1	Comprehensive security for Hadoop
☐	Ranger KMS	1.2.0	Key Management Server
☒	Spark2	2.3.0	Apache Spark 2.3 is a fast and general engine for large-scale data processing.
☒	Clickhouse	21.3.4.25	open source distributed column-oriented DBMS.
☒	Flink	1.10.0	Apache Flink 1.10.0 is a framework and distributed processing engine for stateful computations over unbounded and bounded data streams.
☒	Flink-1.11.2	1.11.2	Apache Flink 1.11.2 is a framework and distributed processing engine for stateful computations over unbounded and bounded data streams.
☒	Flink-1.12	1.12	Apache Flink 1.12 is a framework and distributed processing engine for stateful computations over unbounded and bounded data streams.
☐	Hudi	0.8.0	Apache Hudi ingests and manages storage of large analytical datasets over DFS (hdfs or cloud stores).
☒	Hue	4.10.0	Hue is a graphical user interface to operate and develop applications for Apache Hadoop.
☒	Iceberg	0.11.0	Apache Iceberg is an open table format for huge analytic datasets.
☐	Kafka2	2.7.0	Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.
☒	Kudu	1.7.0	Apache Kudu is an open source distributed data storage engine that makes fast analytics on fast and changing data easy.
☒	Presto	0.257	Presto is an open source distributed SQL query engine for running interactive analytic queries against data sources of all sizes ranging from gigabytes to petabytes.
☒	Spark3	3.0.2	Apache Spark 3.0.2 is a fast and general engine for large-scale data processing.
☒	Trino	358	Trino is an open source distributed SQL query engine for running interactive analytic queries against data sources of all sizes ranging from gigabytes to petabytes.

部署

按需勾选要部署的服务

自测

下一步

杭州数澜科技有限公司

10 选择

按照前期规划，配置服务所在节点

指定主机

将master组件分配给要运行它们的主机

NameNode: master1.emr.local (15.5 GB, 8 cores)

SNameNode: master2.emr.local (15.5 GB, 8 cores)

ResourceManager: master1.emr.local (15.5 GB, 8 cores)

Timeline Service V2.0 Reader: master1.emr.local (15.5 GB, 8 cores)

YARN Registry DNS: master1.emr.local (15.5 GB, 8 cores)

Timeline Service V1.5: master2.emr.local (15.5 GB, 8 cores)

History Server: master2.emr.local (15.5 GB, 8 cores)

Hive Metastore: master2.emr.local (15.5 GB, 8 cores)

HiveServer2: master2.emr.local (15.5 GB, 8 cores)

HBase Master: master1.emr.local (15.5 GB, 8 cores)

ZooKeeper Server: node1.emr.local (15.5 GB, 8 cores)

ZooKeeper Server: master1.emr.local (15.5 GB, 8 cores)

ZooKeeper Server: master2.emr.local (15.5 GB, 8 cores)

Infra Solr Instance: master1.emr.local (15.5 GB, 8 cores)

Metrics Collector: node1.emr.local (15.5 GB, 8 cores)

Grafana: master1.emr.local (15.5 GB, 8 cores)

按照前期规划，配置服务所在节点

master1.emr.local (15.5 GB, 8 cores)

NameNode

ResourceManager

Timeline Service V2.0 Reader

YARN Registry DNS

HBase Master

ZooKeeper Server

Infra Solr Instance

Grafana

Atlas Metadata Server

Kafka Broker

Ranger KMS Server

Spark2 History Server

Spark3 History Server

Hue Server

Clickhouse Server

Presto coordinator

Trino coordinator

Kudu Master

master2.emr.local (15.5 GB, 8 cores)

