使用 Ambari 部署 HDP 平台

Name ： 曲中岭

Email: zlingqu@126.com

Q Q ： 441869115

日 期 ： 2021 年 08 月

目录

1 介绍

1.1 ambari 介绍

Ambari 是 Apache Software Foundation 中的一个顶级项目，它可以创建、管理、监视 Hadoop 整个生态圈（例如 Hive、HBase、Sqoop、ZooKeeper 等）的集群，使得 Hadoop 以及相关的大数据软件更容易使用。

Ambari 是一个分布式架构的软件，由 Ambari Server 和 Ambari Agent 两部分组成，用户可通过 Ambari Server 通知 Ambari Agent 安装对应的软件，Ambari Agent 会定时地发送各个机器每个软件模块的状态给 Ambari Server，最终这些状态信息会呈现在 Ambari 的 GUI 界面上，方便用户了解到集群的各种状态，并进行相应的维护。

1.2 规划

序号	IP	hostname	OS	数据盘	角色
0	10.12.19.31	devops-10-12-19-31	CentOS7.8		ambari-server,yum-repo
1	10.11.16.1	10-11-16-1.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop
2	10.11.16.2	10-11-16-2.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop
3	10.11.16.3	10-11-16-3.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop
4	10.11.16.4	10-11-16-4.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop
5	10.11.16.5	10-11-16-5.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop
6	10.11.16.6	10-11-16-6.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop
7	10.11.16.7	10-11-16-7.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop
8	10.11.16.8	10-11-16-8.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop
9	10.11.16.9	10-11-16-9.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop
10	10.11.16.10	10-11-16-10.dm-ai.com	CentOS7.8	/data	ambari-agent,hadoop

说明：

- 第 0 台作为主控服务器，部署 ambari-server，搭建私有 yum 源
- 第 1-10 台作为 slave，部署 ambari-agent，也是 hadoop 运行的地方。
- mysql 数据库位于 192.168.3.199，使用 docker 启动的的，后面会说。实际生产中，一般归 DBA 管理，这里不再单独列出。

1.3 ansible 脚本

本文中的内容有一部分可以使用 ansible 方便的管理，地址：
https://github.com/zlingqu/ansible/tree/master/hdp

2 Ambari-server 初始化

本章节的操作针对 Ambari-server 节点操作！

2.1 准备相关包

在未订阅的情况下，Cloudera 官方已经不提供相关包的下载了（），，商业化进程进一步加快。官方公告：https://cn.cloudera.com/downloads/paywall-expansion.html 。

我自己在百度网盘保存了一份：

链接: https://pan.baidu.com/s/1bRtnpyYMIK8N1zXj61Y86Q

提取码: qjs3

下载好后，放到/data/hdp 目录，会得到如下四个 tgz 压缩文件

```
# ls -lh *gz
-rw-r--r-- 1 root root 2.0G Jul 27 23:43 ambari-2.7.5.0-centos7.tar.gz
-rw-r--r-- 1 root root 9.2G Jul 29 00:55 HDP-3.1.5.0-centos7-rpm.tar.gz
-rw-r--r-- 1 root root 159K Jul 29 00:55 HDP-GPL-3.1.5.0-centos7-gpl.tar.gz
-rw-r--r-- 1 root root  87M Jul 29 01:10 HDP-UTILS-1.1.0.22-centos7.tar.gz
# pwd
/data/hdp
```

分别解压四个文件，解压后的目录结构如下类似：

```
# ls -lh
total 12G
drwxr-xr-x 3 root root    21 Aug  2 18:03 ambari
-rw-r--r-- 1 root root  2.0G Jul 27 23:43 ambari-2.7.5.0-centos7.tar.gz
drwxr-xr-x 3 1001 users   21 Dec 18  2019 HDP
-rw-r--r-- 1 root root  9.2G Jul 29 00:55 HDP-3.1.5.0-centos7-rpm.tar.gz
drwxr-xr-x 3 1001 users   21 Dec 18  2019 HDP-GPL
-rw-r--r-- 1 root root  159K Jul 29 00:55 HDP-GPL-3.1.5.0-centos7-gpl.tar.gz
drwxr-xr-x 3 1001 users   21 Aug 13  2018 HDP-UTILS
-rw-r--r-- 1 root root   87M Jul 29 01:10 HDP-UTILS-1.1.0.22-centos7.tar.gz

# tree -L 4
.
├── ambari
│   └── centos7
│       └── 2.7.5.0-72
│           ├── ambari
│           ├── ambari.repo
│           ├── artifacts.txt
│           ├── build.id
│           ├── build_metadata.txt
│           ├── hotfix_index.html
│           ├── index.html
```

```
│       ├── private_index.html
│       ├── public_index.html
│       ├── repodata
│       ├── RPM-GPG-KEY
│       ├── smartsense
│       └── tars
├── ambari-2.7.5.0-centos7.tar.gz
├── HDP
│   └── centos7
│       └── 3.1.5.0-152
│       ├── accumulo
│       ├── artifacts.txt
│       ├── atlas
│       ├── bigtop-jsvc
│       ├── bigtop-tomcat
│       ├── build.id
│       ├── build_metadata.txt
│       ├── datafu
│       ├── druid
│       ├── hadoop
│       ├── hbase
│       ├── HDP-3.1.5.0-152-MAINT.xml
│       ├── HDP-3.1.5.0-152.xml
│       ├── hdp.repo
│       ├── hdp-select
│       ├── hive
│       ├── hive_warehouse_connector
│       ├── hotfix_index.html
│       ├── index.html
│       ├── kafka
│       ├── knox
│       ├── livy
│       ├── oozie
│       ├── phoenix
│       ├── pig
│       ├── private_index.html
│       ├── public_index.html
│       ├── ranger
│       ├── repodata
│       ├── RPM-GPG-KEY
│       ├── shc
│       ├── spark2
│       ├── spark_atlas_connector
│       ├── spark_schema_registry
```

```
│           ├──── sqoop
│           ├──── ssl_hdp.repo
│           ├──── storm
│           ├──── superset
│           ├──── tez
│           ├──── vrpms
│           ├──── zeppelin
│           └──── zookeeper
├──── HDP-3.1.5.0-centos7-rpm.tar.gz
├──── HDP-GPL
│    └──── centos7
│        └──── 3.1.5.0-152
│           ├──── hadooplzo
│           ├──── hdp.gpl.repo
│           ├──── repodata
│           ├──── RPM-GPG-KEY
│           ├──── ssl_hdp.gpl.repo
│           └──── vrpms
├──── HDP-GPL-3.1.5.0-centos7-gpl.tar.gz
├──── HDP-UTILS
│    └──── centos7
│        └──── 1.1.0.22
│           ├──── hdp-utils.repo
│           ├──── openblas
│           ├──── repodata
│           ├──── RPM-GPG-KEY
│           └──── snappy
└──── HDP-UTILS-1.1.0.22-centos7.tar.gz
```

2.2 安装 nginx 服务

```
yum install nginx -y
```

配置 nginx

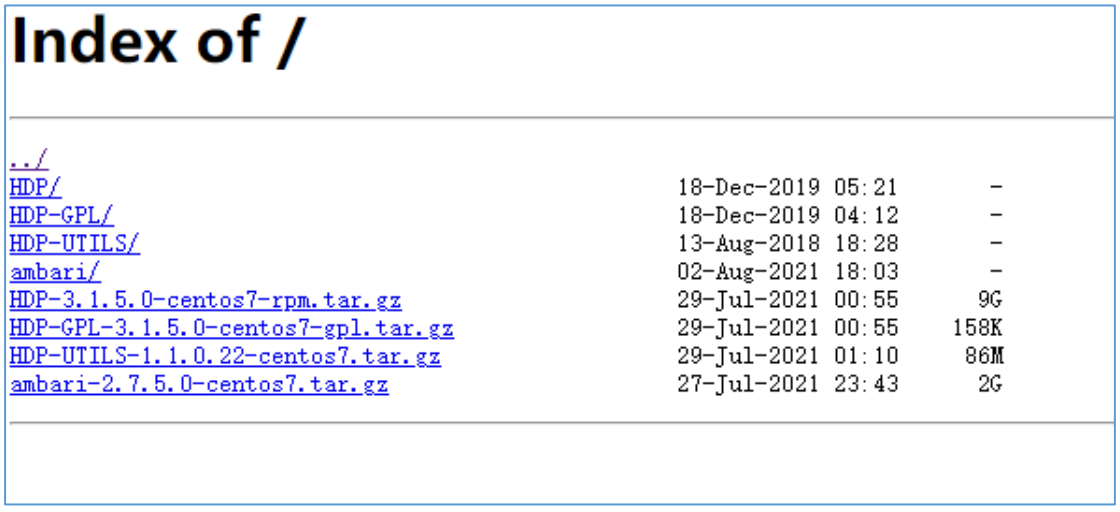
```
# cat /etc/nginx/conf.d/yum.conf
server {
    listen      80;
    server_name  localhost;
    large_client_header_buffers 4 128k;
    client_max_body_size 300m;
    client_body_buffer_size 128k;
    proxy_connect_timeout 600;
    proxy_read_timeout 600;
    proxy_send_timeout 600;
    proxy_buffer_size 64k;
```

```
    proxy_buffers   4 32k;
    proxy_busy_buffers_size 64k;
    proxy_temp_file_write_size 64k;

    #charset koi8-r;

    #access_log  logs/host.access.log  main;

    location / {
       root   /data/hdp;
       index  index.html index.htm;
       autoindex on;
       autoindex_exact_size off;
       autoindex_localtime on;
       proxy_set_header X-Real-IP $remote_addr;
       proxy_set_header REMOTE-HOST $remote_addr;
       proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

启动 nginx

```
service nginx start
```

浏览器访问，看到以下类似内容表示正常



这一步是制作了一个私有的 yum 源，提供相关服务。

2.3 添加 yum 私有 repo

```
# cat /etc/yum.repos.d/ambari.repo
[ambari]
baseurl = http://10.12.19.31/ambari/centos7/2.7.5.0-72/
enabled = 1
```

```
gpgcheck = 0
name = ambari server
```

2.4 设置免密登录

Ambari Server 访问 Ambari Agent 需要配置 ssh 无密互访，Ambari Server 通过 SSH 公钥授权远程登录其他主机。

这里不再演示，可用 ansible 实现。

3 Ambari-agent 初始化

本章节的操作，针对 10 台 Ambari-agent 进行

3.1 修改主机名

根据规划，修改 11 台服务器的主机名，这里不再演示，可用 ansible 实现。

3.2 hosts 配置

ambari-server 的 hosts 添加如下配置。

```
# Ansible inventory hosts BEGIN
10.11.16.1 10-11-16-1.dm-ai.com
10.11.16.2 10-11-16-2.dm-ai.com
10.11.16.3 10-11-16-3.dm-ai.com
10.11.16.4 10-11-16-4.dm-ai.com
10.11.16.5 10-11-16-5.dm-ai.com
10.11.16.6 10-11-16-6.dm-ai.com
10.11.16.7 10-11-16-7.dm-ai.com
10.11.16.8 10-11-16-8.dm-ai.com
10.11.16.9 10-11-16-9.dm-ai.com
10.11.16.10 10-11-16-10.dm-ai.com
# Ansible inventory hosts END
10.12.19.31 devops-10-12-19-31
```

可用 ansible 实现

3.3 关闭防火墙

进行安装任务，集群所有主机都必须关闭防火墙

```
systemctl disable firewalld
systemctl stop firewalld
```

如果用了 iptables,也需要清空相关规则

```
iptables -F && iptables -t nat -F && iptables -t mangle -F && iptables -X
```

可用 ansible 实现

3.4 关闭 SELinux

关闭 SeLinux 功能，实现细节这里不再说。

可用 ansible 实现。

3.5 配置 limits 参数

编辑/etc/security/limits.conf 文件：

保证文件中有如下内容：

```
* hard    nproc   65535
* soft    nproc   65535
* hard    nofile  65535
* soft    nofile  65535
```

可用 ansible 实现。

3.6 禁用交换分区

集群内所有主机禁用交换分区。

```
sysctl vm.swappiness=0
echo vm.swappiness=0 >> /etc/sysctl.conf
```

可用 ansible 实现。

3.7 禁用大内存页功能

```

echo never > /sys/kernel/mm/transparent_hugepage/defrag
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

可用 ansible 实现。

3.8 时间同步

hadoop 集群中各节点时间要同步，方法有很多，但都大同小异，这里不再细说。

可用 ansible 实现。

3.9 安装 JDK

我选用的 oracle 的 jdk，JAVA_HOME=/usr/local/jdk，这里不再细说。

可使用 ansible 实现。

3.10添加 ambari 源

```
# cat /etc/yum.repos.d/ambari.repo
[ambari]
baseurl = http://10.12.19.31/ambari/centos7/2.7.5.0-72/
enabled = 1
gpgcheck = 0
name = ambari server
```

可以使用 ansible 实现。

3.11 安装 ambari-agent

```
yum install ambari-agent -y
service ambari-agent start
```

可以使用 ansible 实现。

4 Ambari 部署

4.1 mysql 准备

ambari 可使用多种数据库存储数据，我这里选择最常用的 mysql。mysql 的搭建这里不再细说，如果只是测试可以使用 docker 部署，比较方便，这里提供 docker-compose 文件。

```
# cat docker-compose.yml
version: "3"
services:
  mysql:
    image: mysql:5.7.27
    ports:
      - "3306:3306"
    environment:
      MYSQL_ROOT_PASSWORD: 123456
      MYSQL_DATABASE: test
      TZ: Asia/Shanghai
    volumes:
      - /data/mysql-3306/data:/var/lib/mysql
```

4.2 部署 Ambari-server

在规划中的第 0 台服务器操作！

4.2.1 安装服务

```
yum install ambari-server -y
```

4.2.2 数据库驱动配置

保证环境有 mysql 的 java 驱动，比如/usr/share/java/mysql-connector-java-5.1.49.jar

```
# ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java-
5.1.49.jar
Using python  /usr/bin/python
Setup ambari-server
Copying /usr/share/java/mysql-connector-java-5.1.49.jar to /var/lib/ambari-
server/resources/mysql-connector-java-5.1.49.jar
Creating symlink /var/lib/ambari-server/resources/mysql-connector-java-5.1.49.jar to
/var/lib/ambari-server/resources/mysql-connector-java.jar
```

If you are updating existing jdbc driver jar for mysql with mysql-connector-java-5.1.49.jar. Please remove the old driver jar, from all hosts. Restarting services that need the driver, will automatically copy the new jar to the hosts.
JDBC driver was successfully initialized.
Ambari Server 'setup' completed successfully.

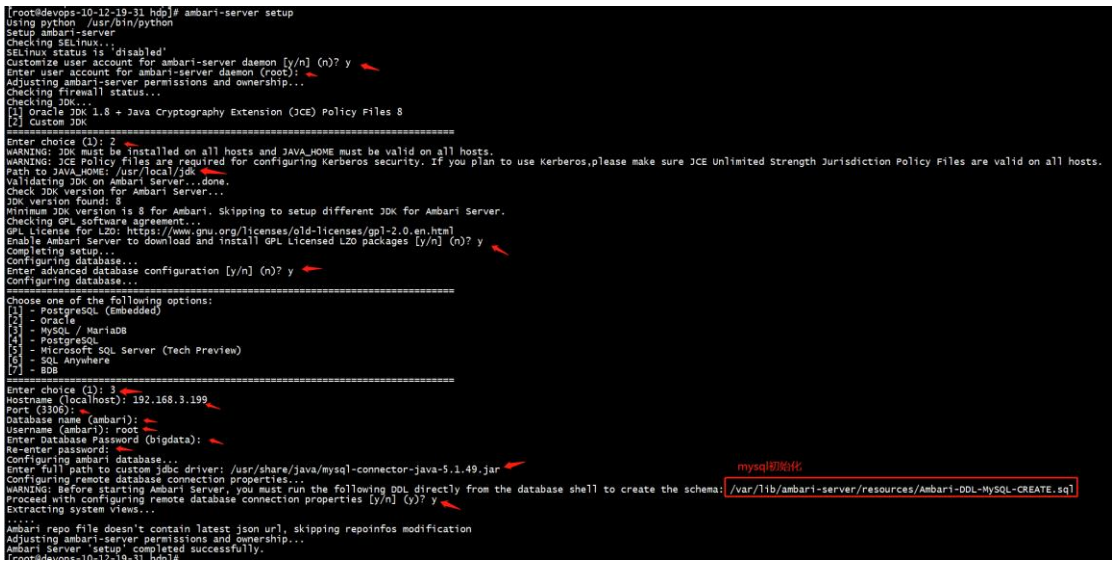
设置后，配置文件会产生变化，最重要的变化是添加了

```
custom.mysql.jdbc.name=mysql-connector-java-5.1.49.jar
```

4.2.3 配置初始化

```
ambari-server setup
```

如下图所示，图中箭头标识处需要进行选择或者输入相关内容



这一步也会自动修改配置文件，比如添加了如下内容

```
server.jdbc.user.name=root
server.jdbc.url=jdbc:mysql://192.168.3.199:3306/ambari
server.jdbc.driver.path=/usr/share/java/mysql-connector-java-5.1.49.jar
java.home=/usr/local/jdk
java.releases=jdk1.8
custom.mysql.jdbc.name=mysql-connector-java-5.1.49.jar
stack.java.home=/usr/local/jdk
```

