



TDH-CE-DEV 2023-5 安装手册

星环信息科技（上海）股份有限公司

版本号 T00993x-01-96, 2023-05

目录

1. 安装前说明	2
1.1. 资源要求	2
1.2. 操作前提	2
2. 安装TDH-CE-DEV	3
2.1. 安装前系统配置改动	3
2.2. 环境准备	3
2.3. 安装TDH社区开发版	4
2.4. 自助申请许可证	10
3. 其他使用方面的操作	13
3.1. 开启安全	13
3.2. 使用核心组件Hyperbase的操作设置	14
3.3. 容器重启后的操作	16
3.4. 快速下载TDH-Client	17
3.5. 集群外使用hadoop client	19
3.6. 如何查看资源占用	20
4. 常见的问题排查方法	21
客户服务	23

免责声明

本说明书依据现有信息制作, 其内容如有更改, 恕不另行通知。星环信息科技(上海)股份有限公司在编写该说明书的时候已尽最大努力保证期内容准确可靠, 但星环信息科技(上海)股份有限公司不对本说明书中的遗漏、不准确或印刷错误导致的损失和损害承担责任。具体产品使用请以实际使用为准。

注释: Hadoop® 和 SPARK® 是ApacheTM 软件基金会在美国和其他国家的商标或注册的商标。Java® 是Oracle公司在美国和其他国家的商标或注册的商标。Intel® 和Xeon® 是英特尔公司在美国、中国和其他国家的商标或注册的商标。

版权所有 © 2013年-2022年星环信息科技(上海)股份有限公司。保留所有权利。

©星环信息科技(上海)股份有限公司版权所有, 并保留对本说明书及本声明的最终解释权和修改权。本说明书的版权归星环信息科技(上海)股份有限公司所有。未得到星环信息科技(上海)股份有限公司的书面许可, 任何人不得以任何方式或形式对本说明书内的任何部分进行复制、摘录、备份、修改、传播、翻译成其他语言、或将其全部或部分用于商业用途。

手册版本信息

版本号: T00993x-01-96

发布日期: 2023-05

1. 安装前说明

本手册将指导您进行TDH社区开发版（后续简称：TDH-CE-DEV）的安装。此产品包目前暂时还不支持通过window部署的形式，预计在7-8月将支持windows，具体动向请关注星环开发者社区或添加官方小助手进入社区版交流群第一时间获悉产品动向。

1.1. 资源要求

目前社区开发版需要在安装了Docker的Linux系统上运行，仅使用最基础组件（Inceptor）所需的最低资源要求为：

- 1颗4核心或以上带超线程x86指令集CPU的服务器
- 16GB以上内存

如果您需要使用多个组件，如Aquila/Guardian/Hyperbase等等，为了您更好的体验以及更稳定的性能，建议配置：

- 1颗8核心或以上带超线程x86指令集CPU的服务器
- 32GB以上内存

您可以基于自身需求调整资源。

1.2. 操作前提

为实现持久化，您需要把本地目录挂载到容器内的/opt/transwarp目录下。挂载后服务的配置，日志，数据都会放在这个本地目录下，同时，服务的数据目录默认也都会配置到/opt/transwarp下面，请勿随意修改该配置并做好备份，如果配到了其他路径，那么则不能实现持久化了。 注意：不能持久化则意味着当容器被删除后，数据将会丢失。



当环境变更时，启动新的容器可挂载到保存的该路径，TDH集群配置/日志和用户数据库数据都会保留并可继续使用。

如果安装过程中有任何问题，请随时与社区版官方小助手联系。



2. 安装TDH-CE-DEV

2.1. 安装前系统配置改动

在安装前，请打开环境下的/etc/hosts文件，确保该文件包含节点的hostname和IP地址的映射关系列表。例如：

```
172.16.2.125 tw-node2125
```

您可以登入到节点使用 `hostname` 命令检查节点主机名，确保节点名称与上述 /etc/hosts 里设置的名称相同。请注意，集群的节点名称不能重复，且必须符合DNS-1123规范，由数字、小写字母或“-”组成，不能包含大写字母，长度小于63。

您可以使用 `hostnamectl set-hostname <hostname>` 命令来修改主机名，参考以下示例：

```
[root@localhost ~]# hostnamectl set-hostname tw-node2125
[root@localhost ~]# hostname
tw-node2125
[root@localhost ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@localhost ~]# vi /etc/hosts
[root@localhost ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
172.16.2.125 tw-node2125
```

请注意 /etc/hosts 文件的第一行必须为127.0.0.1的记录，不能将此行注释掉，也不能将当前主机名写在该行中。通常如下：



```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
```

2.2. 环境准备

1. 步骤一 Linux环境上安装docker

此处以Centos系统为例

```
yum install docker
```

```
[root@Intest122127 ~]# docker version
-bash: docker: command not found
[root@Intest122127 ~]# yum install docker
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
base                                     | 3.6 kB  00:00:00
extras                                  | 2.9 kB  00:00:00
updates                                 | 2.9 kB  00:00:00
(1/2): extras/7/x86_64/primary_db       | 249 kB  00:00:00
(2/2): updates/7/x86_64/primary_db     | 21 MB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 2:1.13.1-209.git7d71120.el7.centos will be installed
--> Processing Dependency: docker-common = 2:1.13.1-209.git7d71120.el7.centos for package: 2:docker-1.13.1-209.git7d71120.el7.centos.x86_64
--> Processing Dependency: docker-client = 2:1.13.1-209.git7d71120.el7.centos for package: 2:docker-1.13.1-209.git7d71120.el7.centos.x86_64
--> Processing Dependency: subscription-manager-rhsm-certificates for package: 2:docker-1.13.1-209.git7d71120.el7.centos.x86_64
--> Running transaction check
--> Package docker-client.x86_64 2:1.13.1-209.git7d71120.el7.centos will be installed
--> Package docker-common.x86_64 2:1.13.1-209.git7d71120.el7.centos will be installed
--> Processing Dependency: skopeo-containers >= 1:0.1.26-2 for package: 2:docker-common-1.13.1-209.git7d71120.el7.centos.x86_64
--> Processing Dependency: oci-umount >= 2:2.3.3-3 for package: 2:docker-common-1.13.1-209.git7d71120.el7.centos.x86_64
--> Processing Dependency: oci-systemd-hook >= 1:0.1.4-9 for package: 2:docker-common-1.13.1-209.git7d71120.el7.centos.x86_64
--> Processing Dependency: oci-registry-machine >= 1:0-5.13 for package: 2:docker-common-1.13.1-209.git7d71120.el7.centos.x86_64
--> Processing Dependency: lvm2 >= 2.02.112 for package: 2:docker-common-1.13.1-209.git7d71120.el7.centos.x86_64
```