SNameNode

Timeline Service V1.5

History Server

Hive Metastore

HiveServer2

ZooKeeper Server

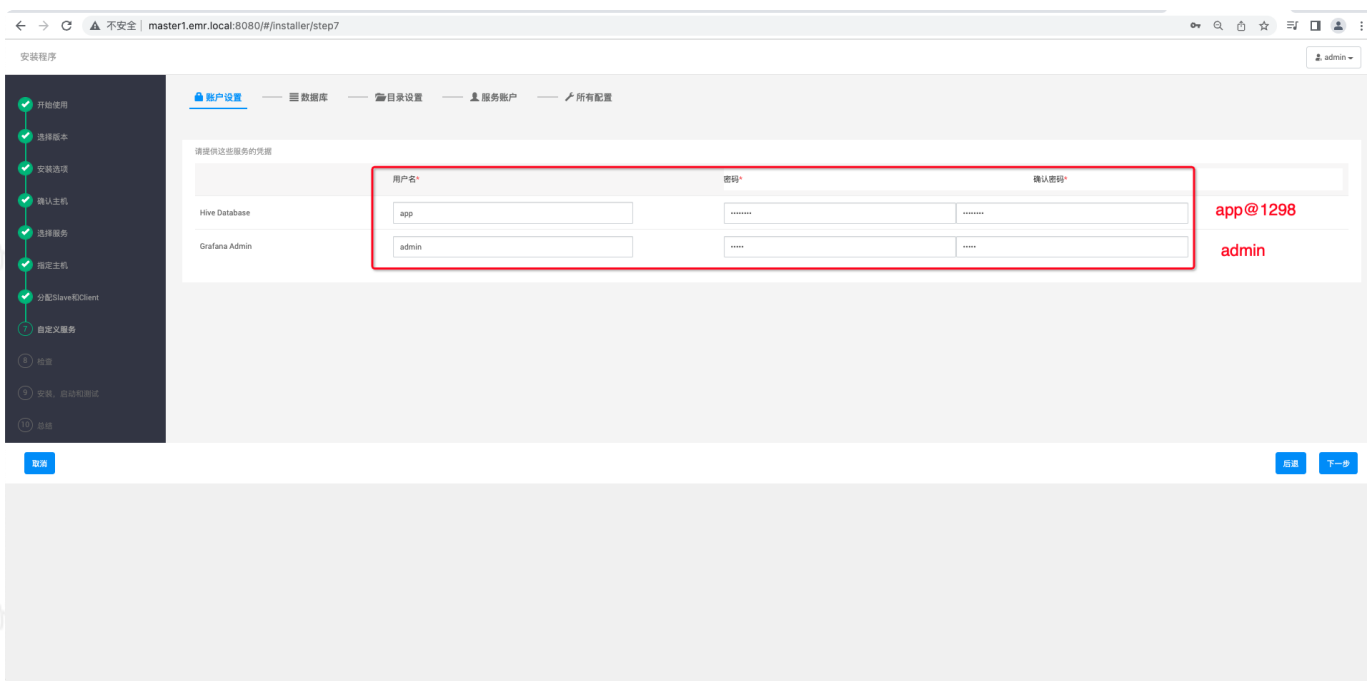
Ranger Usersync

Ranger Admin

node1.emr.local (15.5 GB, 8 cores)

ZooKeeper Server

Metrics Collector



依次修改各个服务的配置，以HDFS为例：

安装选项

确认主机

选择服务

指定主机

分配slave和Client

自定义服务

检查

安装、启动和测试

总结

HDFS

YARN

MAPREDUCE2

TEZ

HIVE

HBASE

ZOOKEEPER

INFRA SOLR

STARRY METRICS

KAFKA

RANGER

SPARK2

SPARK3

DATA DIRS

按照服务器数据盘规划，依次各个服务修改配置项，主要修改数据目录和日志目录

DataNode directories

/data/hadoop/hdfs/data

NameNode directories

/data/hadoop/hdfs/namenode

SecondaryNameNode Checkpoint directories

/data/hadoop/hdfs/namesecondary

NFSGateway dump directory

/tmp/hdfs-nfs

NameNode Backup directory

/tmp/upgrades

JournalNode Edits directory

/data/hadoop/hdfs/journalnode

NameNode Checkpoint Edits directory

\$(dfs.namenode.checkpoint.dir)