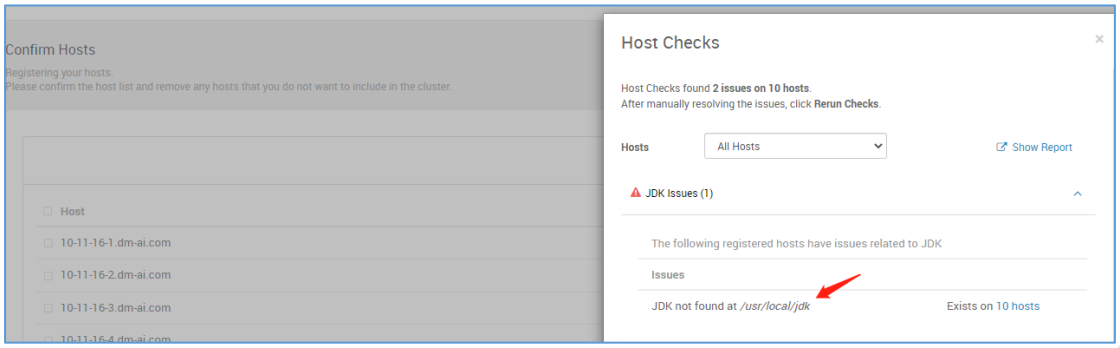
4.2.4 数据库初始化

如前图 mysql 初始化脚本在：/var/lib/ambari-server/resources/Ambari-DDL-MySQL-CREATE.sql 可以使用 navicat 或者命令行等方式，对 ambari 库进行初始化。
初始化完成后，会有如下表，共 111 个：

打开表 设计表 新建表 删除表 导入向导 导出向导

名	自动递增值	修改日期	数据长度	表类型	行	注释
adminpermission		2021-08-03 17:16:54	16 KB	InnoDB	7	
adminprincipal		2021-08-03 17:16:54	16 KB	InnoDB	8	
adminprincipaltype		2021-08-03 17:16:54	16 KB	InnoDB	3	
adminprivilege		2021-08-03 17:16:54	16 KB	InnoDB	1	
adminresource		2021-08-03 17:16:54	16 KB	InnoDB	0	
adminresourcetype		2021-08-03 17:16:54	16 KB	InnoDB	3	
alert_current			16 KB	InnoDB	0	
alert_definition			16 KB	InnoDB	0	
alert_group			16 KB	InnoDB	0	
alert_group_target			16 KB	InnoDB	0	
alert_grouping			16 KB	InnoDB	0	
alert_history			16 KB	InnoDB	0	
alert_notice			16 KB	InnoDB	0	
alert_target			16 KB	InnoDB	0	
alert_target_states			16 KB	InnoDB	0	
ambari_configuration			16 KB	InnoDB	0	
ambari_operation_history			16 KB	InnoDB	0	
ambari_sequences		2021-08-03 17:16:54	16 KB	InnoDB	50	
artifact			16 KB	InnoDB	0	
blueprint			16 KB	InnoDB	0	
blueprint_configuration			16 KB	InnoDB	0	
blueprint_setting			16 KB	InnoDB	0	
clusterconfig			16 KB	InnoDB	0	
clusterhostmapping			16 KB	InnoDB	0	
clusters			16 KB	InnoDB	0	
clusterservices			16 KB	InnoDB	0	
clusterstate			16 KB	InnoDB	0	
confgroupclusterconfigmapping			16 KB	InnoDB	0	
configgroup			16 KB	InnoDB	0	
configgrouphostmapping			16 KB	InnoDB	0	
execution_command			16 KB	InnoDB	0	
extension			16 KB	InnoDB	0	
extensionlink			16 KB	InnoDB	0	
groups			16 KB	InnoDB	0	
host_role_command			16 KB	InnoDB	0	
host_version			16 KB	InnoDB	0	
hostcomponentdesiredstate			16 KB	InnoDB	0	
hostcomponentstate			16 KB	InnoDB	0	
hostconfigmapping			16 KB	InnoDB	0	

注意：图中自定义了 jdk 路径 /usr/local/jdk, 经测试发现，对于 ambari-agent 也会从这里路径查找 jdk，否则在主机检测时会报错如下



4.2.5 启动服务

```
# service ambari-server start
```

```
[root@devops-10-12-19-31 hdp]# service ambari-server start
Using python  /usr/bin/python
Starting ambari-server
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start.....................................................
Server started listening on 8080

DB configs consistency check: no errors and warnings were found.
Ambari Server 'start' completed successfully.
```

默认监听在 8080 端口。

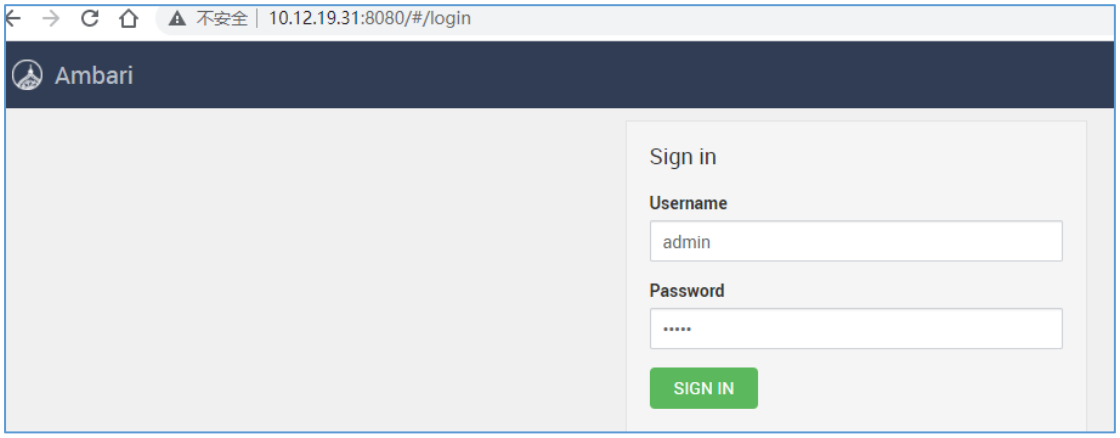
已自动设置了开机自启。

```
[root@devops-10-12-19-31 hdp]# chkconfig

Note: This output shows SysV services only and does not include native
      systemd services. SysV configuration data might be overridden by native
      systemd configuration.

      If you want to list systemd services use 'systemctl list-unit-files'.
      To see services enabled on particular target use
      'systemctl list-dependencies [target]'.

ambari-server   0:off   1:off   2:on    3:on    4:on    5:on    6:off
netconsole      0:off   1:off   2:off   3:off   4:off   5:off   6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@devops-10-12-19-31 hdp]#
[root@devops-10-12-19-31 hdp]# service ambari-server stop
```

5 部署 Hadoop 集群

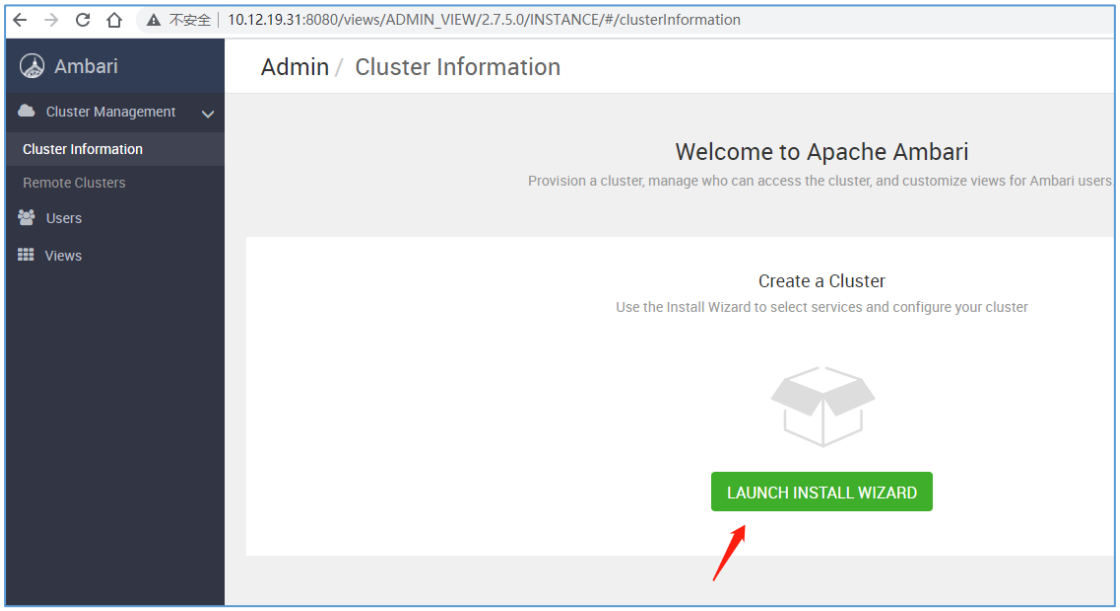
5.1 登录 ambari 界面

启动 ambari-server 后，通过客户端浏览器登录 Ambari 的界面。
浏览器网址：http://10.12.19.31:8080,默认监听 8080
默认登录用户名/密码为：admin/admin。

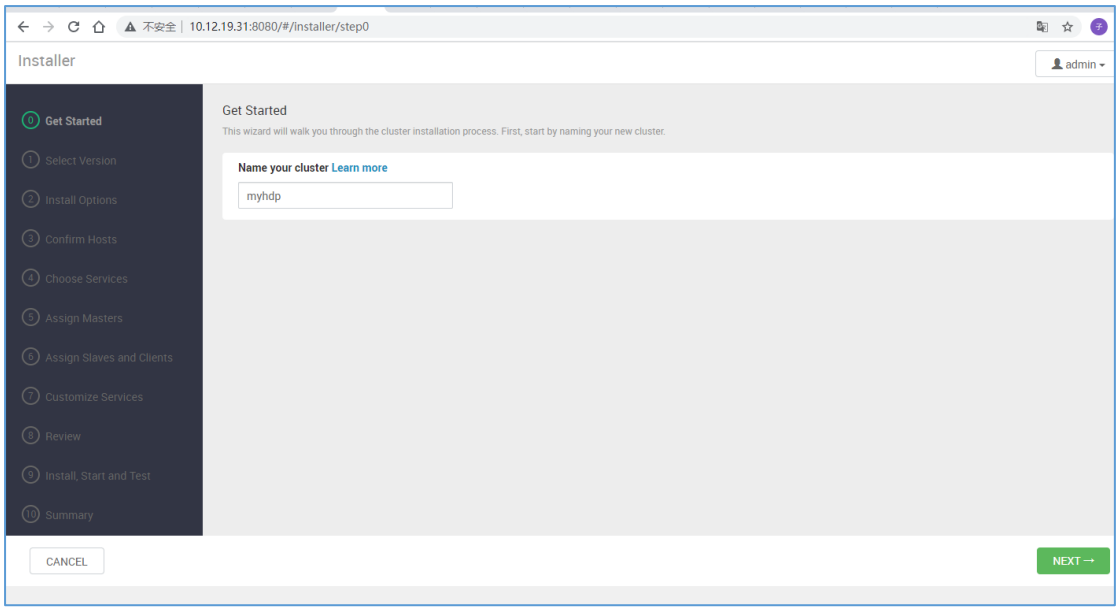


5.2 启动 Ambari 部署向导

点击 **Launch Install Wizard** 搭建集群

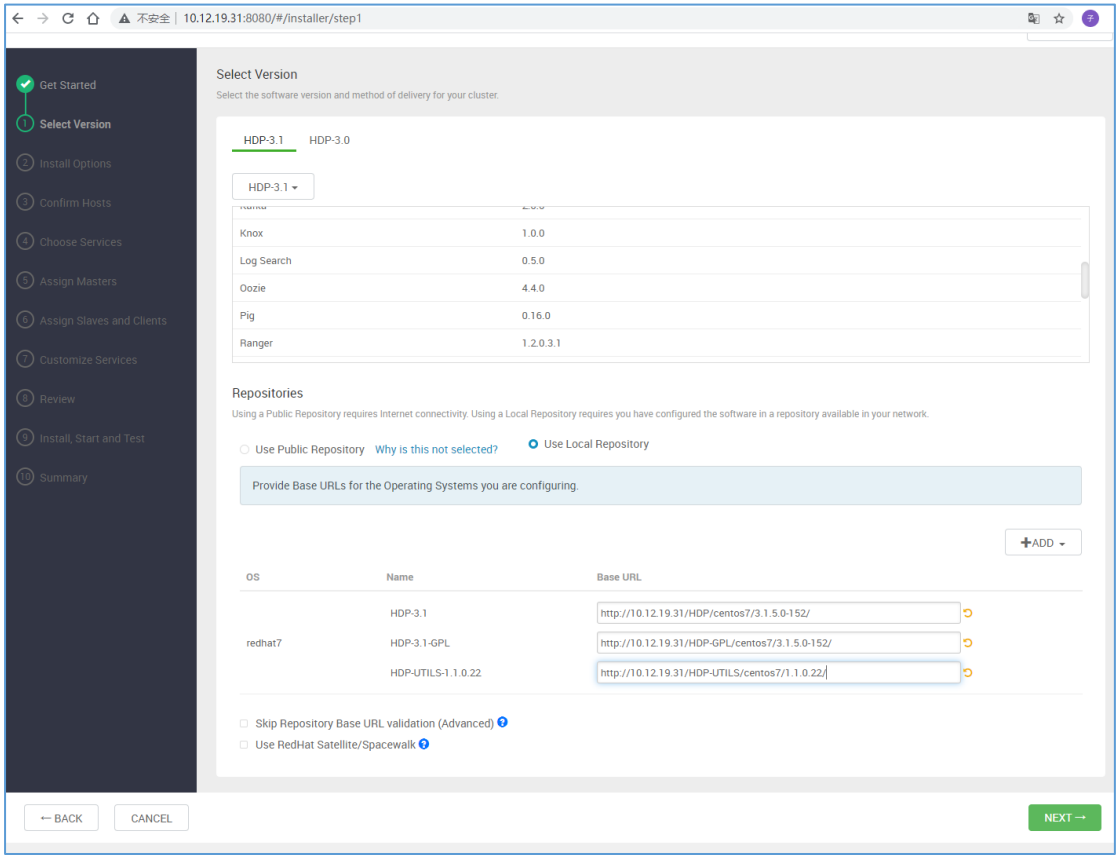


输入集群的名字，不能使用空格和其他特殊字符，我这里使用 myhdp



5.3 选择版本

这一步是配置各种 os 的源，我这里全都是 CentOS，所以只保留 redhat7 即可，其他的进行移除，当然有需要后续也可以添加。



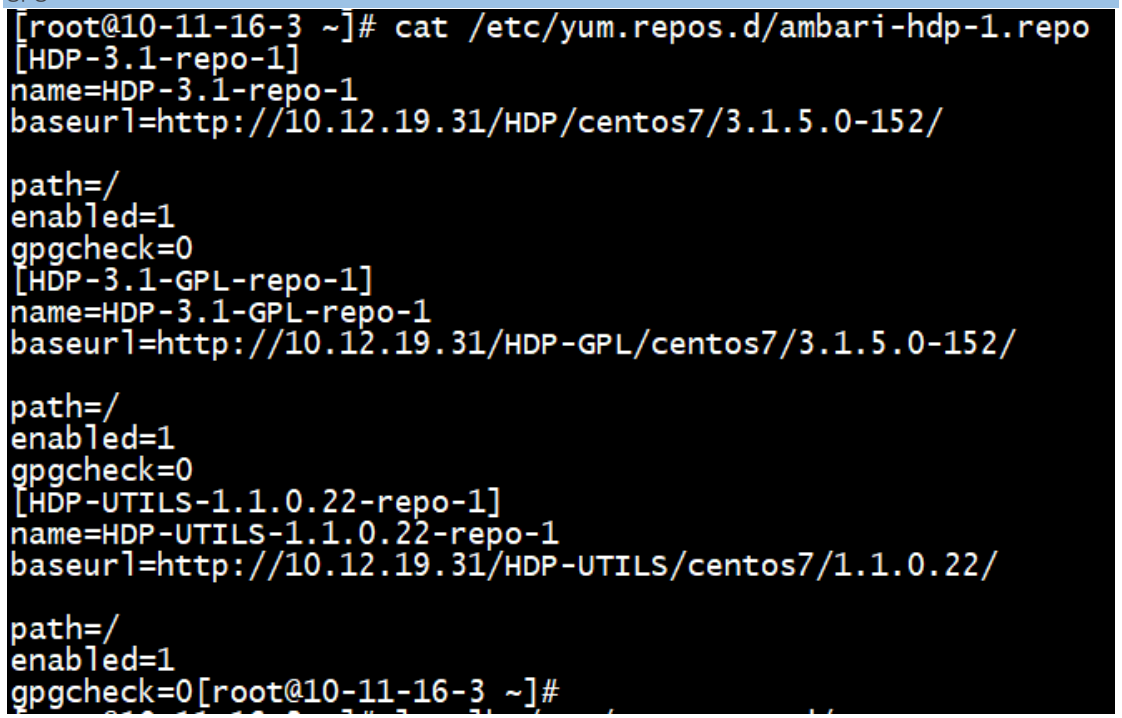
输入的三个地址，是前面创建好的。

```
http://10.12.19.31/HDP/centos7/3.1.5.0-152/
http://10.12.19.31/HDP-GPL/centos7/3.1.5.0-152/
```