2. 安装后启动docker服务

systemctl start docker

```

Installed:
docker.x86_64 2:1.13.1-209.git7d71120.el7.centos

Dependency Installed:
PyYAML.x86_64 0:3.10-11.el7
checkpolicy.x86_64 0:2.5-8.el7
containers-common.x86_64 1:0.1.40-11.el7_8
device-mapper-persistent-data.x86_64 0:0.8.5-3.el7_9.2
fuse-overlayfs.x86_64 0:0.7.2-6.el7_8
libcgroup.x86_64 0:0.41-21.el7
libxml2-python.x86_64 0:2.9.1-6.el7_9.6
lvm2-libs.x86_64 7:2.02.187-6.el7_9.5
oci-umount.x86_64 2:2.5-3.el7
python-backports.x86_64 0:1.0-8.el7
python-dateutil.x86_64 0:3.12.2-4.el7
python-ipaddress.noarch 0:1.0.16-2.el7
python-six.noarch 0:1.9.0-2.el7
slirp4netns.x86_64 0:0.4.3-4.el7_8
subscription-manager-rhsm-certificates.x86_64 0:1.24.51-1.el7.centos

Dependency Updated:
audit.x86_64 0:2.8.5-4.el7 audit-libs.x86_64 0:2.8.5-4.el7 device-mapper.x86_64 7:1.02.170-6.el7_9.5 device-mapper-libs.x86_64 7:1.02.170-6.el7_9.5 libxml2.x86_64 0:2.9.1-6.el7_9.6
policycoreutils.x86_64 0:2.5-34.el7

Complete!
[root@lntest122127 ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: http://docs.docker.com

[root@lntest122127 ~]# systemctl start docker
[root@lntest122127 ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2023-05-26 10:06:49 CST; 2s ago
     Docs: http://docs.docker.com
   Main PID: 15039 (dockerd-current)
   CGroup: /system.slice/docker.service
           └─15039 /usr/bin/dockerd-current --add-runtime docker=/usr/libexec/docker/docker-runc-current --default-runtime=docker-runc --exec-opt native.cgroupdriver=systemd --userland-p...
           └─15050 /usr/bin/docker-containerd-current -l unix:///var/run/docker/libcontainerd/docker-containerd.sock --metrics-interval=0 --start-timeout=2m --state-dir /var/run/docker/libcon...

May 26 10:06:48 lntest122127 dockerd-current[15039]: time="2023-05-26T10:06:48.553066124+08:00" level=info msg="libcontainerd: new containerd process, pid: 15050"
May 26 10:06:49 lntest122127 dockerd-current[15039]: time="2023-05-26T10:06:49.69515526+08:00" level=info msg="Graph migration to content-addressability took 0.00 seconds"
May 26 10:06:49 lntest122127 dockerd-current[15039]: time="2023-05-26T10:06:49.696951642+08:00" level=info msg="Loading containers: start."
May 26 10:06:49 lntest122127 dockerd-current[15039]: time="2023-05-26T10:06:49.763474830+08:00" level=info msg="FirewallD running: false"
May 26 10:06:49 lntest122127 dockerd-current[15039]: time="2023-05-26T10:06:49.837020386+08:00" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/... IP address"
May 26 10:06:49 lntest122127 dockerd-current[15039]: time="2023-05-26T10:06:49.876695738+08:00" level=info msg="Loading containers: done."
May 26 10:06:49 lntest122127 dockerd-current[15039]: time="2023-05-26T10:06:49.897148302+08:00" level=info msg="Daemon has completed initialization"
May 26 10:06:49 lntest122127 dockerd-current[15039]: time="2023-05-26T10:06:49.897175115+08:00" level=info msg="Daemon commit="7d71120/1.13.1" graphdriver=overlay2 verstons=1.13.1
May 26 10:06:49 lntest122127 dockerd-current[15039]: time="2023-05-26T10:06:49.902918014+08:00" level=info msg="API listen on /var/run/docker.sock"
May 26 10:06:49 lntest122127 systemd[1]: Started Docker Application Container Engine.
Hint: Some lines were ellipsized, use -l to show in full.
    
```

2.3. 安装TDH社区开发版

1. 将从官网下载下来的产品包解压后上传至安装环境

产品包名称: TDH-Standalone-Community-Transwarp-9.3.1-X86_64-final.tar.gz

2. 执行下述命令进行解压, 解压后将出现一个镜像tar包

tar -zxvf TDH-Standalone-Community-Transwarp-9.3.1-X86_64-final.tar.gz

```

[root@lntest122127 ~]# ls
anaconda-ks.cfg  nohup.out  TDH-Standalone-Community-Transwarp-9.3.1-X86_64-final.tar.gz
[root@lntest122127 ~]# tar -zxvf TDH-Standalone-Community-Transwarp-9.3.1-X86_64-final.tar.gz

[root@lntest122127 ~]#
[root@lntest122127 ~]# ls
anaconda-ks.cfg  nohup.out  tdh-standalone-2023.5.tar  TDH-Standalone-Community-Transwarp-9.3.1-X86_64-final.tar.gz
[root@lntest122127 ~]#
    
```

3. 执行下述命令加载镜像

docker load -i tdh-standalone-2023.5.tar

```

[root@lntest122127 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED            SIZE
[root@lntest122127 ~]# docker load -i tdh-standalone-2023.5.tar
74ddd0ec08fa: Loading layer [=====>] 238.6 MB/238.6 MB
3bc9d29c351b: Loading layer [=====>] 7.221 GB/7.221 GB
Loaded image: tdh-standalone:2023.5
[root@lntest122127 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED            SIZE
tdh-standalone      2023.5             911f087abafe       13 hours ago      7.43 GB
    
```

4. 执行下方指令启动容器并运行镜像, 运行格式为:

docker run -d --network host -v <本地目录路径>:/opt/transwarp --privileged tdh-standalone:2023.5