LOG DIRS

master1.emr.local:8080/#/installer/step7

安装程序

admin

开始使用

选择版本

安装选项

确认主机

选择服务

指定主机

分配slave和Client

自定义服务

检查

安装、启动和测试

总结

账户设置

数据库

目录设置

服务账户

所有配置

请检查服务账户/组的这些设置

Use Starry to Manage Service Accounts and Groups

Use Starry to Manage Group Memberships

Use Starry to Manage Service Accounts UID's

用户组

用户名

Smoke User

starry-qa

Hadoop Group

hadoop

Metadata User

atlas

Clickhouse User

clickhouse

HBase User

hbase

HDFS User

hdfs

Proxy User Group

users

Hive User

hive

Hue Group

hue

Hue User

hue

Kafka User

kafka

Kudu User

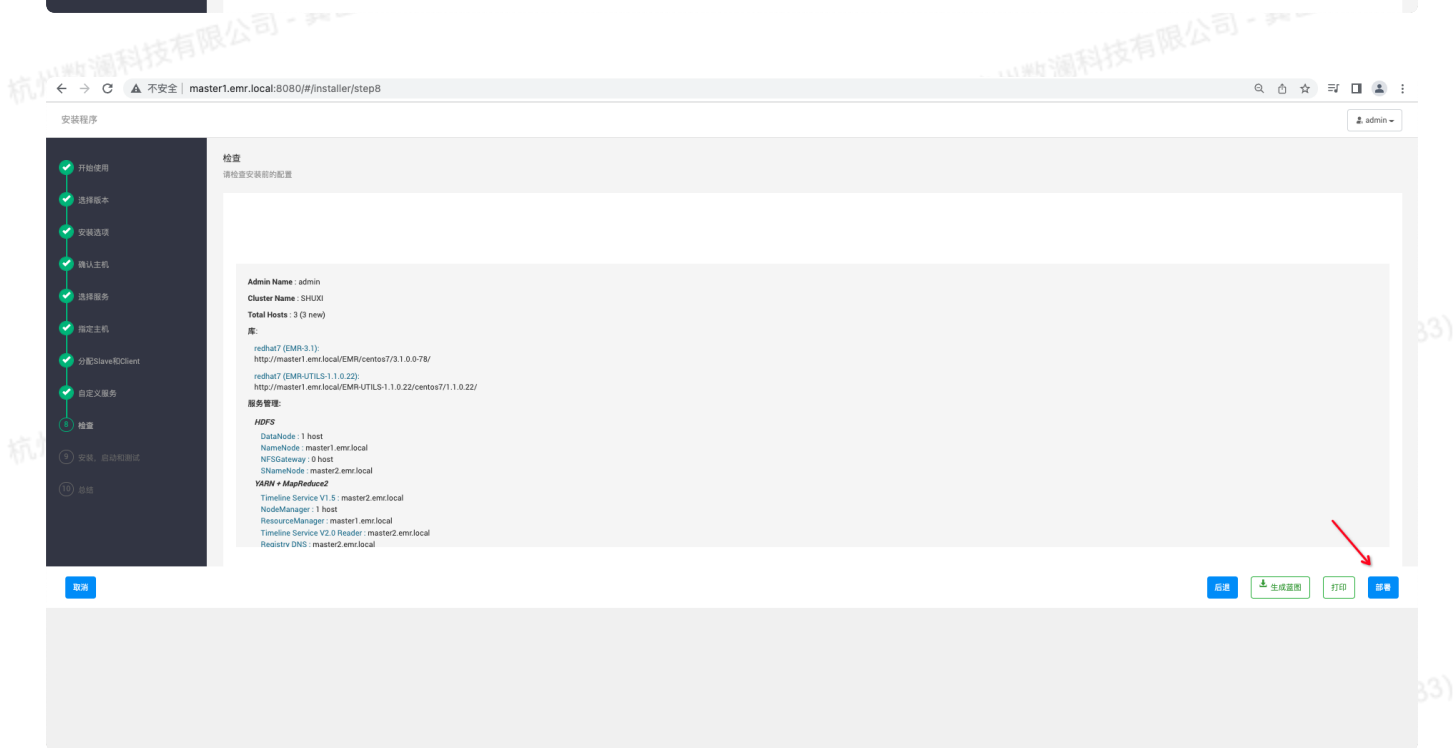
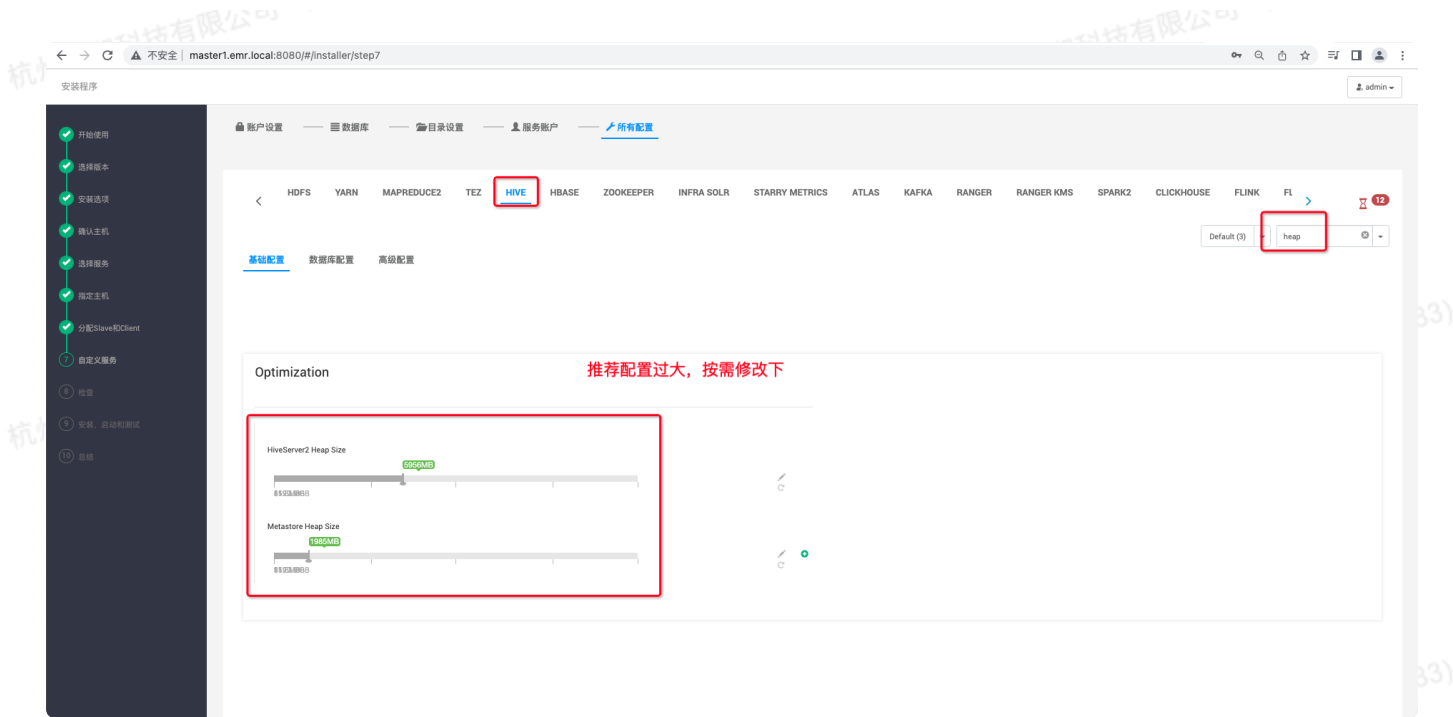
kudu