http://10.12.19.31/HDP-UTILS/centos7/1.1.0.22/

这一步配置，完成后会在各 agent 节点生成 yum 的 repo 文件，如下

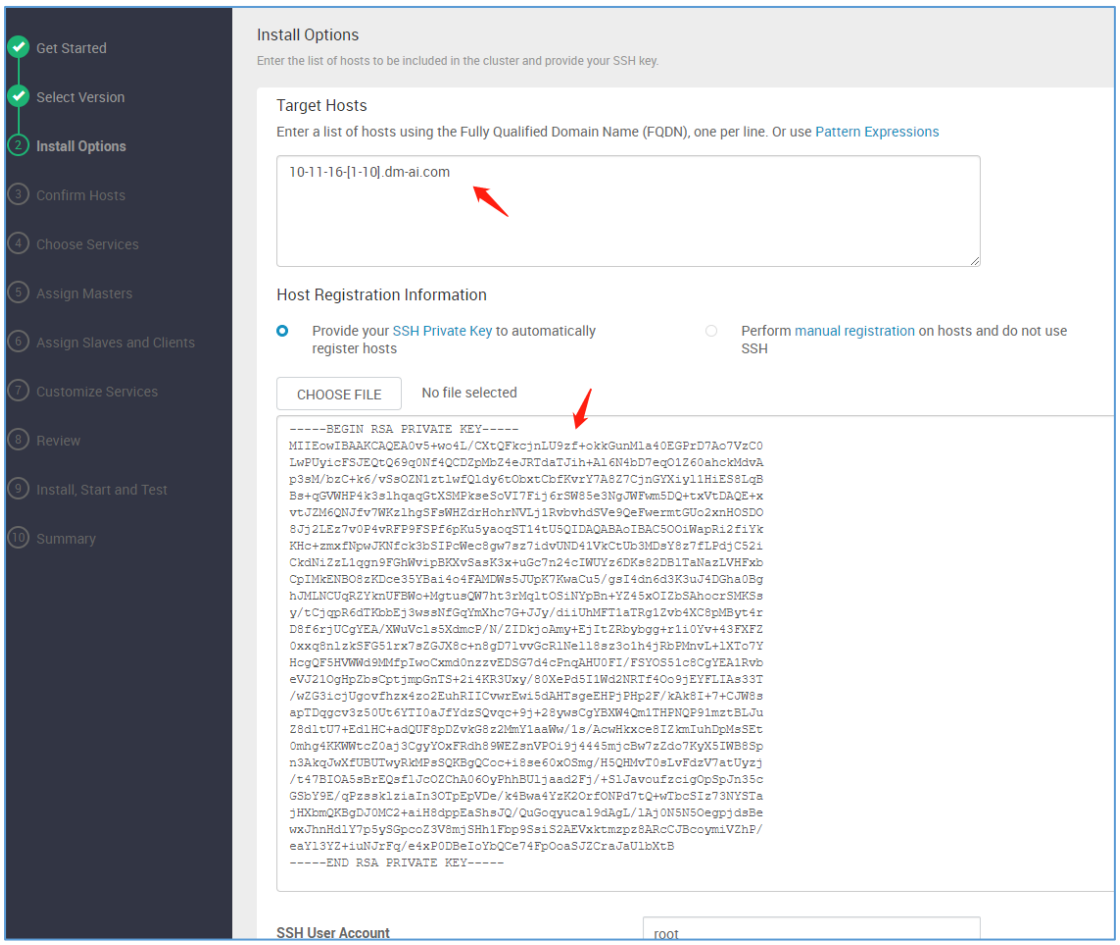
```
# cat /etc/yum.repos.d/ambari-hdp-1.repo
[HDP-3.1-repo-1]
name=HDP-3.1-repo-1
baseurl=http://10.12.19.31/HDP/centos7/3.1.5.0-152/

path=/
enabled=1
gpgcheck=0
[HDP-3.1-GPL-repo-1]
name=HDP-3.1-GPL-repo-1
baseurl=http://10.12.19.31/HDP-GPL/centos7/3.1.5.0-152/

path=/
enabled=1
gpgcheck=0
[HDP-UTILS-1.1.0.22-repo-1]
name=HDP-UTILS-1.1.0.22-repo-1
baseurl=http://10.12.19.31/HDP-UTILS/centos7/1.1.0.22/

path=/
enabled=1
gpgcheck=0
```



5.4 安装选项

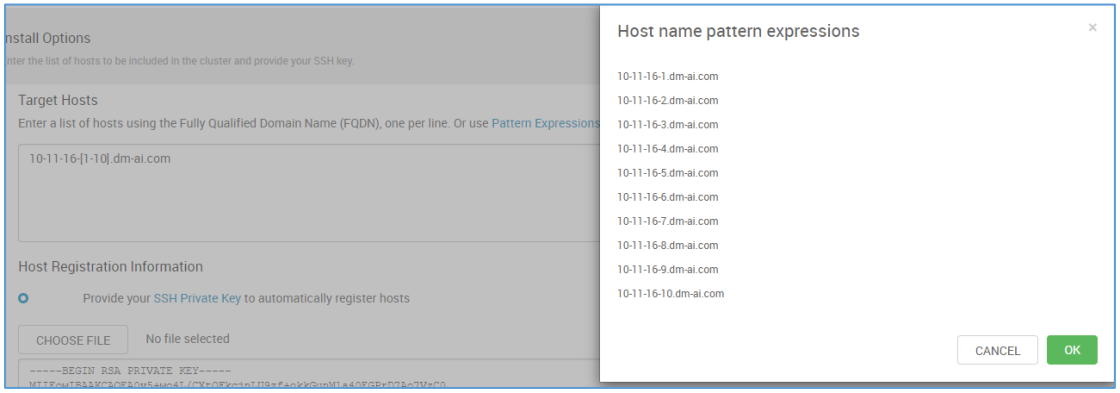
选择管理的主机、ssh 私钥

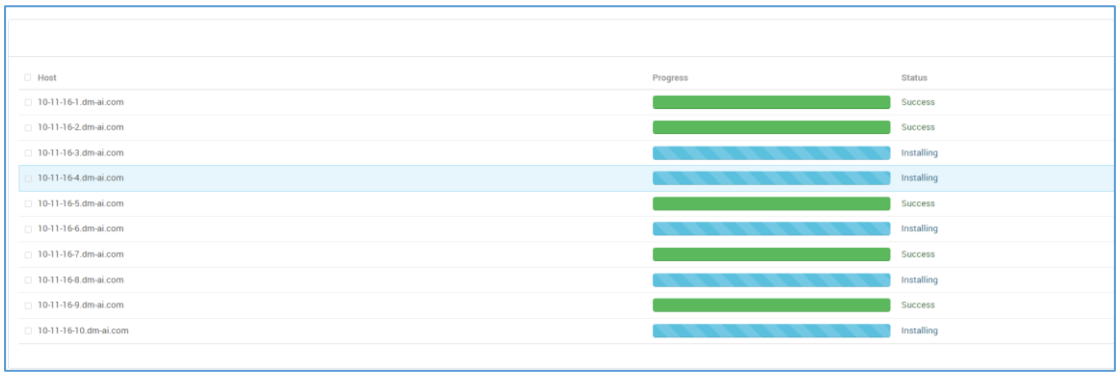
10-11-16-[1-10].dm-ai.com



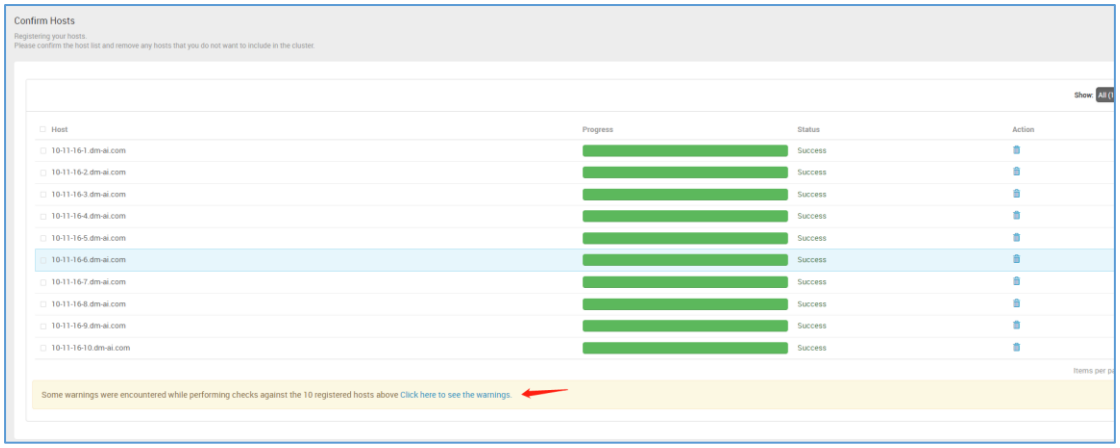
5.5 确认主机

主机名的正则表达式会自动展开，10-11-16-[1-10].dm-ai.com

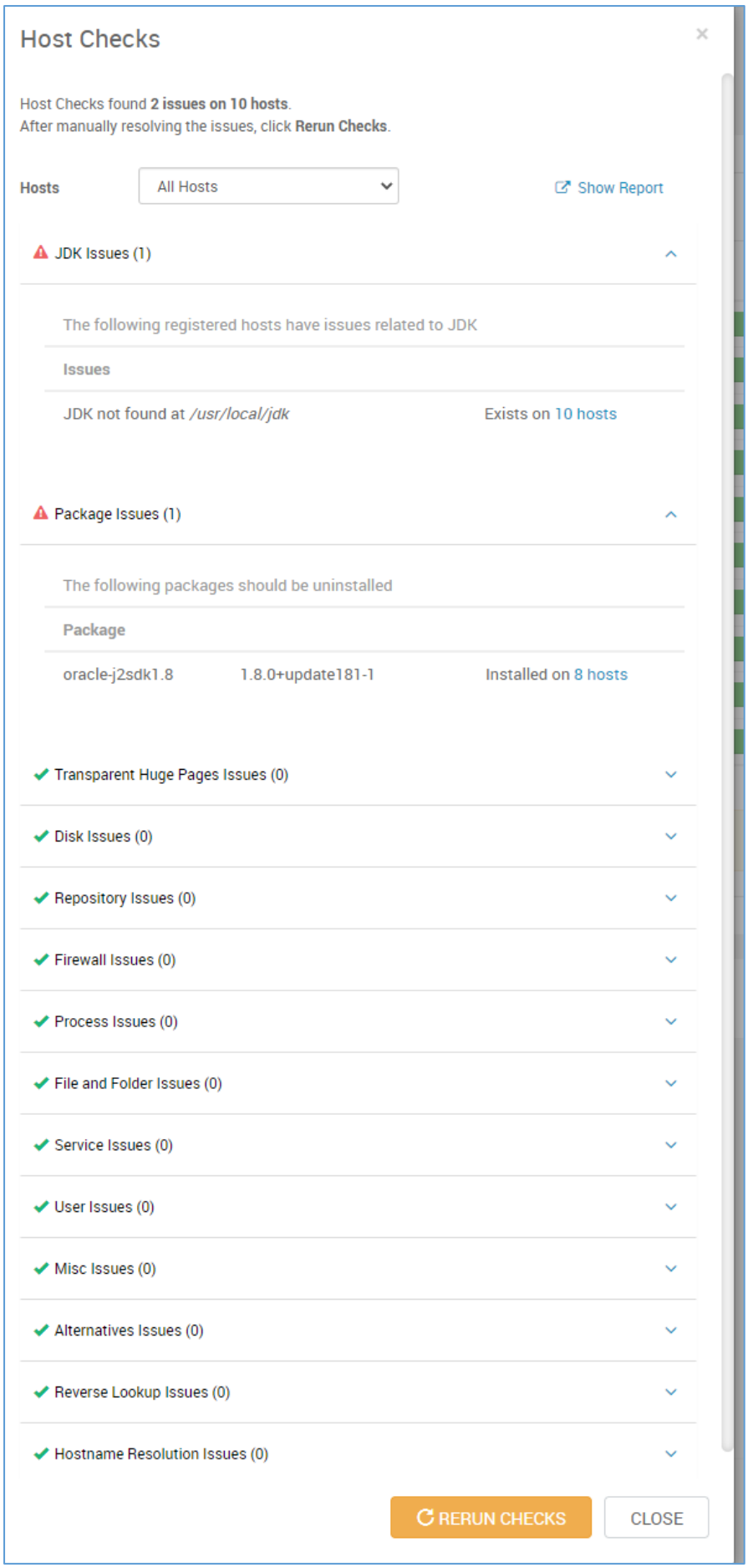


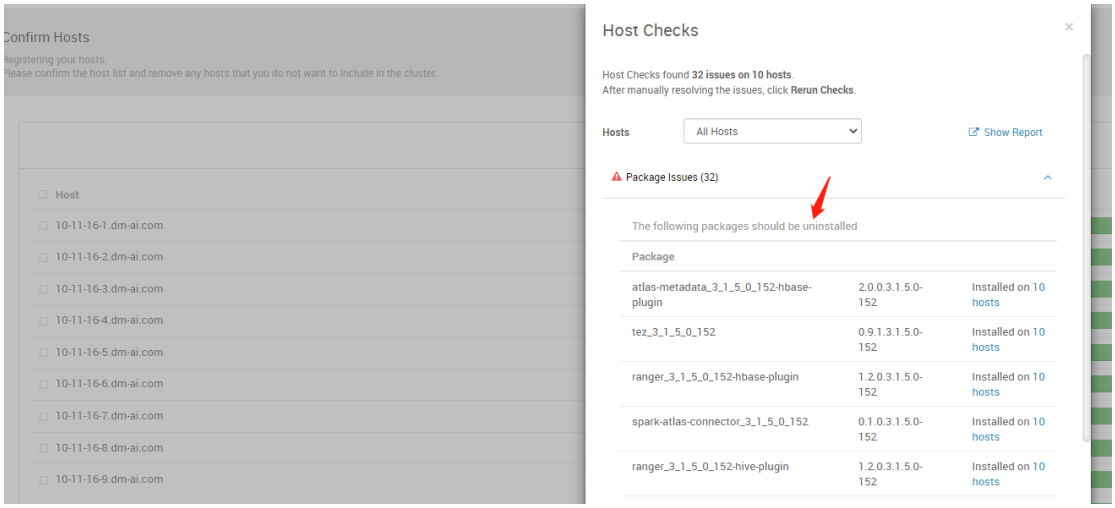


环境检测



检测后如果出现一些警告，比如下图，则需要处理下


Host Checks
Host Checks found 2 issues on 10 hosts.
After manually resolving the issues, click Rerun Checks.
Hosts
All Hosts
Show Report
JDK Issues (1)
The following registered hosts have issues related to JDK
Issues
JDK not found at /usr/local/jdk
Exists on 10 hosts
Package Issues (1)
The following packages should be uninstalled
Package
oracle-j2sdk1.8
1.8.0+update181-1
Installed on 8 hosts
Transparent Huge Pages Issues (0)
Disk Issues (0)
Repository Issues (0)
Firewall Issues (0)
Process Issues (0)
File and Folder Issues (0)
Service Issues (0)
User Issues (0)
Misc Issues (0)
Alternatives Issues (0)
Reverse Lookup Issues (0)
Hostname Resolution Issues (0)
RERUN CHECKS
CLOSE