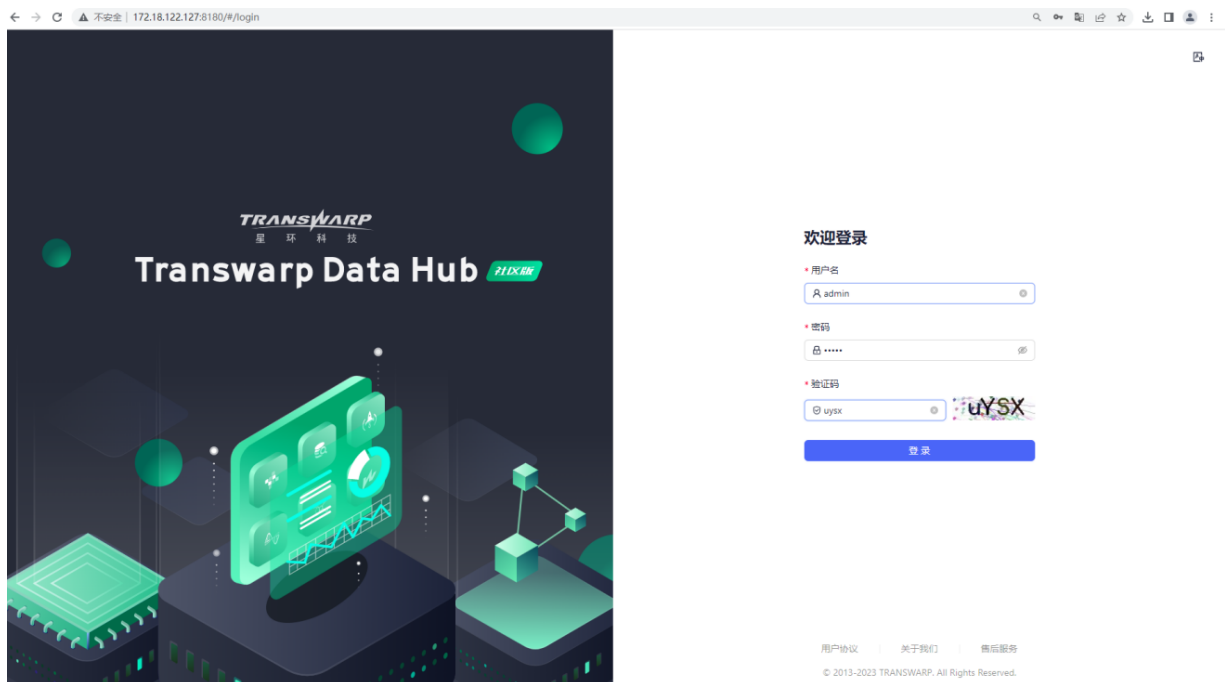
```
[root@Intest122127 tdh]# docker run -d --network host -v /home/tdh:/opt/transwarp --privileged tdh-standalone:2023.5
cd794f88adec0608f147a253eaea7d0925ccdbd05ddafe40c2c965e56960f15f
[root@Intest122127 tdh]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
cd794f88adec        tdh-standalone:2023.5  "/usr/bin/boot.sh"  26 seconds ago    Up 25 seconds      8180               ecstatic_beaver
```



-v参数配置了TDH挂载的本地磁盘路径。该路径下会保存TDH运行过程中产生的配置conf、数据data、日志log。再次提醒请不要随意改动并做好备份。

- 容器启动后需等待30s至2分钟
- 浏览器访问管理节点8180端口

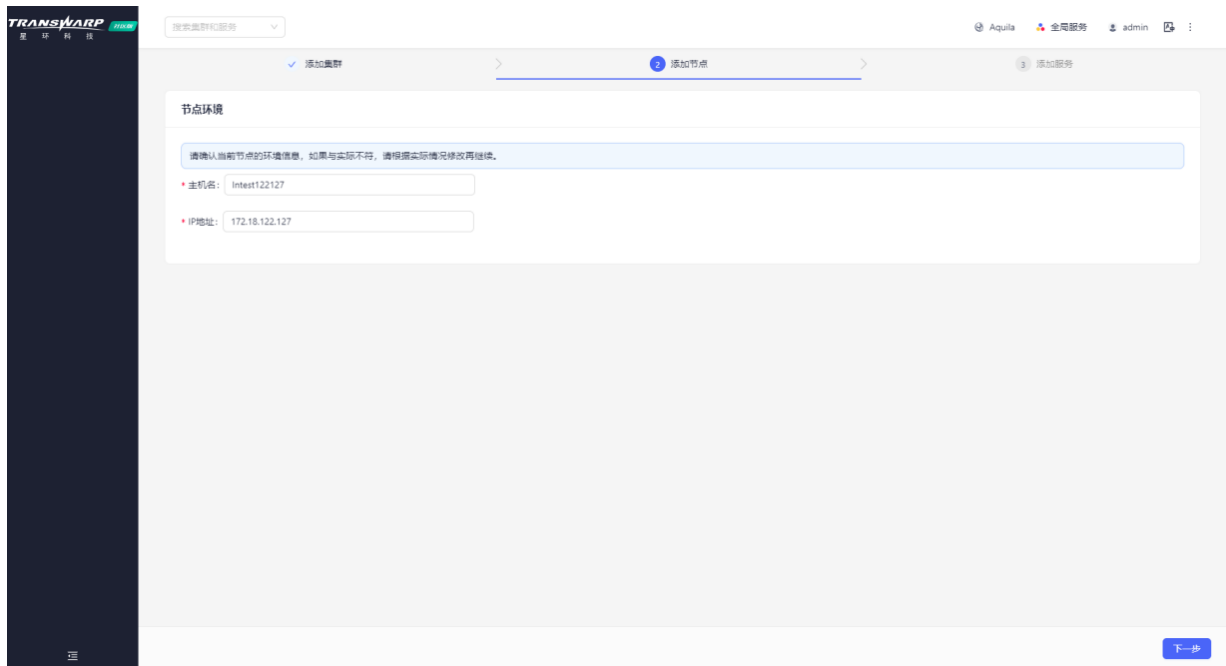
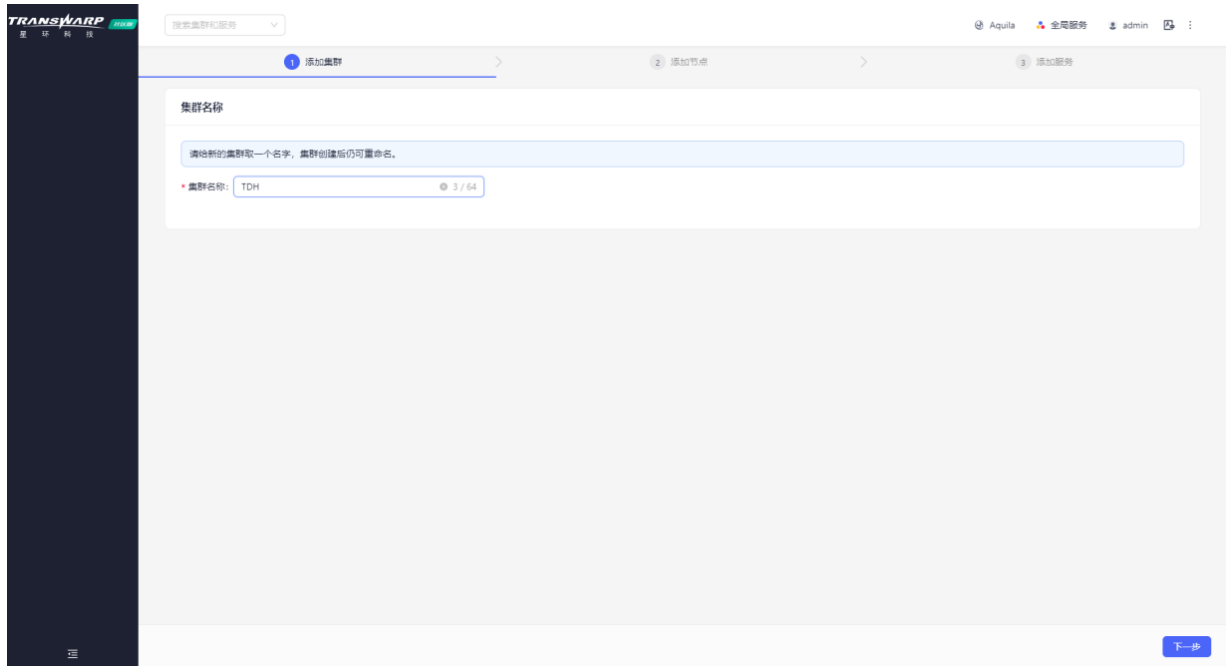
打开客户端浏览器（推荐使用Google Chrome浏览器），访问http://host:8180，比如http://172.16.3.108:8180/。访问这个地址，您会看到下面的登录页面。初次登录以admin的身份登录，密码也是admin。