Kudu Group

hadoop

Mapreduce User

mapred



```

1  注意事项:
2  1. 数据库: hive的元数据库配置增加参数
3  ?characterEncoding=utf8&useSSL=false
4
5  2. 目录配置: 各个服务
6  调整数据目录和日志目录
7
8  3. 所有配置:
9  hive配置中jvm配置不合理, 默认配置非常大, 注意修改
10 hive配置中高级配置搜索hive.security.authorization.enabled, 去掉该配置。(影响创建
    函数)
11 spark配置中搜索spark.ranger.enable, 去掉该配置 (来回点几次, 默认值展示有问题)
12

```

```

1  use hive;
2  alter table COLUMNS_V2 modify column COMMENT varchar(256) character set utf
    8;
3  alter table TABLE_PARAMS modify column PARAM_VALUE varchar(4000) character
    set utf8;
4  alter table PARTITION_PARAMS modify column PARAM_VALUE varchar(4000) chara
    cter set utf8;
5  alter table PARTITION_KEYS modify column PKEY_COMMENT varchar(4000) charac
    ter set utf8;
6  alter table INDEX_PARAMS modify column PARAM_VALUE varchar(4000) charact
    er set utf8;
7  alter table COLUMNS_V2 modify column COLUMN_NAME varchar(767) character set
    utf8;

```

```

1  # EMR安装后HDFS和YARN相关监控指标无法显示问题, 可以按如下操作修复。
2  在HDFS -> Advanced core-site
3  hadoop.http.authentication.simple.anonymous.allowed=true
4  在HDFS -> Custom core-site
5  hadoop.http.authentication.type=simple
6  hadoop.proxyuser.HTTP.groups=*
7  set hadoop.proxyuser.knox.groups=*
8  set hadoop.proxyuser.knox.hosts=*
9  set hadoop.proxyuser.yarn.hosts=*

```