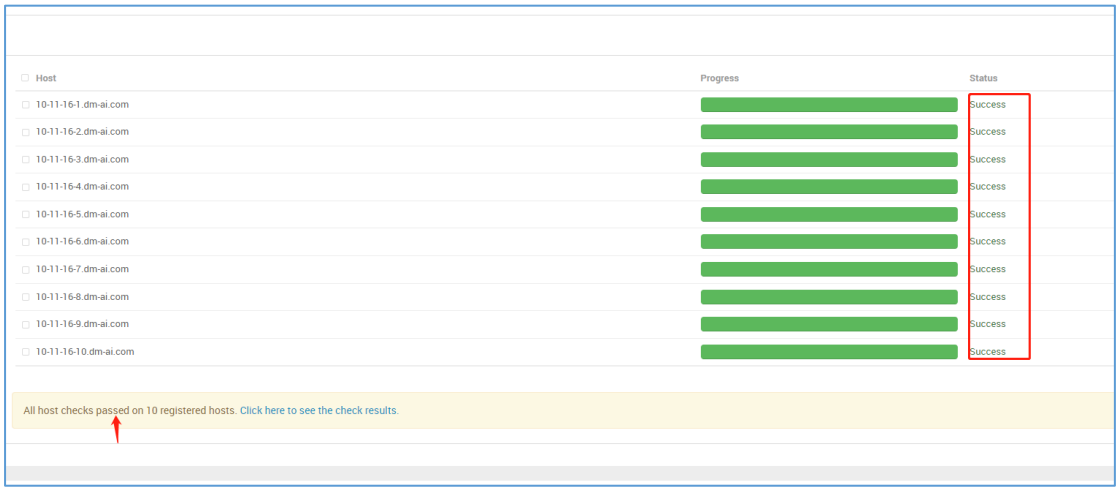


出现图中类似警告，是应为之前的 hadoop 环境没有清理干净，使用 yum 卸载即可。

比如：

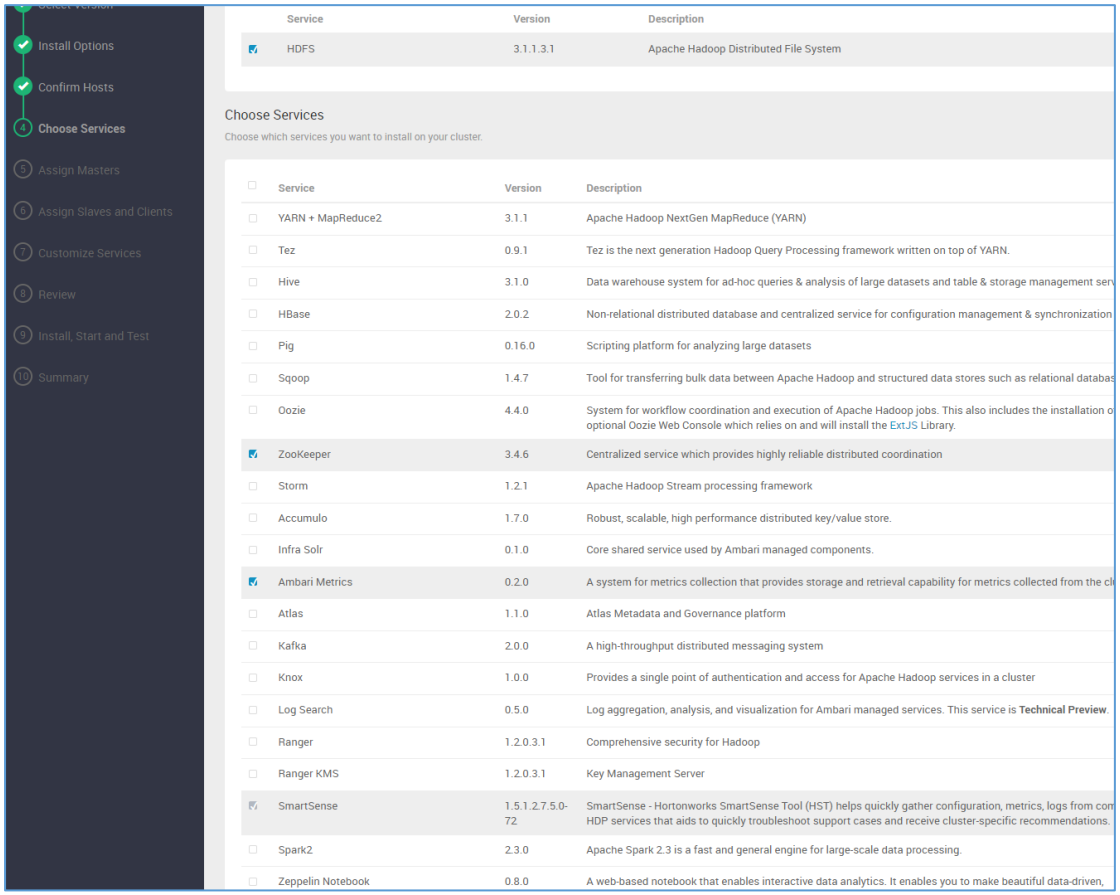
```
ansible ambari-agent -m shell -a "yum remove atlas-metadata_3_1_5_0_152-hbase-plugin tez_3_1_5_0_152 \
ranger_3_1_5_0_152-hbase-plugin spark-atlas-connector_3_1_5_0_152 \
ranger_3_1_5_0_152-hive-plugin livy2_3_1_5_0_152 hadoop_3_1_5_0_152-hdfs \
hive_warehouse_connector_3_1_5_0_152 spark2_3_1_5_0_152 hadoop_3_1_5_0_152-mapreduce \
atlas-metadata_3_1_5_0_152-hive-plugin hive_3_1_5_0_152 hadoop_3_1_5_0_152 hadoop_3_1_5_0_152-yarn \
hive_3_1_5_0_152-jdbc ambari-metrics-grafana spark2_3_1_5_0_152-python spark2_3_1_5_0_152-yarn-shuffle \
hive_3_1_5_0_152-hcatalog hadoop_3_1_5_0_152-client bigtop-jsvc zookeeper_3_1_5_0_152 \
ranger_3_1_5_0_152-hdfs-plugin ambari-metrics-hadoop-sink ambari-metrics-monitor smartsense-hst hbase_3_1_5_0_152 \
ranger_3_1_5_0_152-yarn-plugin hadoop_3_1_5_0_152-libhdfs hdp-select zookeeper_3_1_5_0_152-server \
ambari-metrics-collector -y"
```

全部检测通过，出现 passed 字样，纳入管理后后会出现 success 字样，如下图

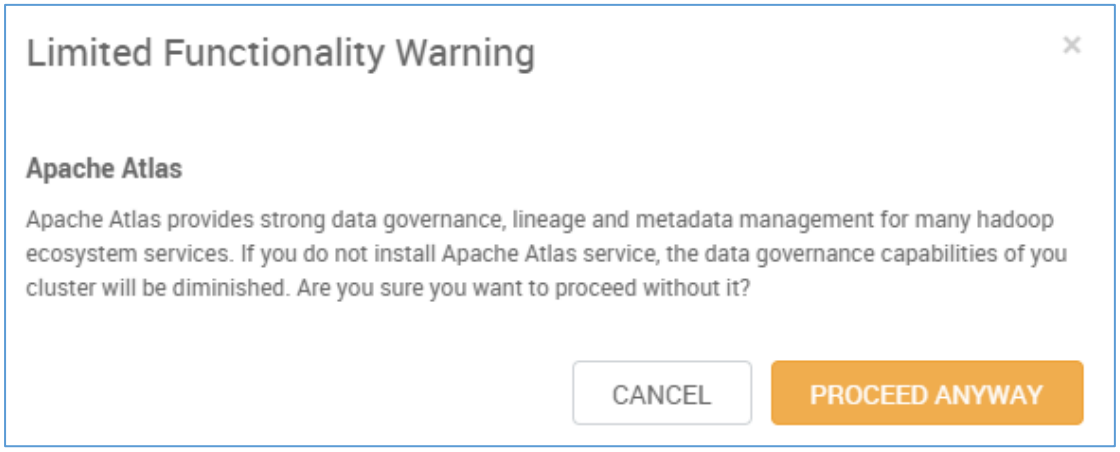


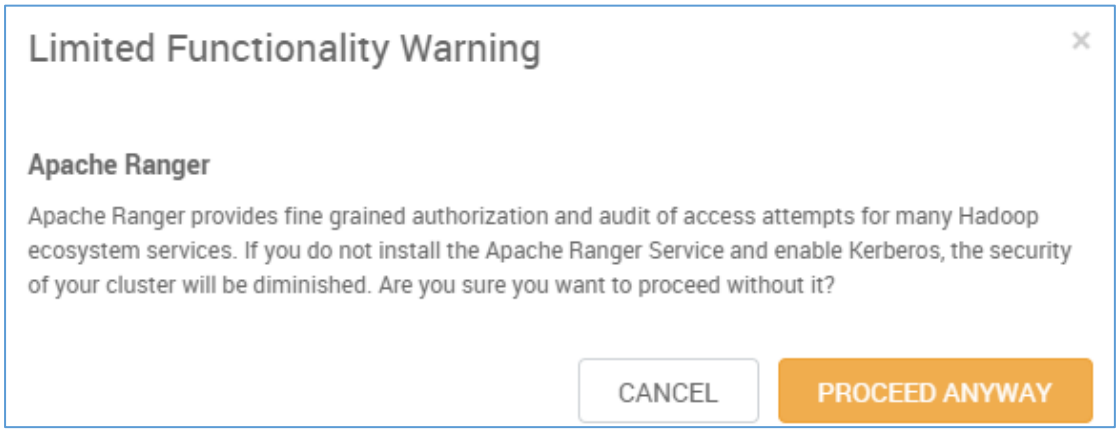
5.6 选择服务

这里会列出所有支持的组件，可根据需要进行选择安装，当然也可以初始化完成后后续再添加。我这里先安装 HDFS、ZooKEEPER、Ambari Metrics 等 3 个组件，另外 SmartSense 是必须组件，所以一共安装了 5 个组件。



下一步，会出现如下两个警告，分别是让安装 Apache Atlas、Apache Range 两个组件，这里暂时不安装，都选择 proceed anyway 即可

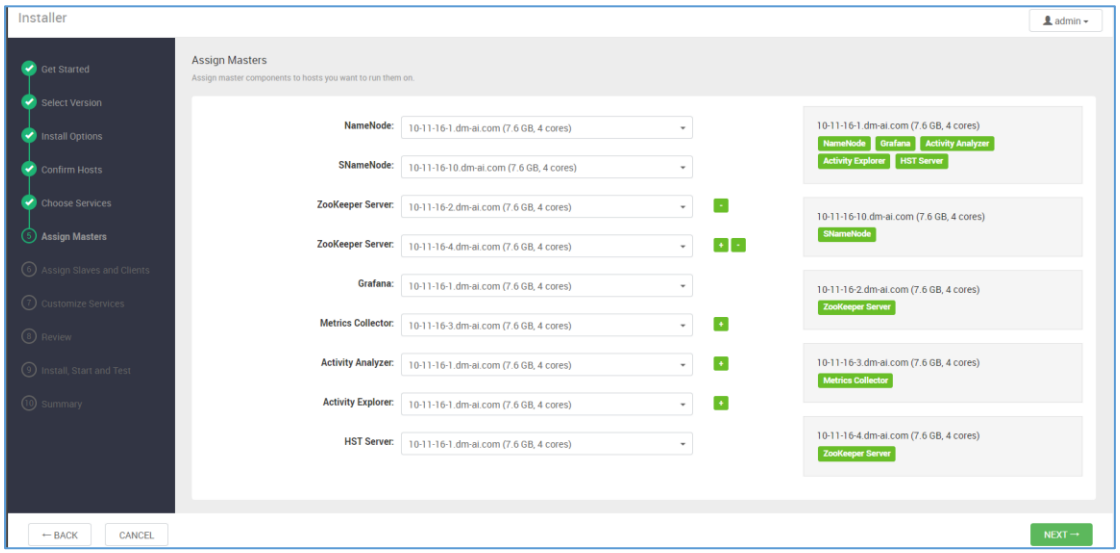




有可能会出现以下几种情况，都选择 continue 或 proceed anyway。

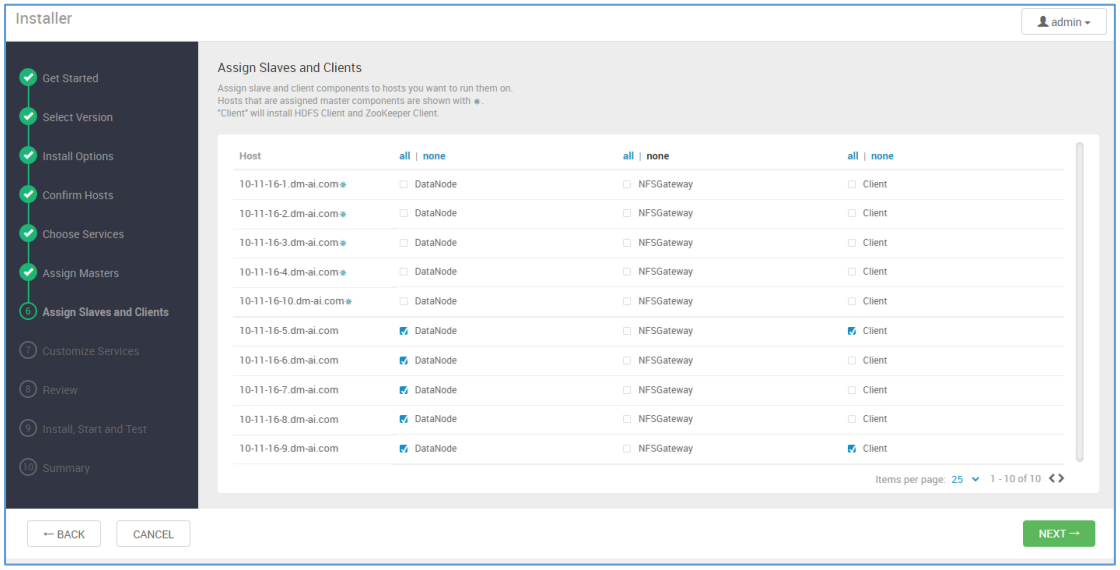
5.7 分配主要组件

分配主要组件，会有一个默认的划分，可以自己调整。中间是以组件为视角，右侧以主机为视角。

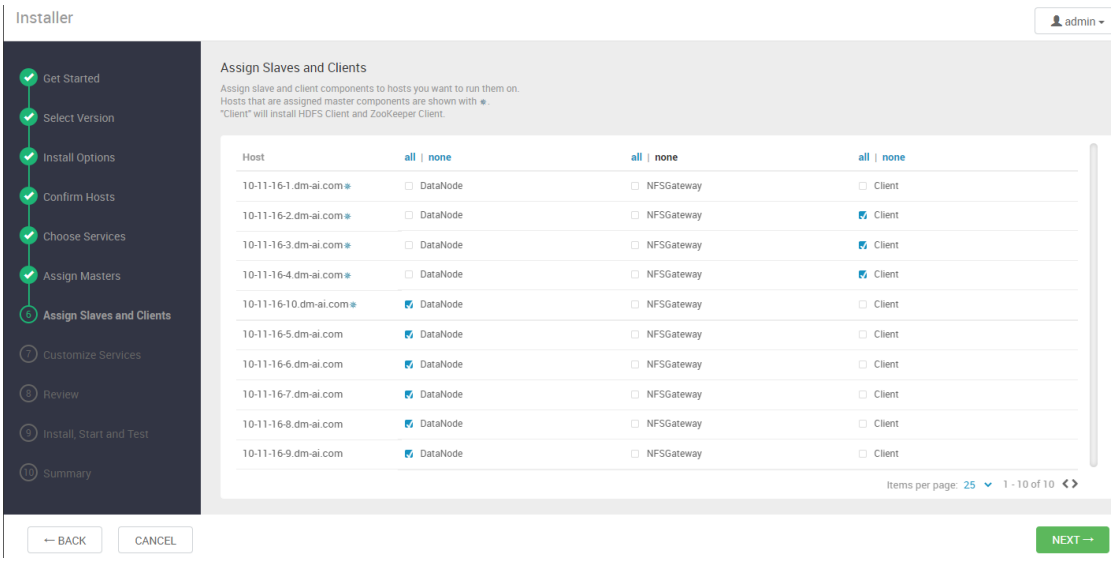


5.8 从属组件配置

如下图所示，前面 5 台已经安装了主要组件，会有*号标记，剩余的几台默认勾选了 DataNode 角色，可根据自己的需要进行更改。Client 角色是指 HDFS、ZK 的客户端。