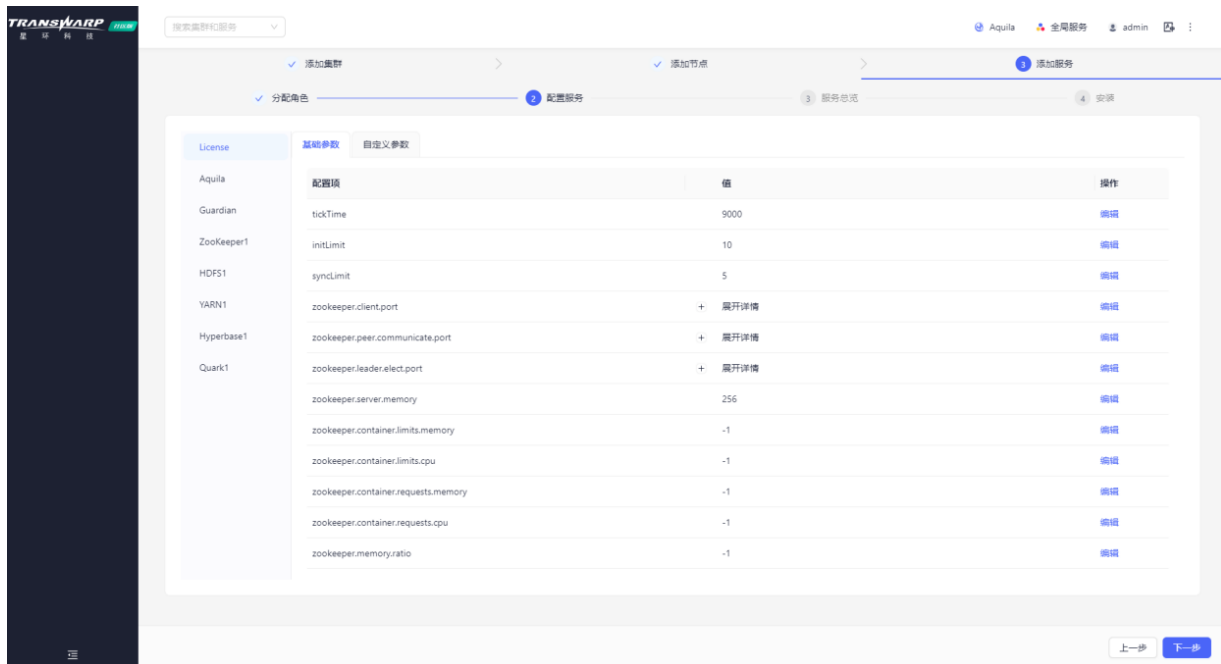
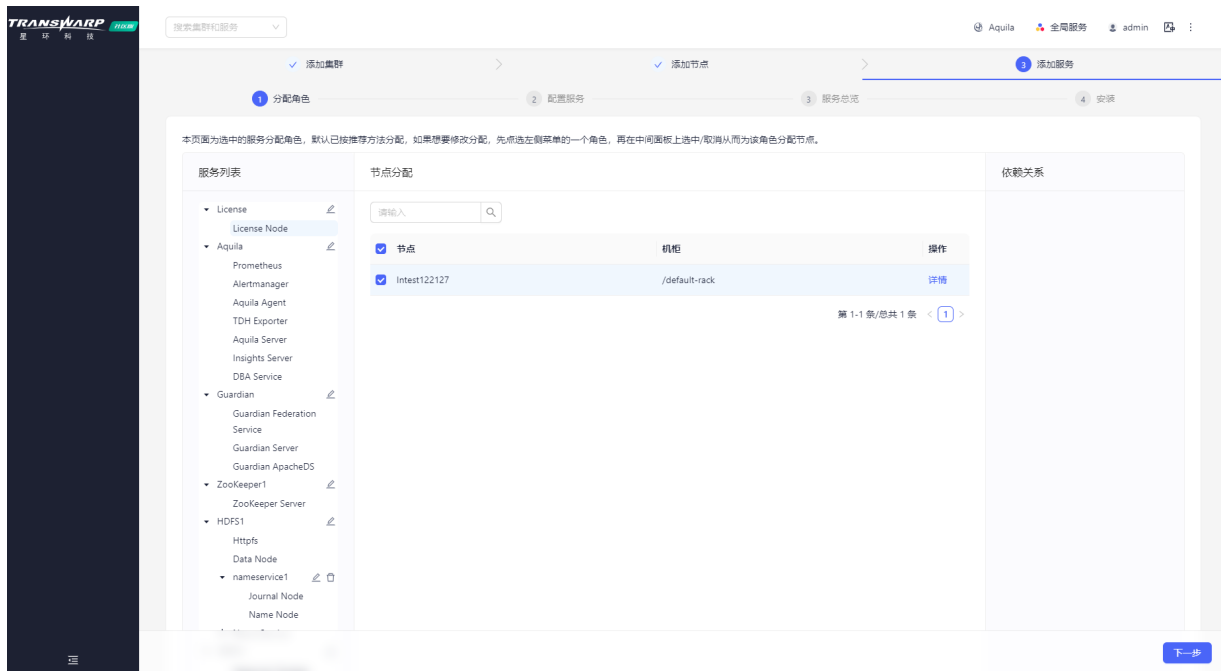