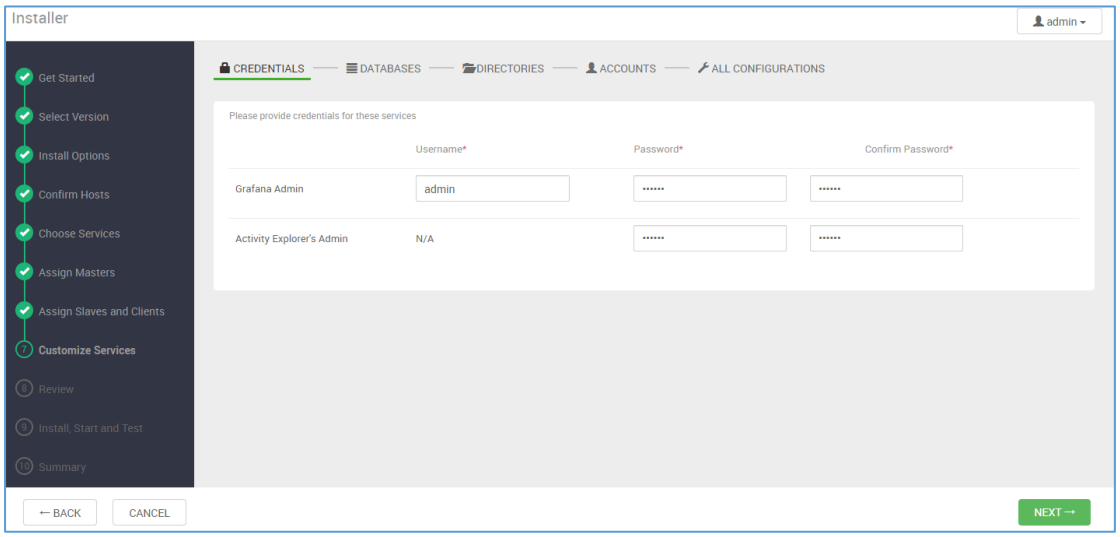


我进行了一个变更，如下

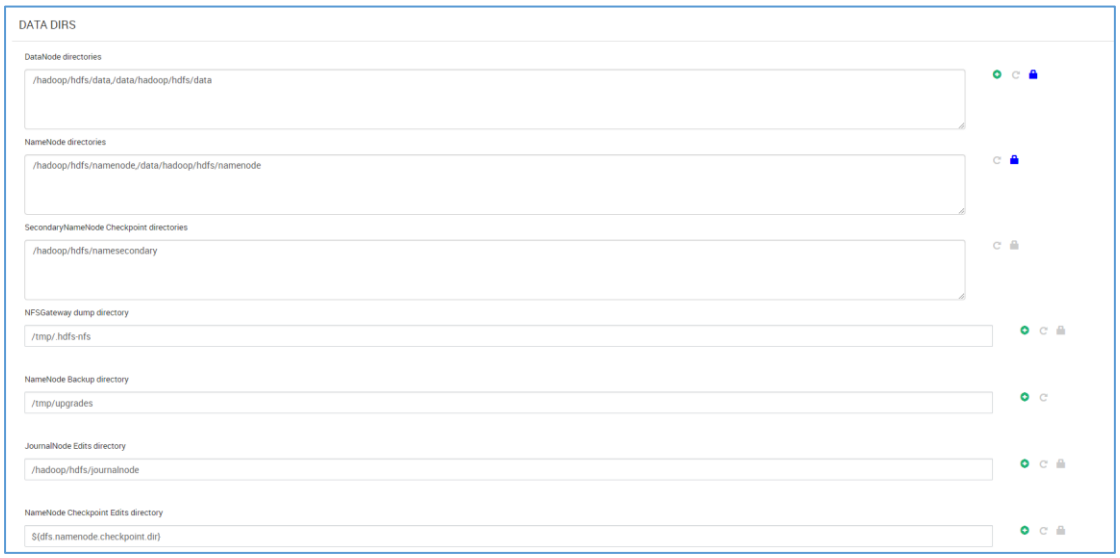


5.9 定制服务

输入相关密码，我这里为了方便，都设置为 123456

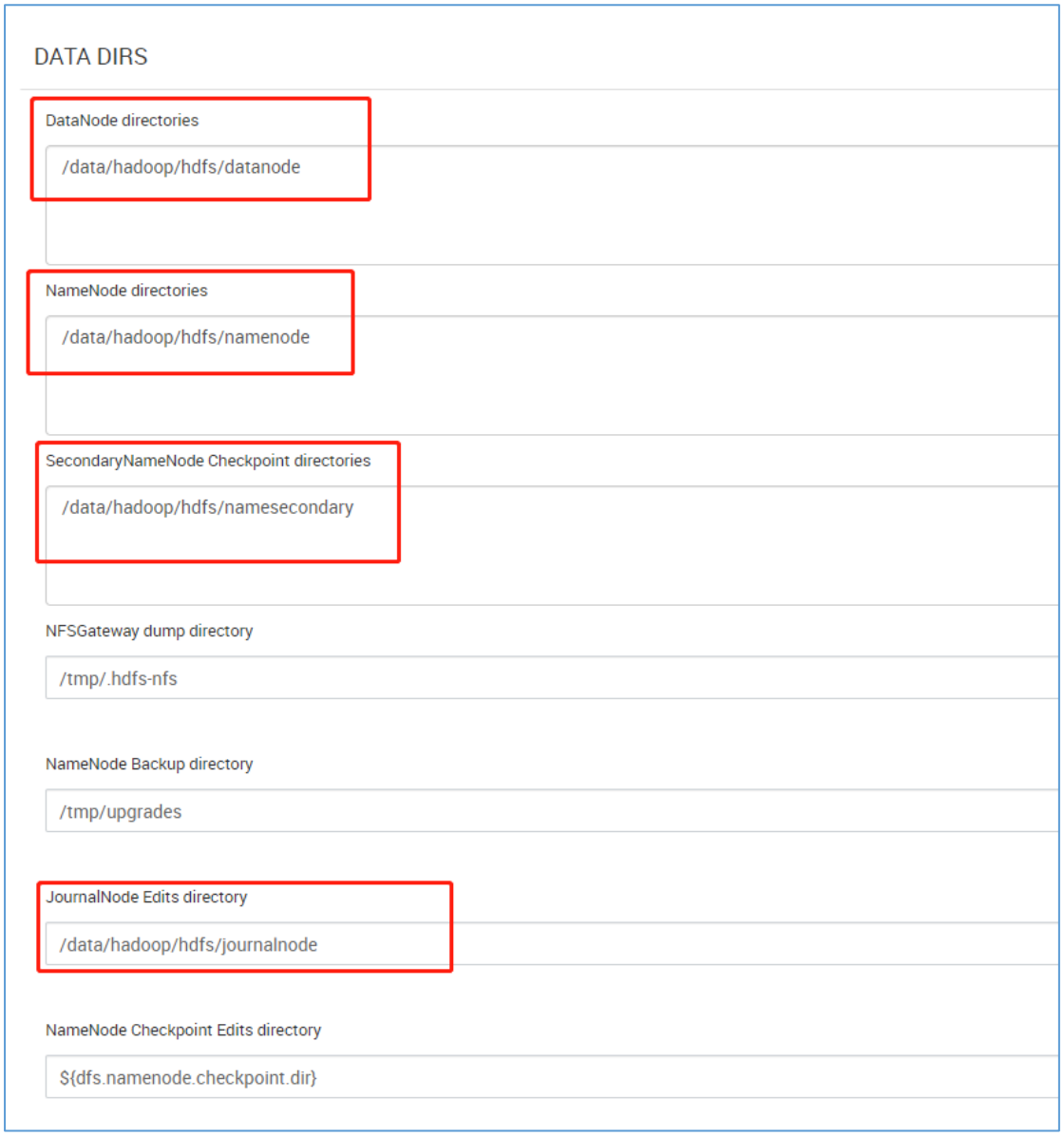


下一步配置相关目录，以下是默认情况，其中 datanode、namenode 使用了 directories（其他是 directory），所以可以配置多个路径，

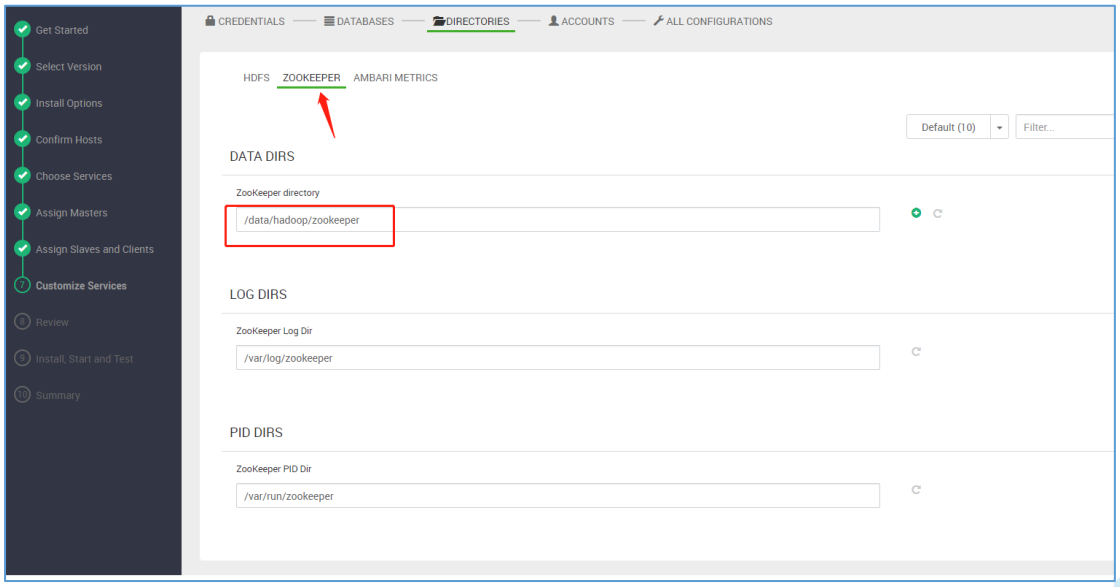


可根据实际情况进行修改，下图中红色部分是我修改过的

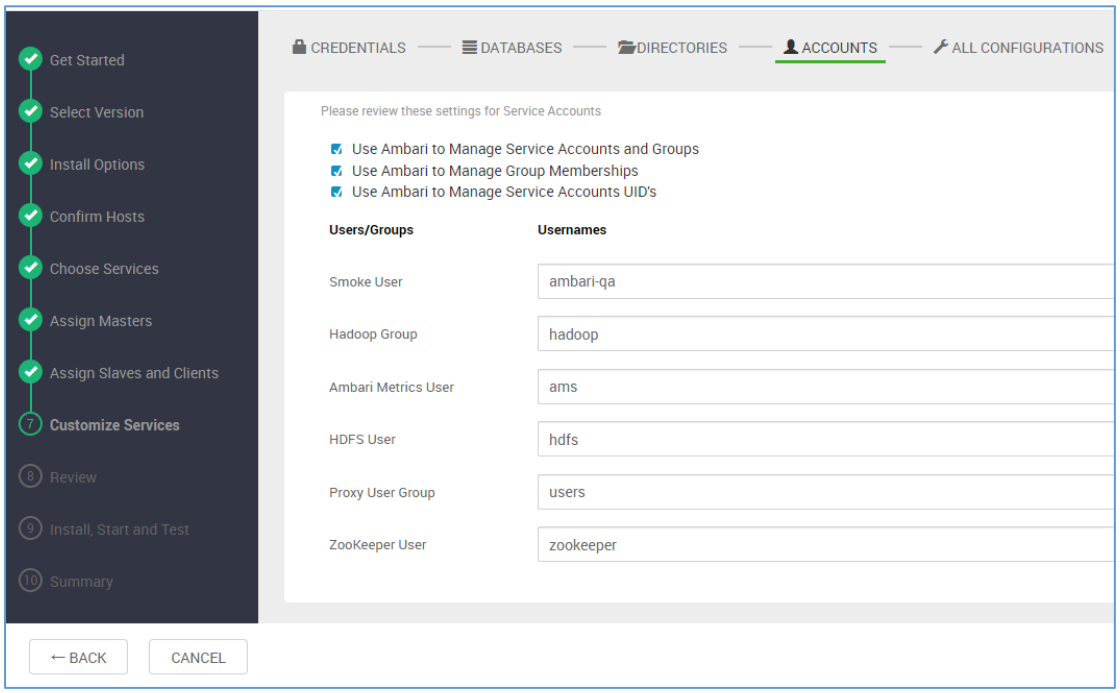
/data/hadoop/hdfs/datanode
/data/hadoop/hdfs/namenode
/data/hadoop/hdfs/namesecondary
/data/hadoop/hdfs/journalnode
/data/hadoop/zookeeper



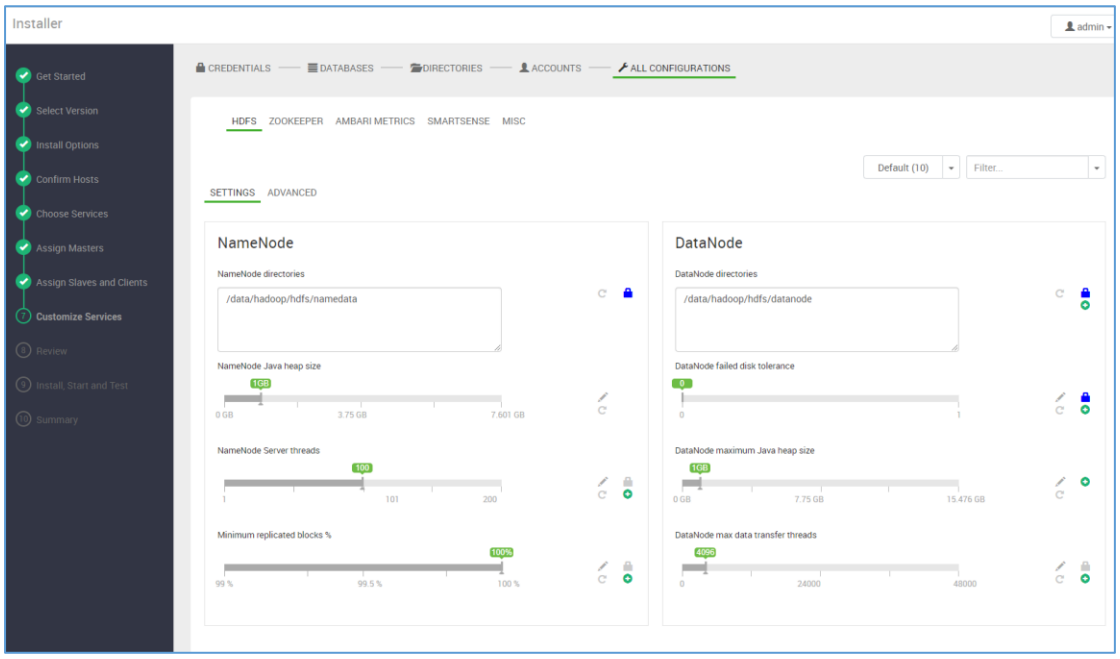
同时我也修改了 zk 的数据目录



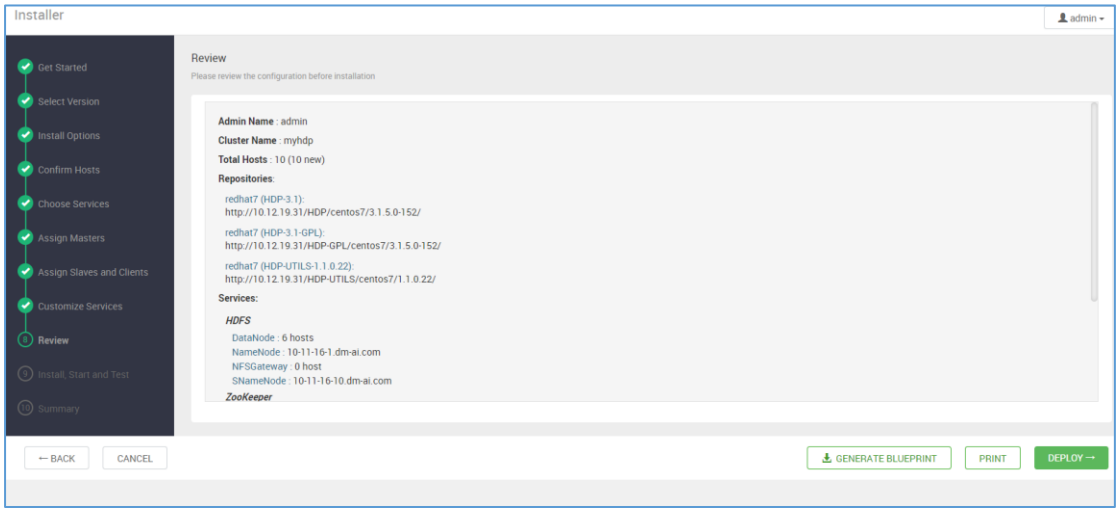
用户和组管理，这里我使用了默认配置，没有修改



所有配置，包含前面几步的配置，也有一些新的配置，可以根据需要定制。我这里没有修改。

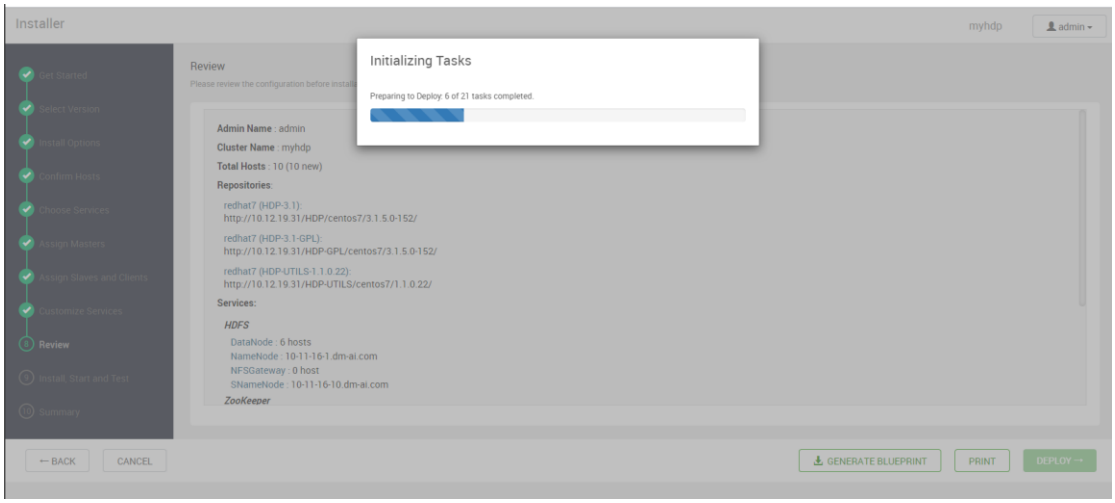


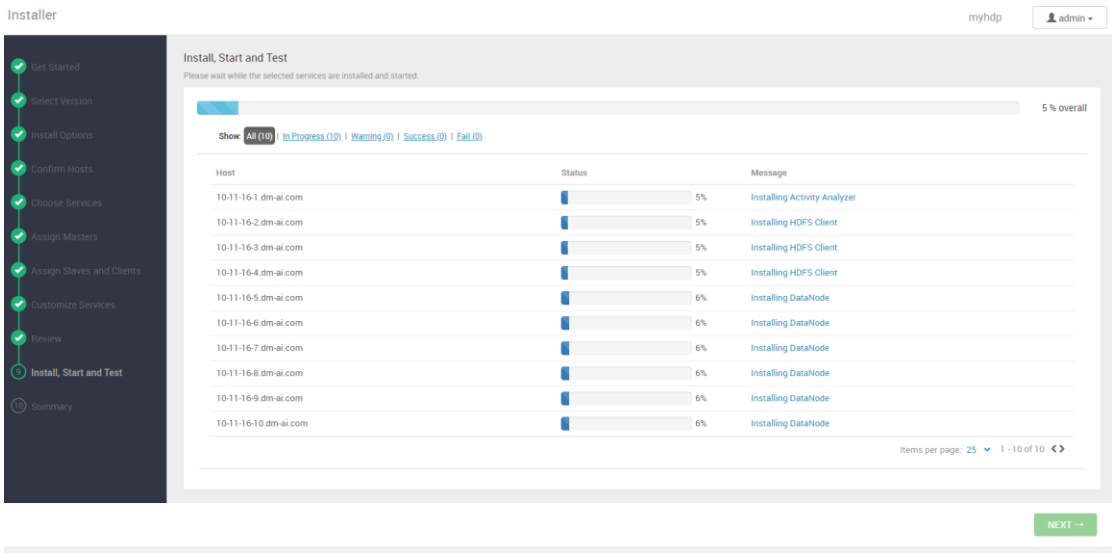
5. 10 review 概要

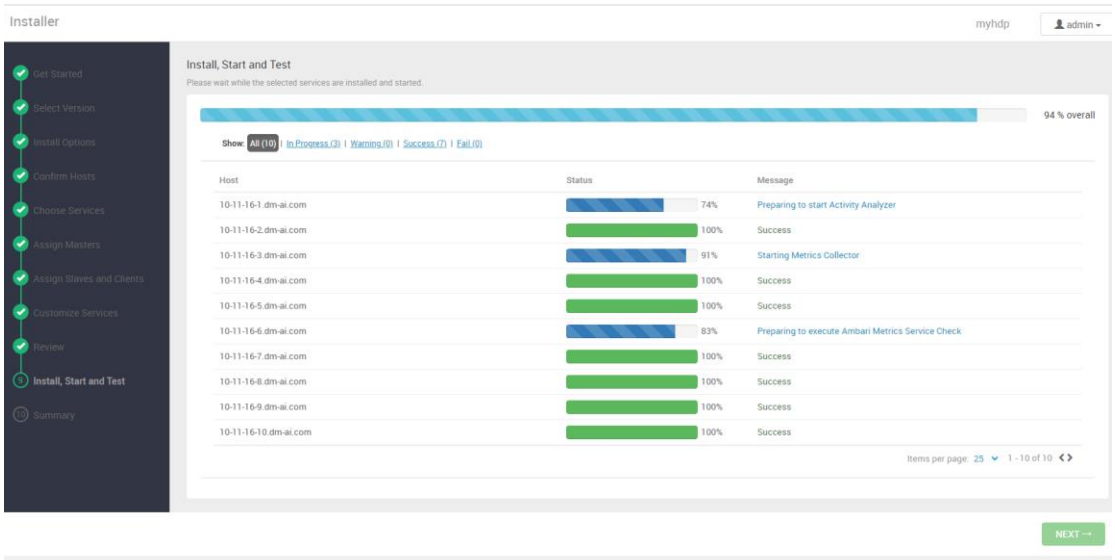


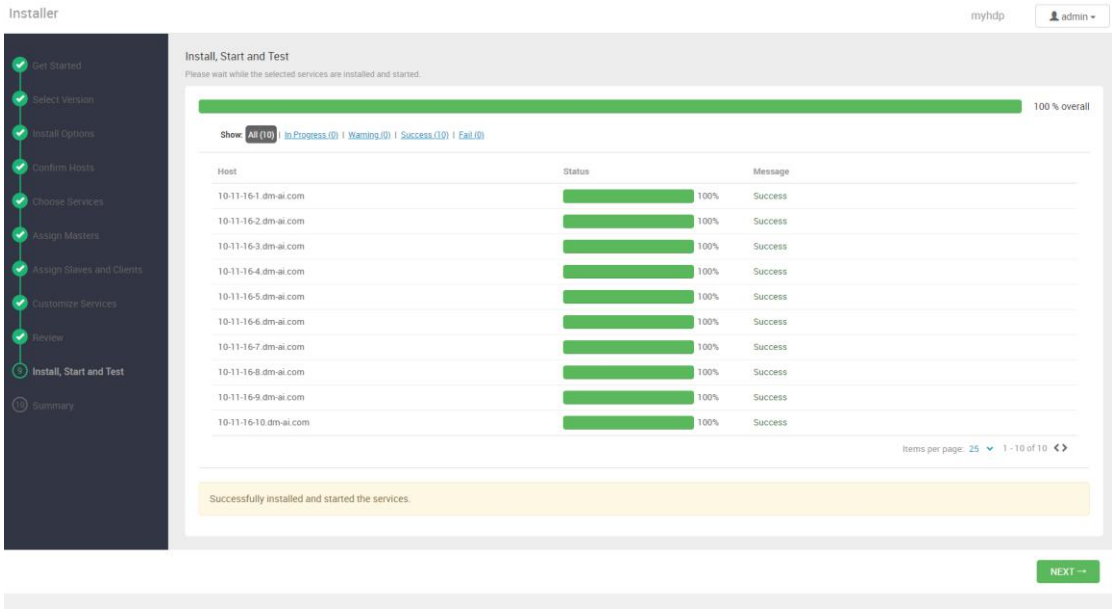
没有问题点击 deploy 进行部署。

5.11安装、启动和检测



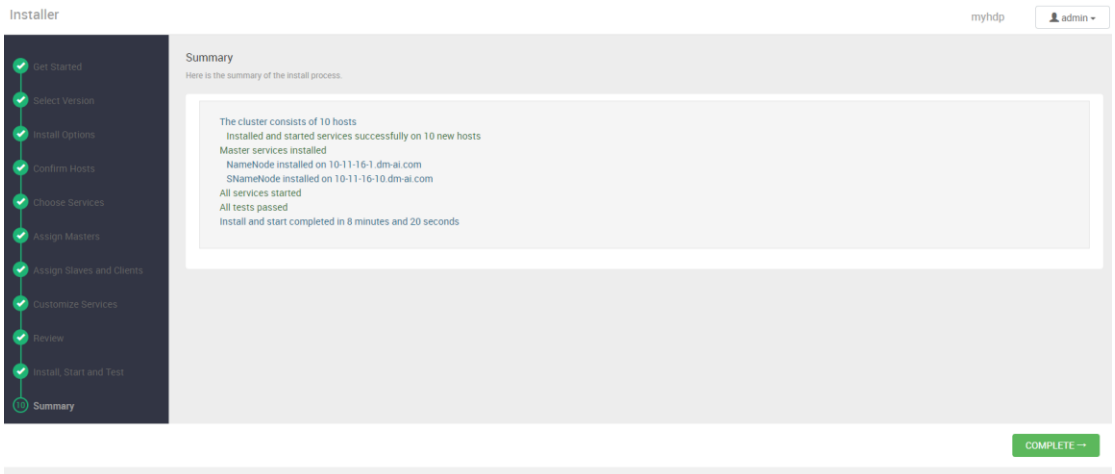


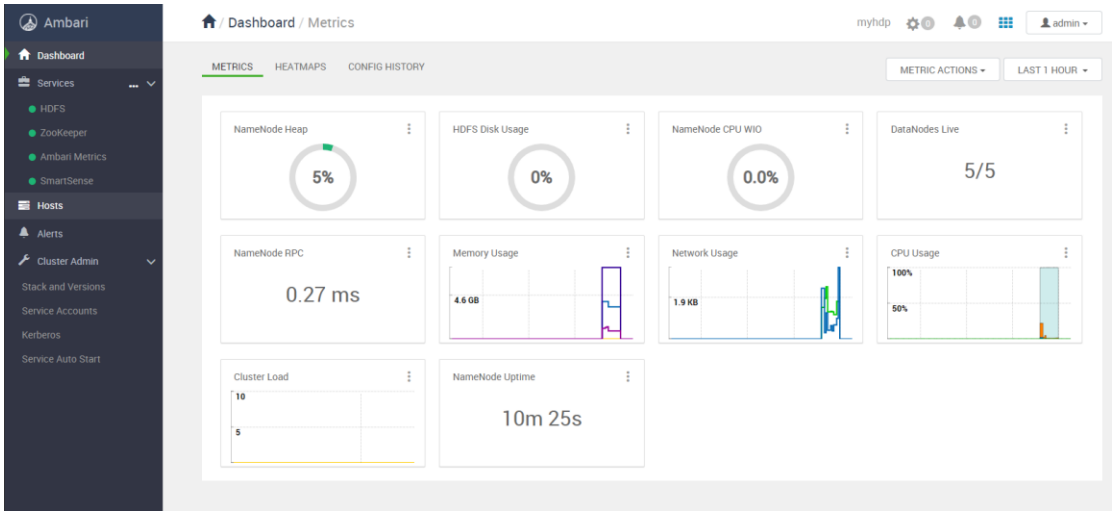


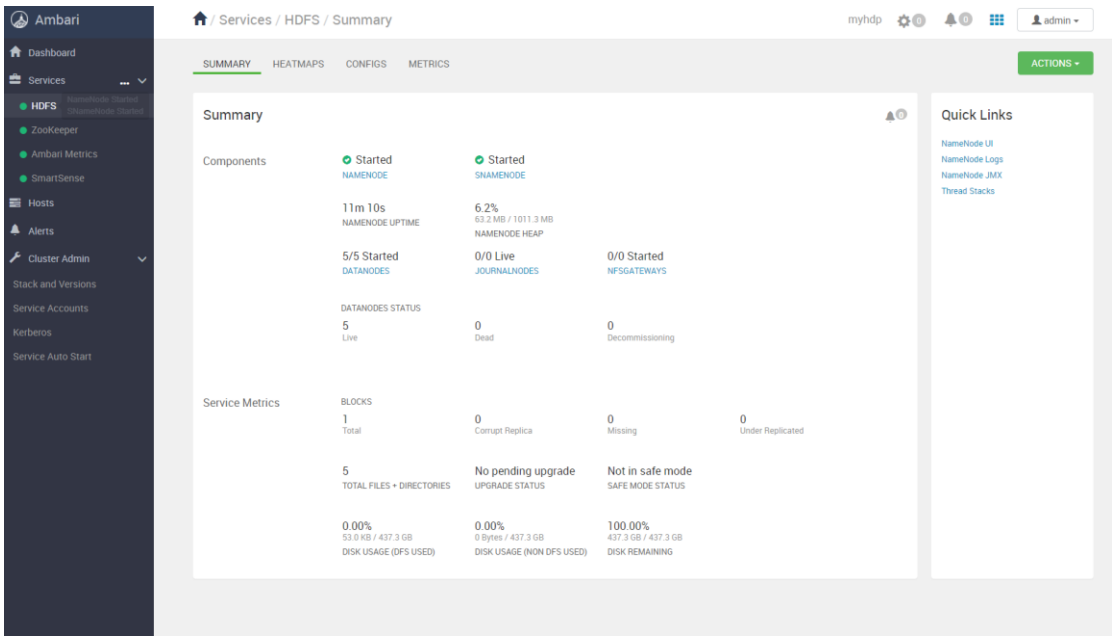


5.12 汇总

节点 1 安装了 NameNode,节点 10 安装了 SNameNode，这也是前面规划好的。



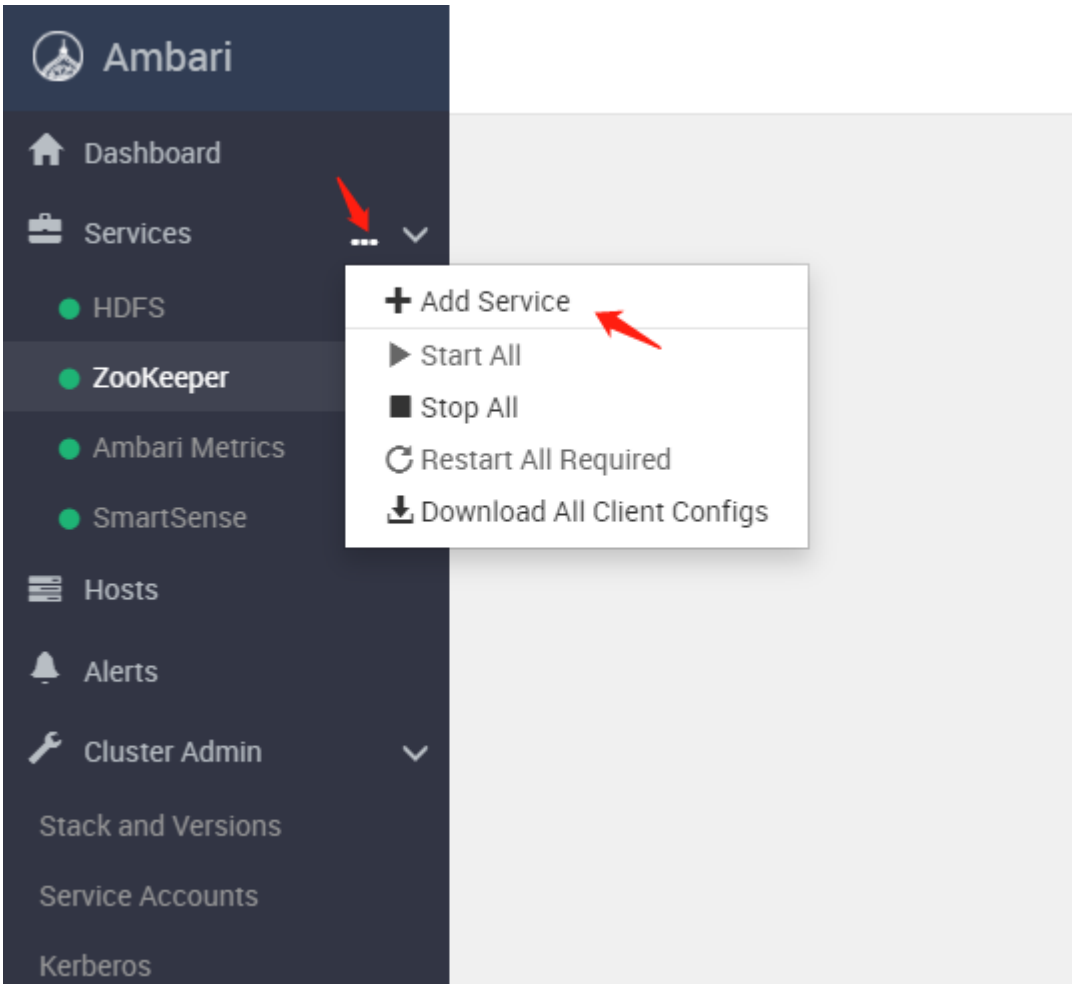

Ambari
Dashboard
Services
HDFS
ZooKeeper
Ambari Metrics
SmartSense
Hosts
Alerts
Cluster Admin
Stack and Versions
Service Accounts
Kerberos
Service Auto Start
/ Dashboard / Metrics
myhdp
admin
METRICS
HEATMAPS
CONFIG HISTORY
METRIC ACTIONS
LAST 1 HOUR
NameNode Heap
5%
HDFS Disk Usage
0%
NameNode CPU WIO
0.0%
DataNodes Live
5/5
NameNode RPC
0.27 ms
Memory Usage
4.6 GB
Network Usage
1.9 KB
CPU Usage
100%
50%
Cluster Load
10
5
NameNode Uptime
10m 25s


Ambari
Dashboard
Services
HDFS
ZooKeeper
Ambari Metrics
SmartSense
Hosts
Alerts
Cluster Admin
Stack and Versions
Service Accounts
Kerberos
Service Auto Start
/ Services / HDFS / Summary
myhdp
admin
SUMMARY
HEATMAPS
CONFIGS
METRICS
ACTIONS
Summary
Components
Started
NAMENODE
Started
SNAMENODE
11m 10s
NAMENODE UPTIME
6.2%
63.2 MB / 1011.3 MB
NAMENODE HEAP
5/5 Started
DATANODES
0/0 Live
JOURNALNODES
0/0 Started
NFSGATEWAYS
DATANODES STATUS
5
Live
0
Dead
0
Decommissioning
Service Metrics
BLOCKS
1
Total
0
Corrupt Replica
0
Missing
0
Under Replicated
5
TOTAL FILES + DIRECTORIES
No pending upgrade
UPGRADE STATUS
Not in safe mode
SAFE MODE STATUS
0.00%
53.0 KB / 437.3 GB
DISK USAGE (DFS USED)
0.00%
0 Bytes / 437.3 GB
DISK USAGE (NON DFS USED)
100.00%
437.3 GB / 437.3 GB
DISK REMAINING
Quick Links
NameNode UI
NameNode Logs
NameNode JMX
Thread Stacks