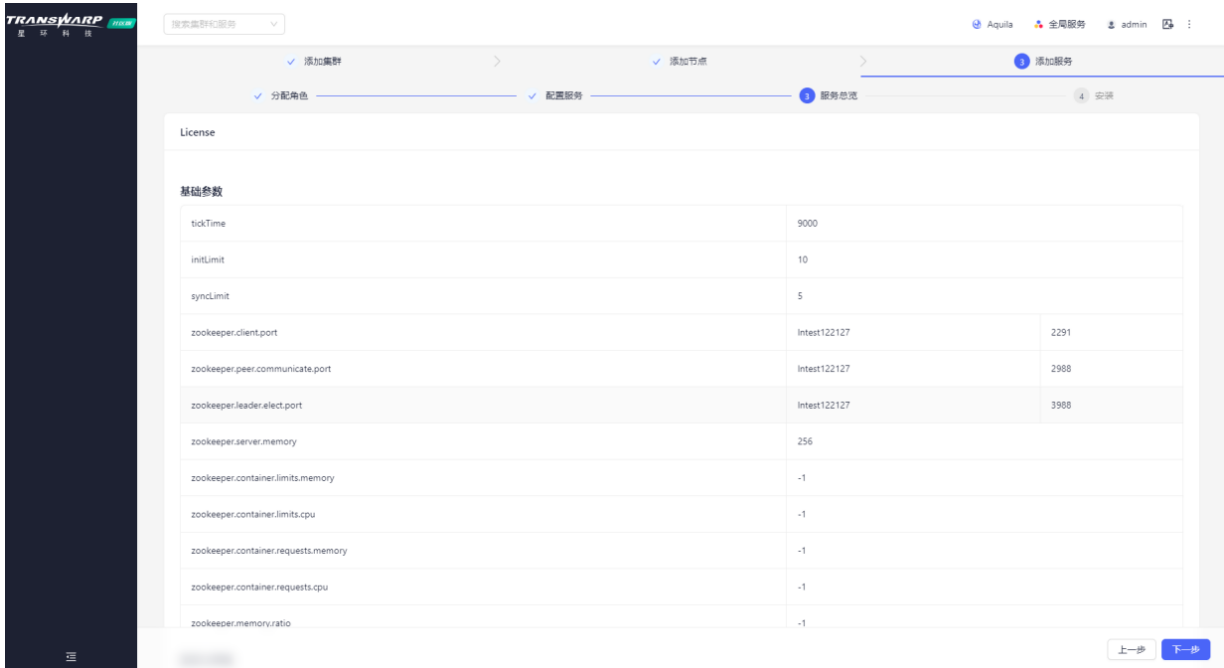
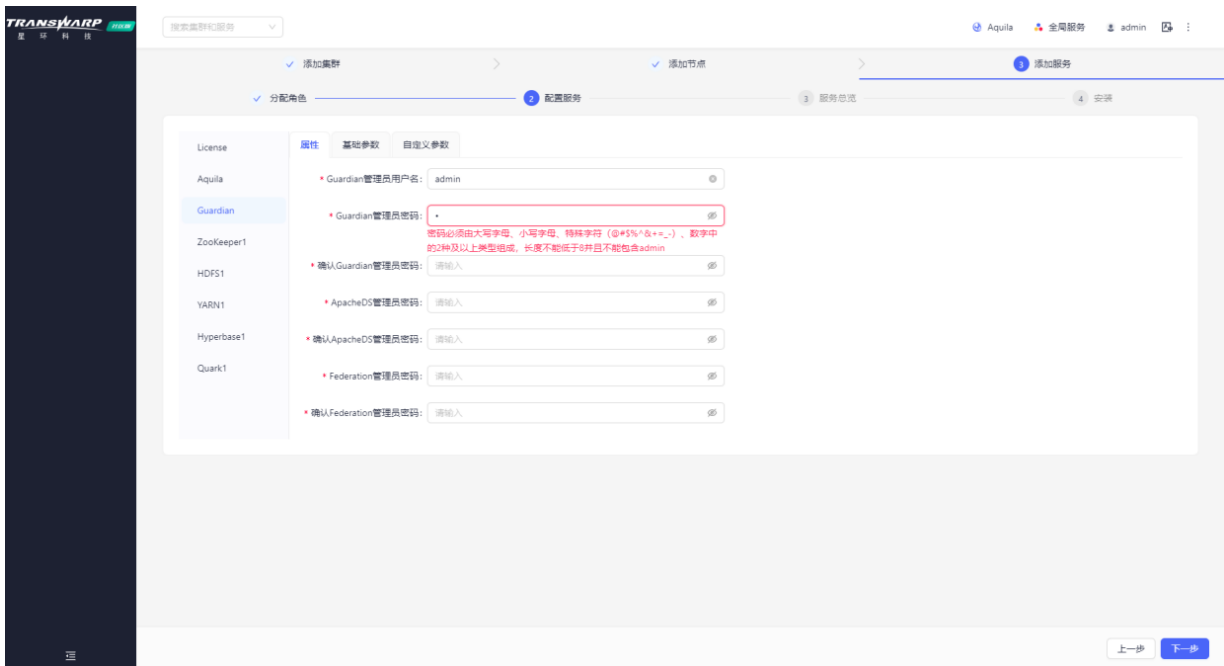
- 登录后，您需要先接受最终用户协议才可以进行进一步操作