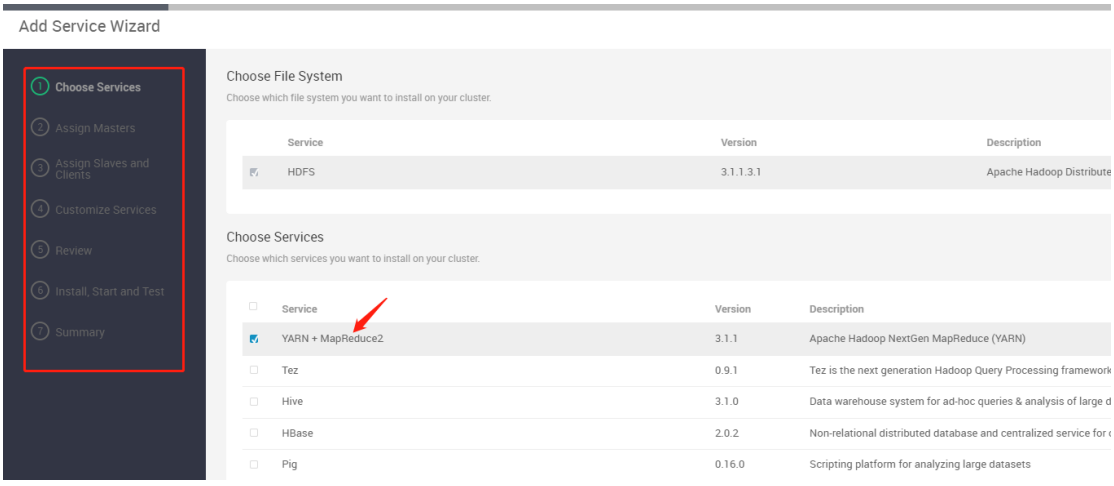
6 添加服务

6.1 添加 yarn+mapreduce2 服务

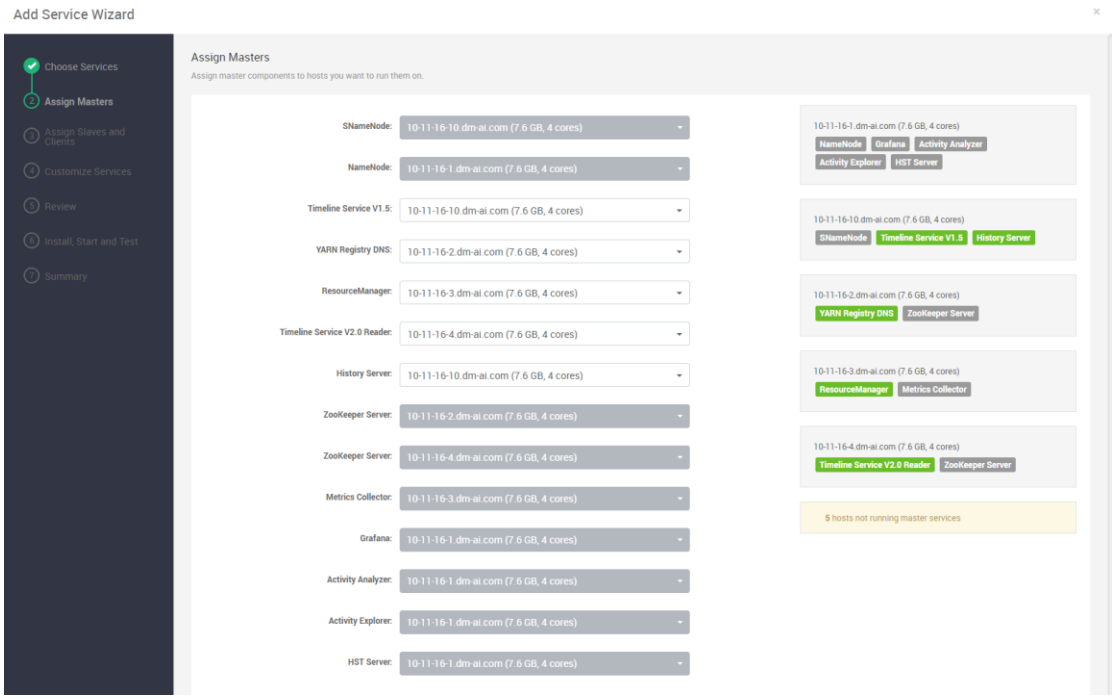
点击 add service



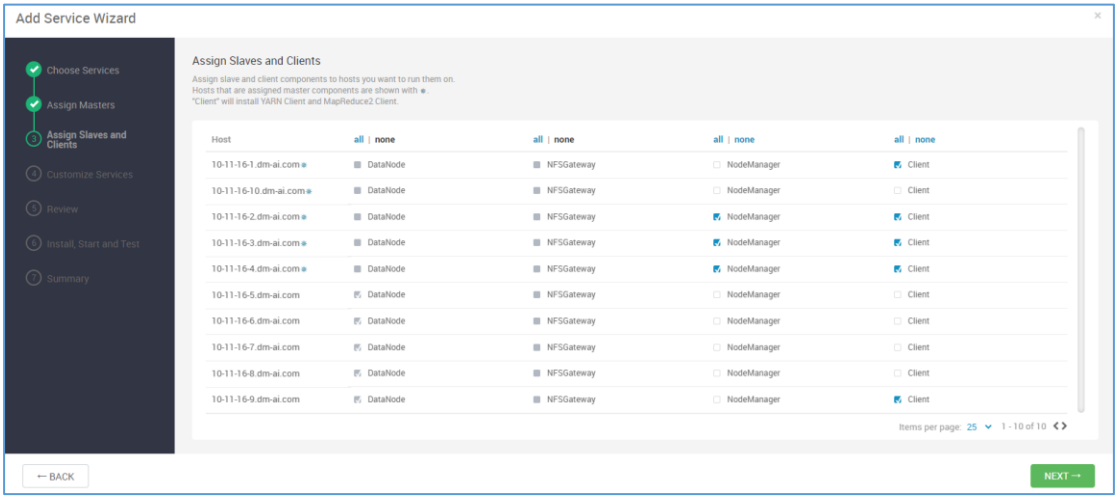
选择 YARN+MapReduce2



从图中也可以看到，服务添加和前一章节的 hadoop 部署中后面的几步是相同的。

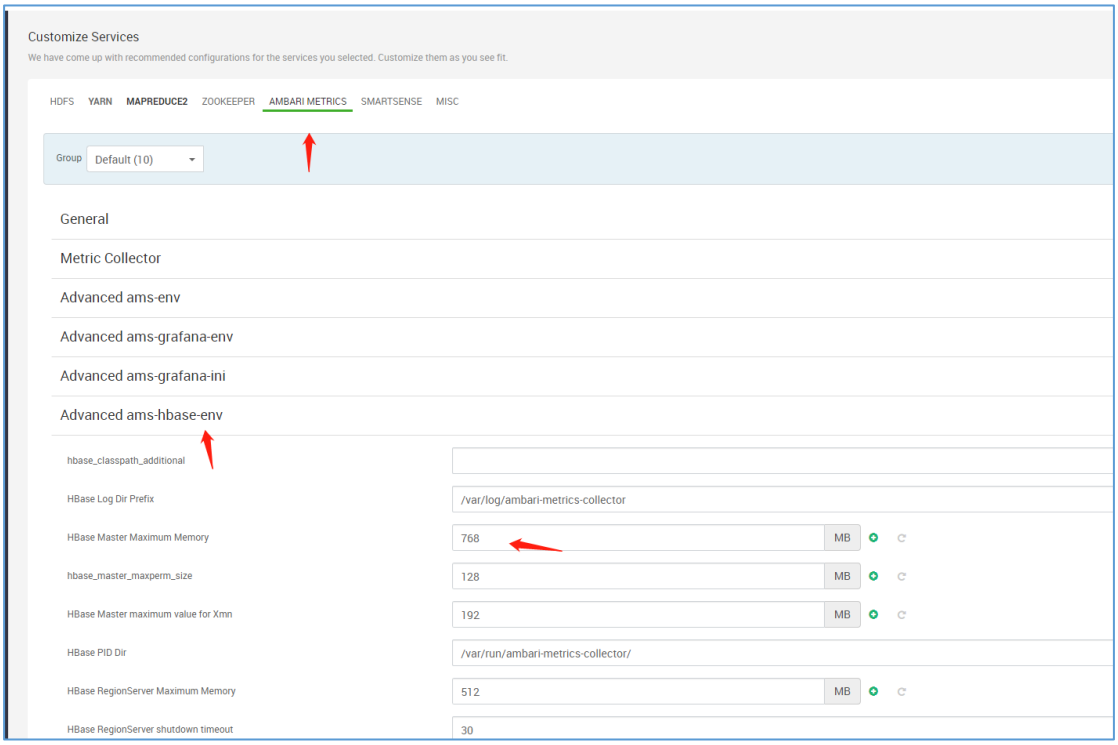


如下图配置 NodeManager 节点和 Client 节点，这里的 client 值 yarn、mapreduce2 的 client，client 的意思可看图中的英文说明，后面章节不再单独说。



出现如下报错

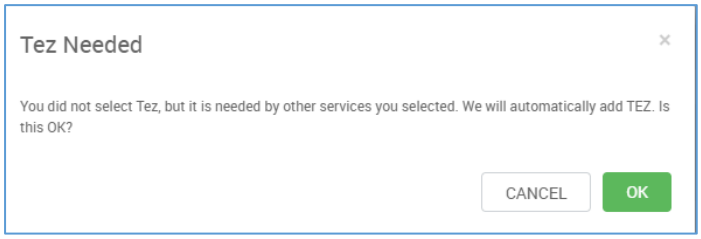
找到如下位置，修改为 1024



后面的步骤不再单独列出。

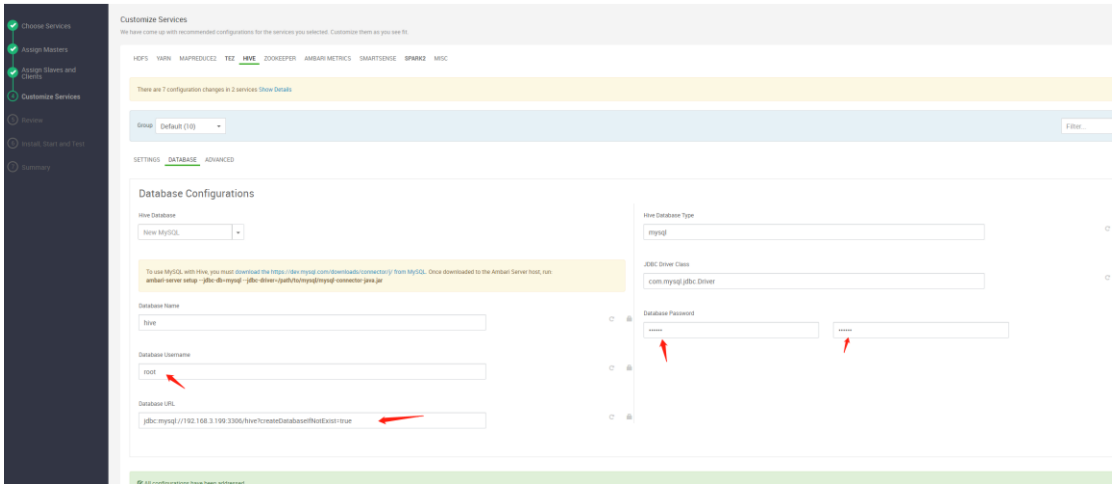
6.2 添加 hive 服务

依赖 Tez 服务，一并安装下



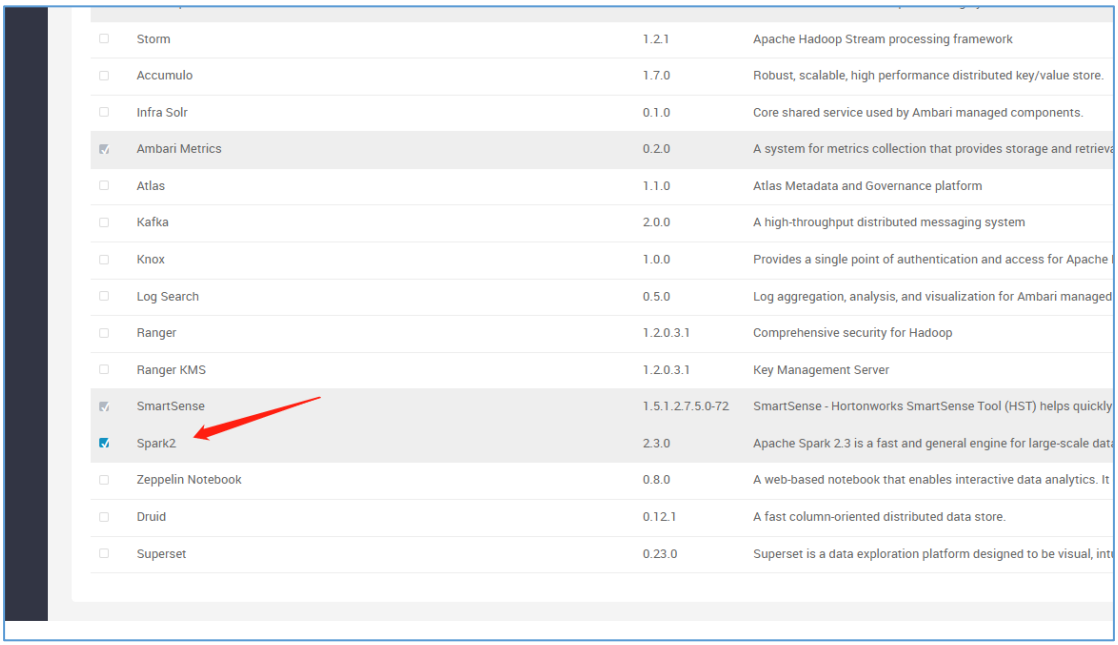
注意：Hive Metastore 和 HiveServer2 要在同一节点。

关键步骤，配置 hive 的数据库



6.3 添加 spark 服务

点击 add service，选择 Spark2

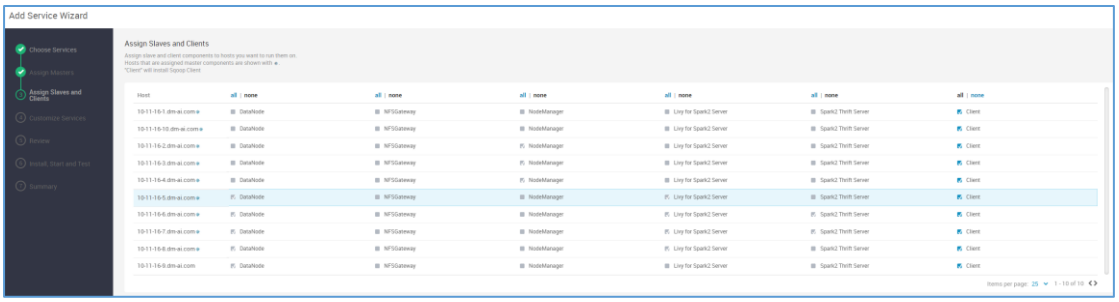


我这里了选了 2 个节点安装 spark2 history server，依赖于 hdfs client、mapreduces2 client、yarn client、hive client 服务

注意：spark2 thrift server 依赖 spark2 history server，前者要装到后者所在节点上。

后续步骤略

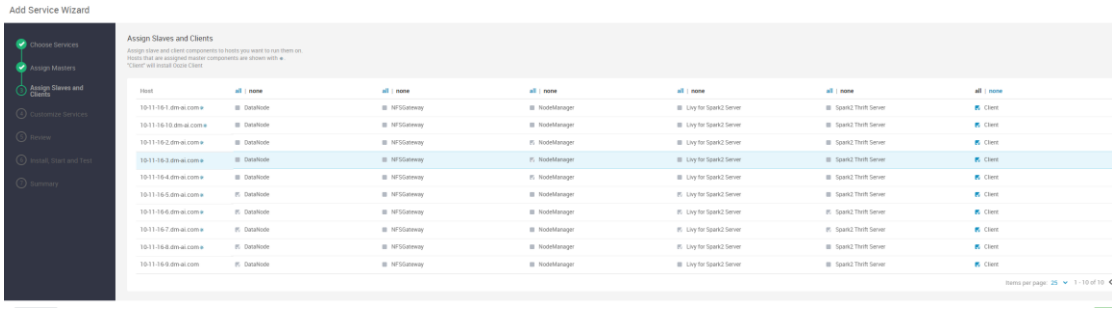
6.4 添加 sqoop 服务



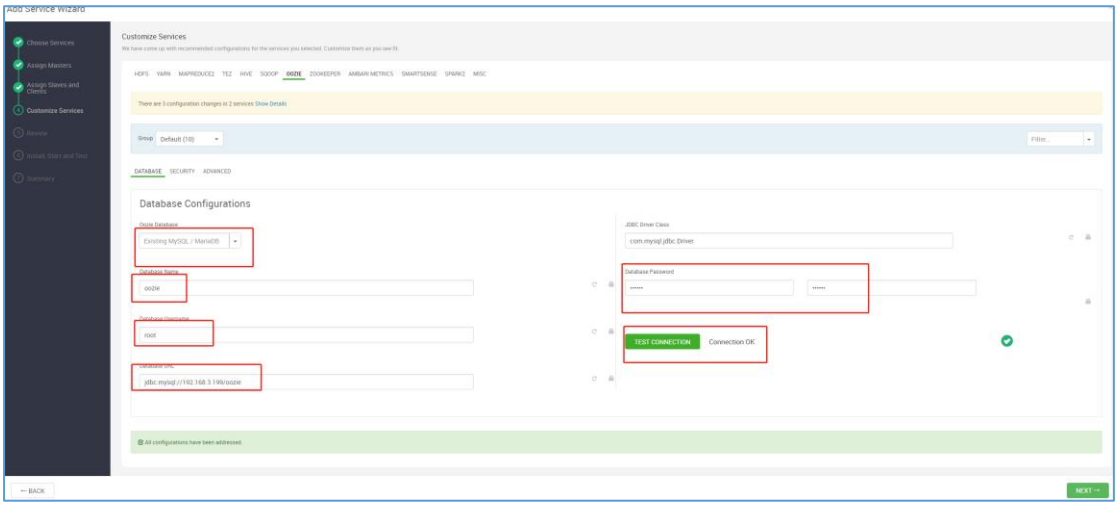
6.5 添加 oozie 服务

提前创建好 oozie 库

```
mysql>
mysql> create database oozie;
Query OK, 1 row affected
```