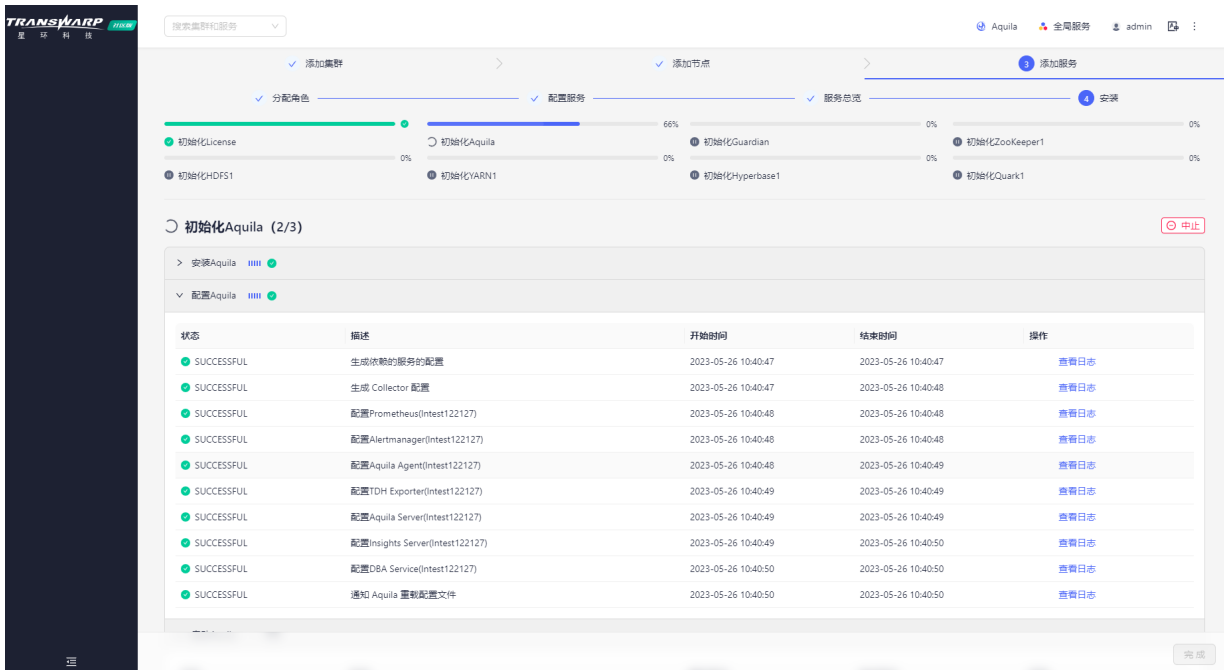
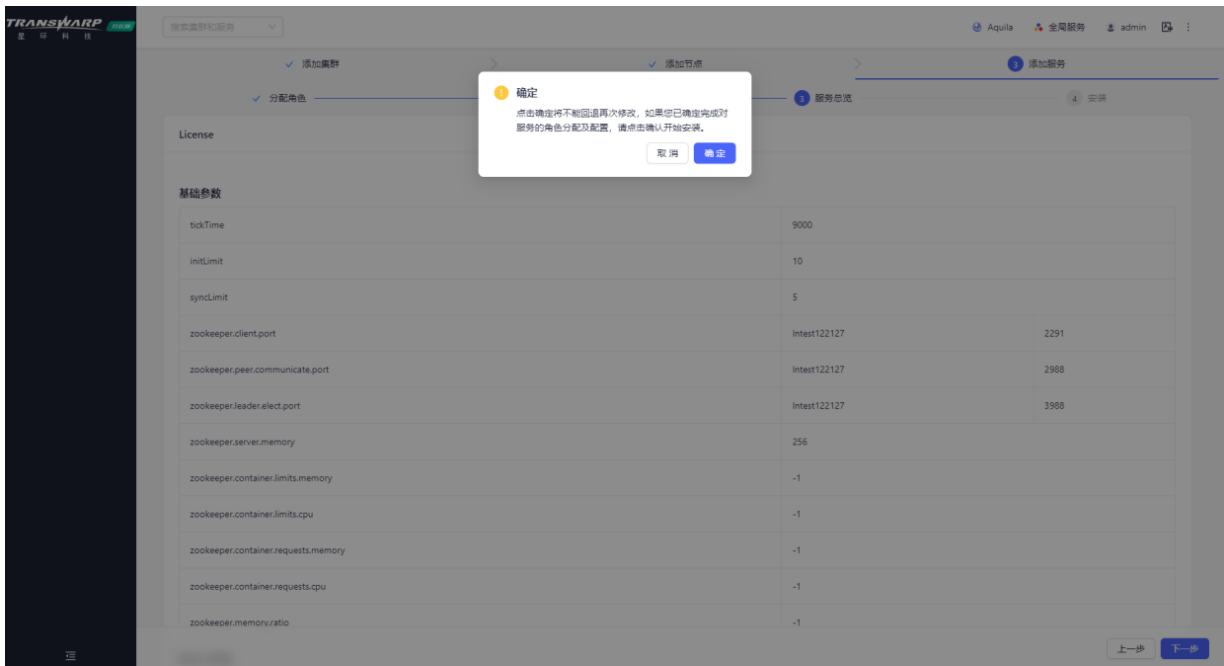