如下图所示，配置数据库相关信息

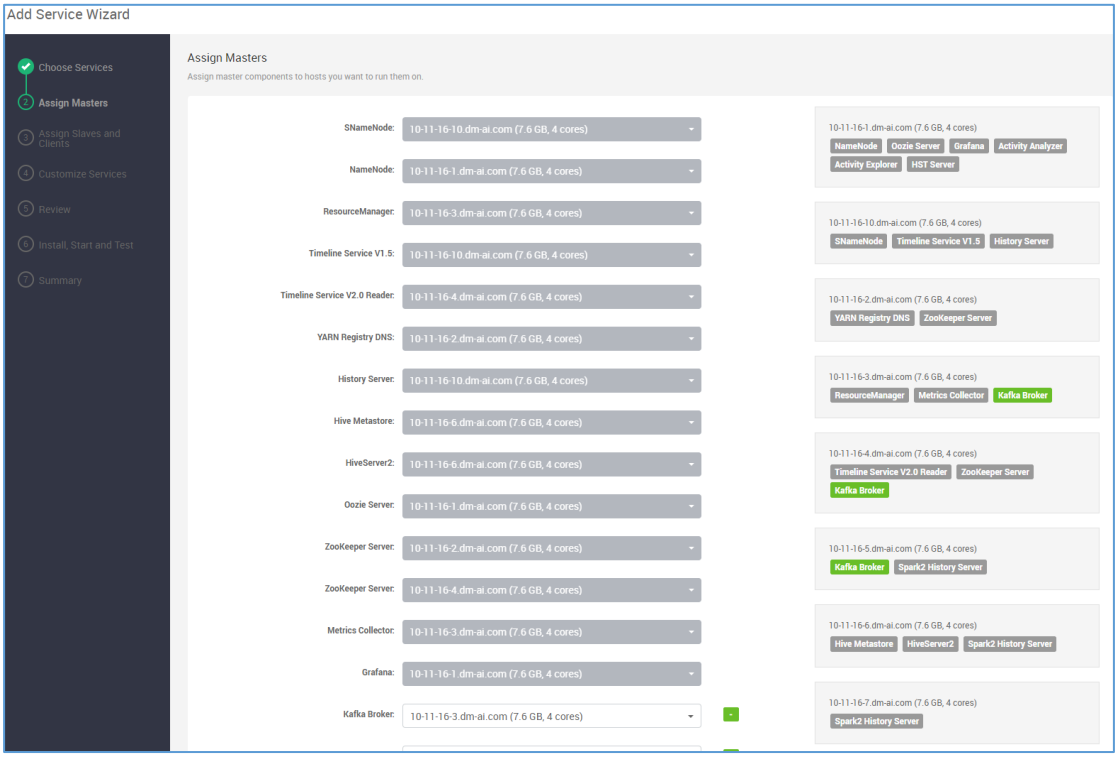


jdbc:mysql://192.168.3.199/oozie

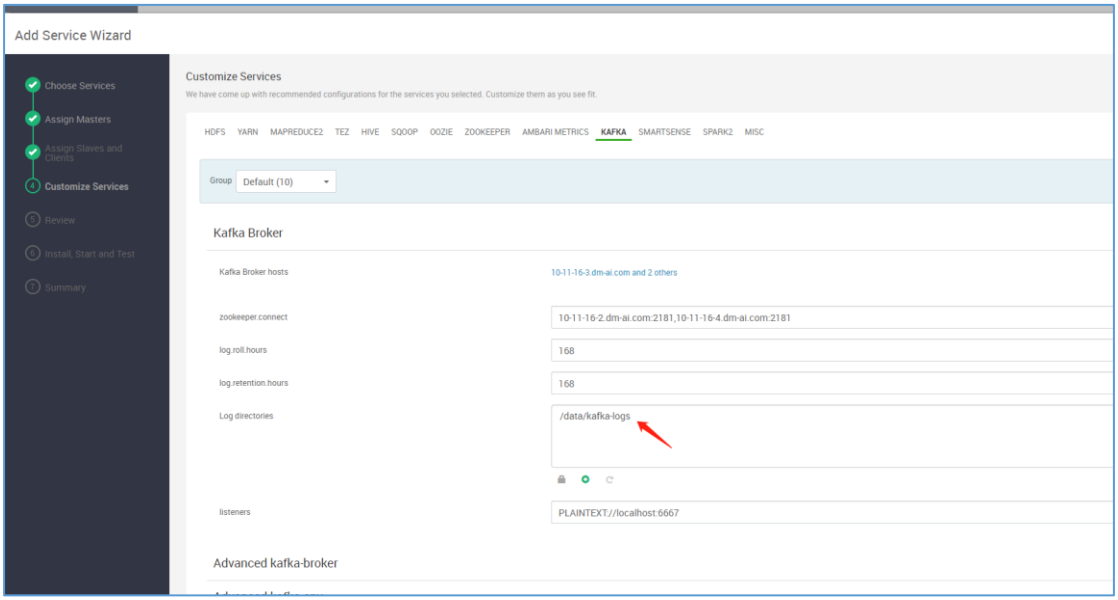
需要测试通过。

下面的步骤略。

6.6 添加 kafka 服务

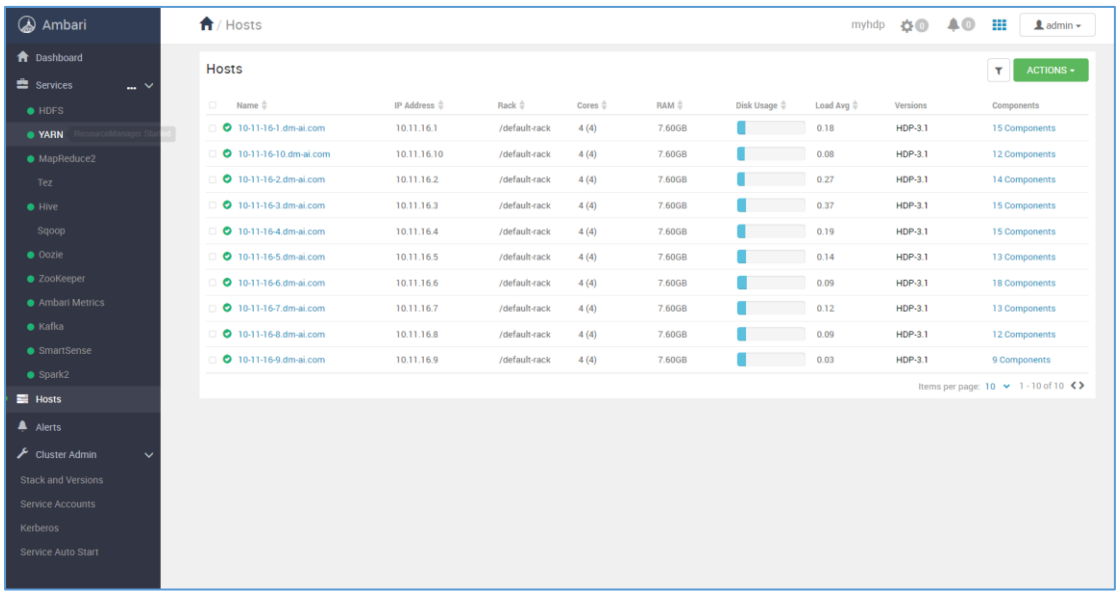


这里我修改了 kafka 的 logs 目录地址



6.7 概览

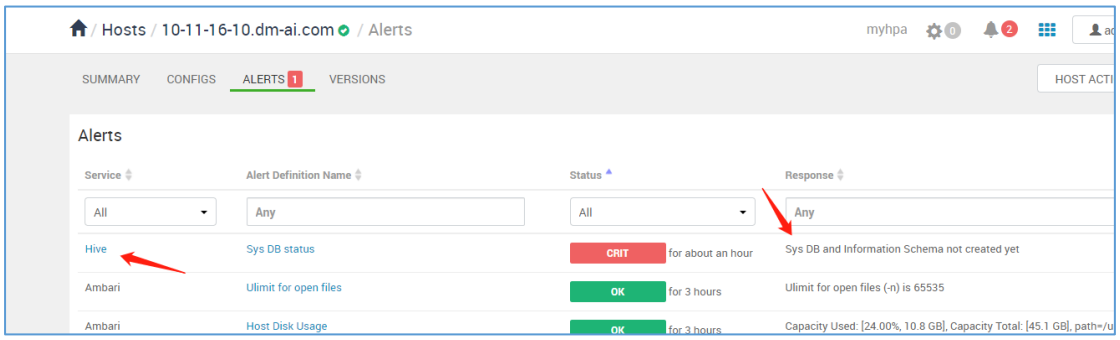
添加完成以上几个服务之后，控制台查看效果如下


Ambari
Dashboard
Services
HDFS
YARN
MapReduce2
Tez
Hive
Sqoop
Oozie
ZooKeeper
Ambari Metrics
Kafka
SmartSense
Spark2
Hosts
Alerts
Cluster Admin
Stack and Versions
Service Accounts
Kerberos
Service Auto Start
/ Hosts
myhdp
admin
Hosts
ACTIONS
Name
IP Address
Rack
Cores
RAM
Disk Usage
Load Avg
Versions
Components
10-11-16-1.dm-ai.com 10.11.16.1 /default-rack 4 (4) 7.60GB 0.18 HDP-3.1 15 Components
10-11-16-10.dm-ai.com 10.11.16.10 /default-rack 4 (4) 7.60GB 0.08 HDP-3.1 12 Components
10-11-16-2.dm-ai.com 10.11.16.2 /default-rack 4 (4) 7.60GB 0.27 HDP-3.1 14 Components
10-11-16-3.dm-ai.com 10.11.16.3 /default-rack 4 (4) 7.60GB 0.37 HDP-3.1 15 Components
10-11-16-4.dm-ai.com 10.11.16.4 /default-rack 4 (4) 7.60GB 0.19 HDP-3.1 15 Components
10-11-16-5.dm-ai.com 10.11.16.5 /default-rack 4 (4) 7.60GB 0.14 HDP-3.1 13 Components
10-11-16-6.dm-ai.com 10.11.16.6 /default-rack 4 (4) 7.60GB 0.09 HDP-3.1 18 Components
10-11-16-7.dm-ai.com 10.11.16.7 /default-rack 4 (4) 7.60GB 0.12 HDP-3.1 13 Components
10-11-16-8.dm-ai.com 10.11.16.8 /default-rack 4 (4) 7.60GB 0.09 HDP-3.1 12 Components
10-11-16-9.dm-ai.com 10.11.16.9 /default-rack 4 (4) 7.60GB 0.03 HDP-3.1 9 Components
Items per page: 10
1 - 10 of 10

7 排错

7.1 hive 安装报警

有如下警告：Sys DB and Information Schema not created yet

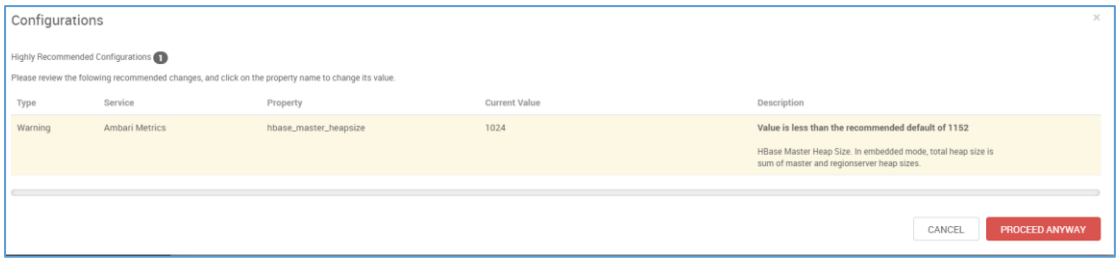


解决方法：进入 hive metastore 所在的节点，执行如下命令即可：

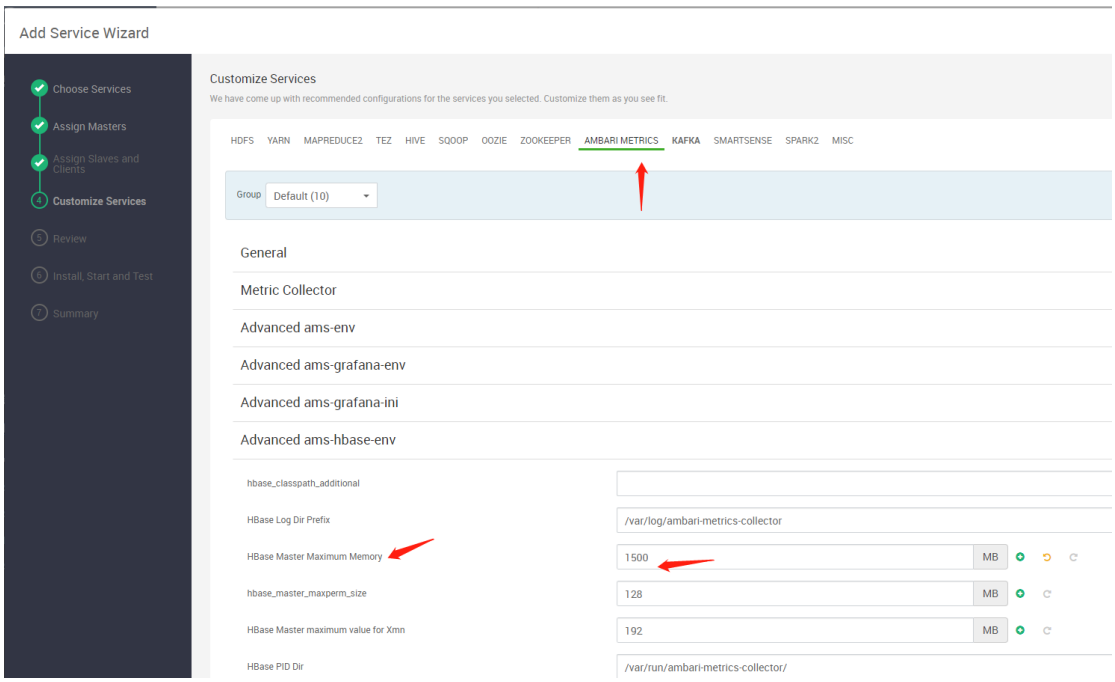
touch /etc/hive/sys.db.created

7.2 kafka 安装报警

如果出现以下警告

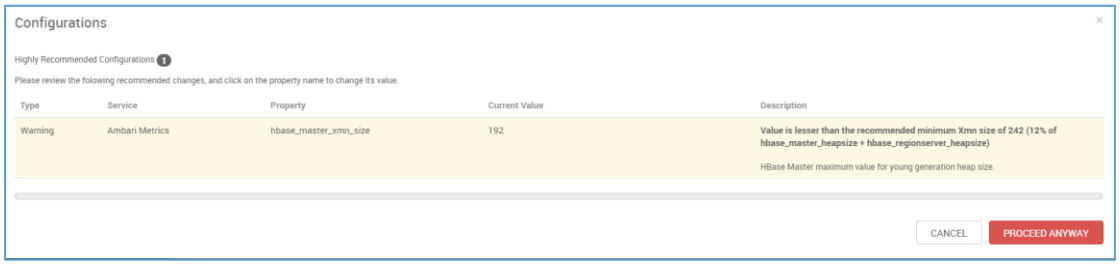


如下图，将这个配置改为 1500



以下告警：

Value is lesser than the recommended minimum Xmn size of 242 (12% of hbase_master_heapsize + hbase_regionserver_heapsize)
HBase Master maximum value for young generation heap size.



1500+512 的 12%等于 242，也就是 hbase master maximum value for xmn 至少要是 242，我这里改为 250