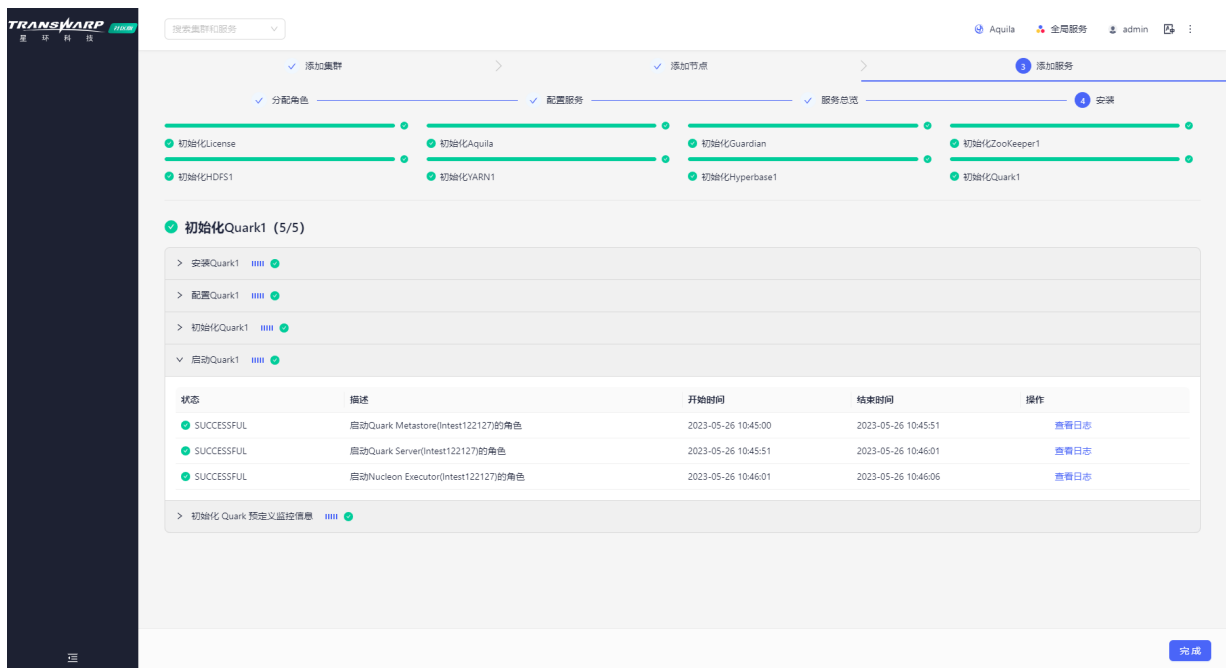
8. 按照弹出窗口默认配置执行安装步骤











2.4. 自助申请许可证

1. 访问星环云页面创建组织

星环云网站: <https://cloud.transwarp.cn/home>





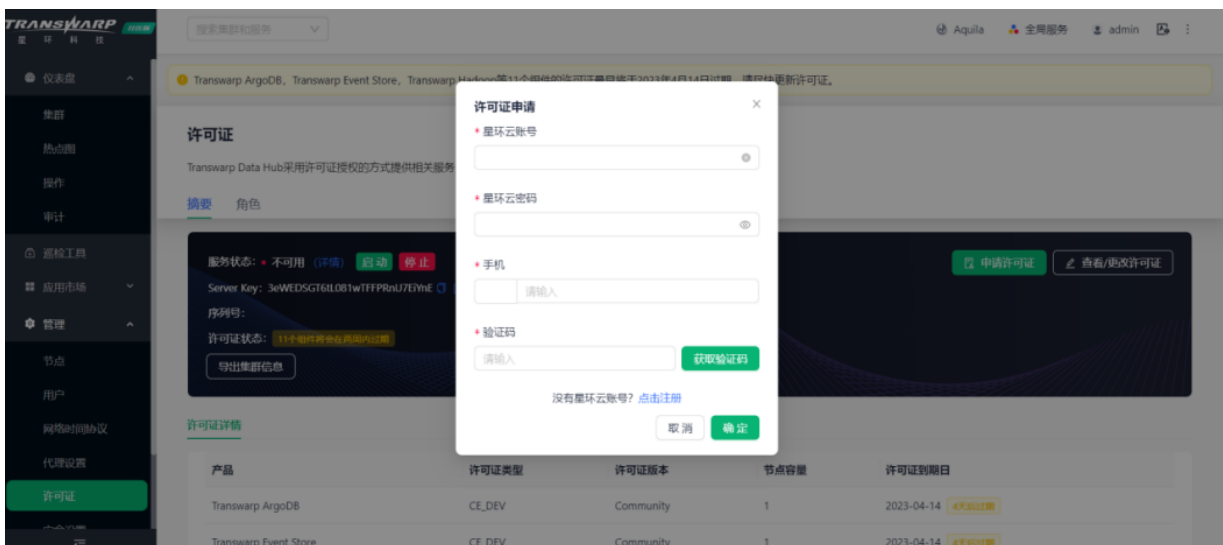
2. 编辑组织信息

创建好后如果需要重新更改名词可以选择编辑



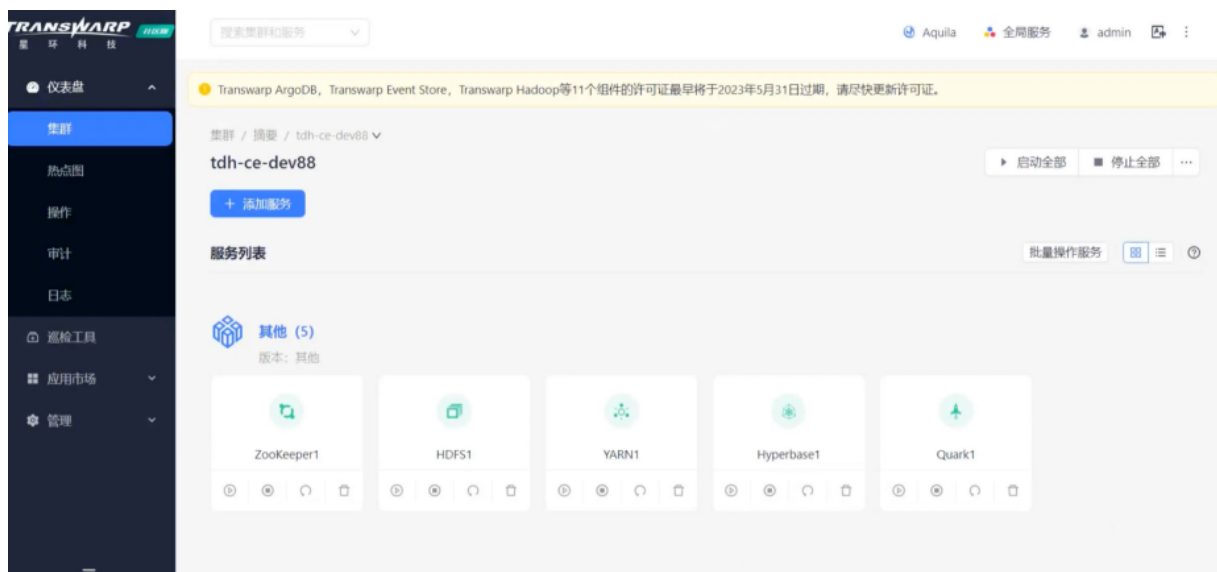
3. 申请许可证

回到TDH Manager，输入账号进行“许可证申请”。



4. 完成申请

需要注意的是，社区开发版产品包已包含当前提供的全部服务，因此不支持添加服务的操作，点击“添加服务”后报错500，请忽略。

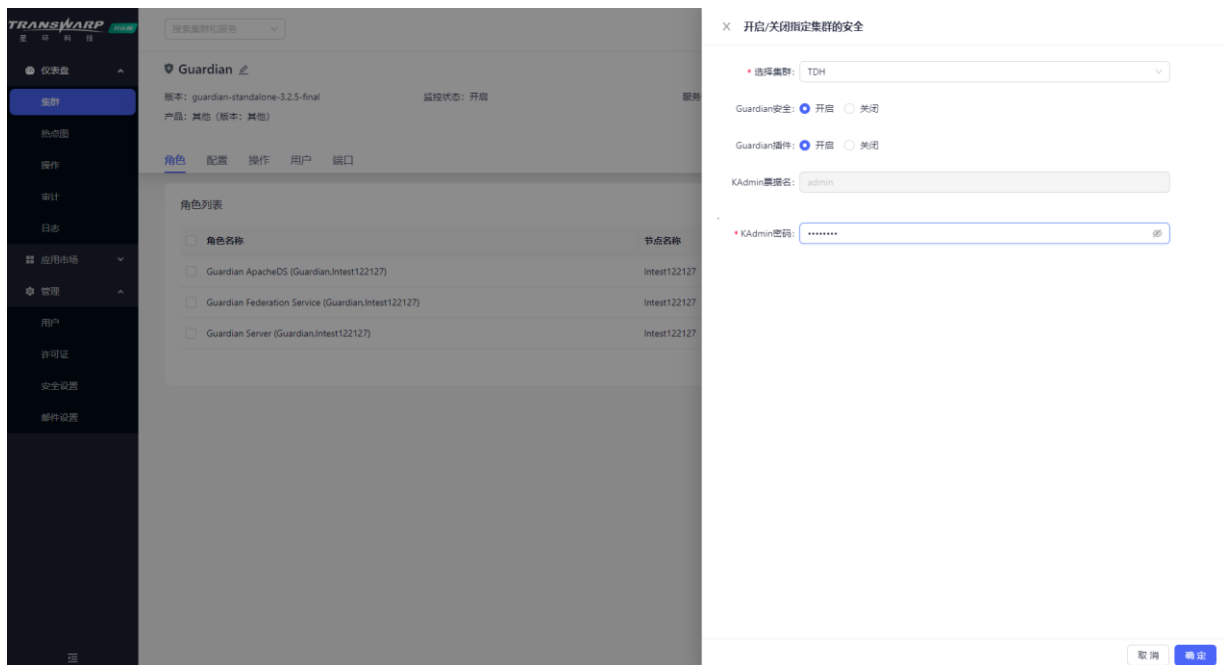
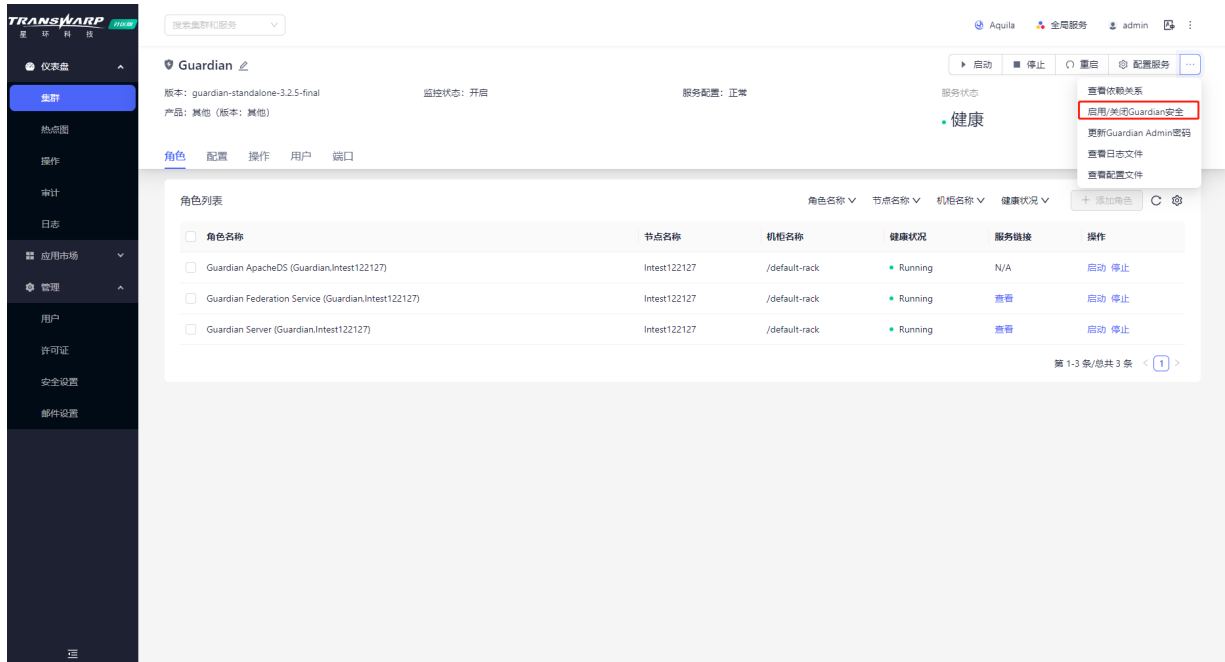


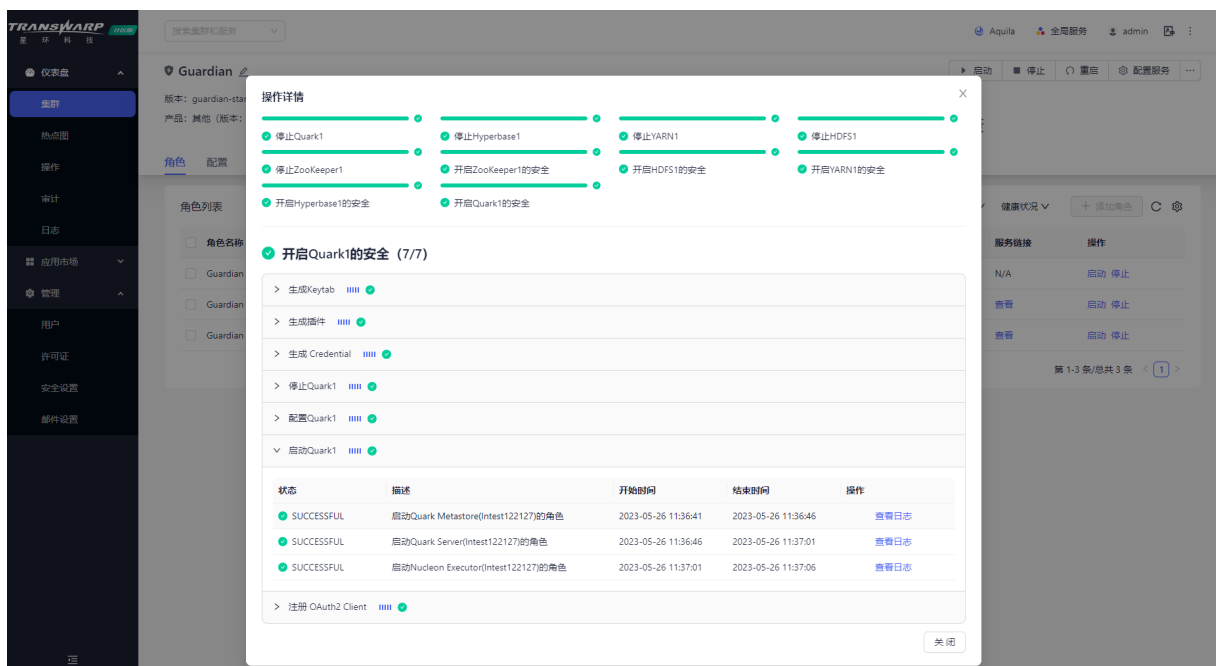
TDH社区开发版安装正式完成

3. 其他使用方面的操作

3.1. 开启安全

1. 进入全局服务下的Guardian服务详情页, 选择一键开启安全



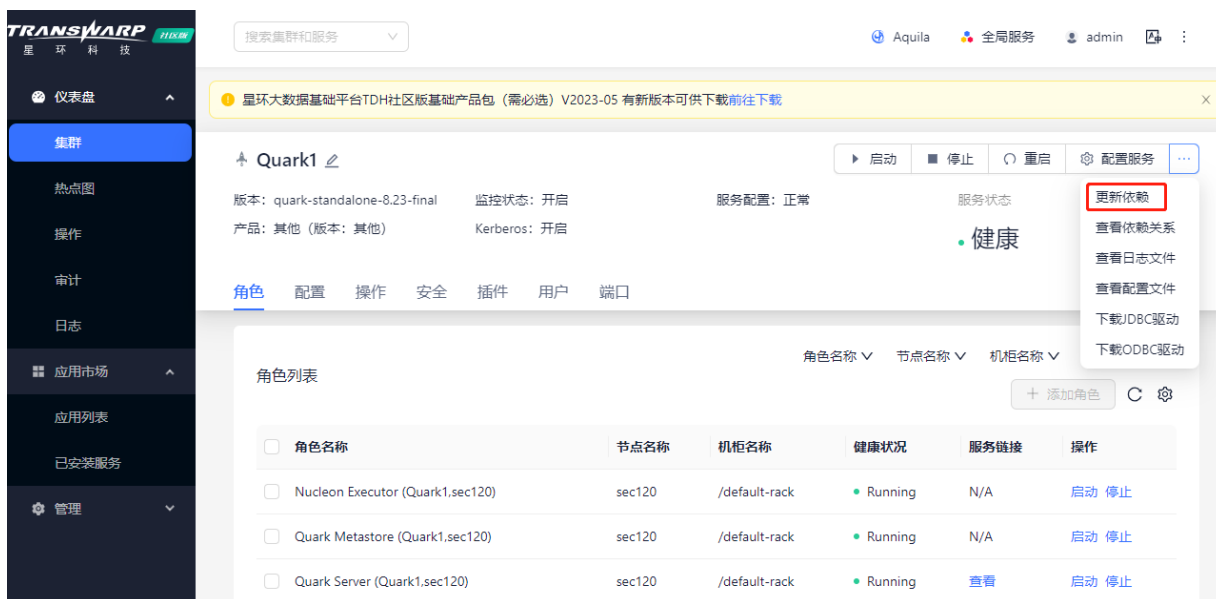


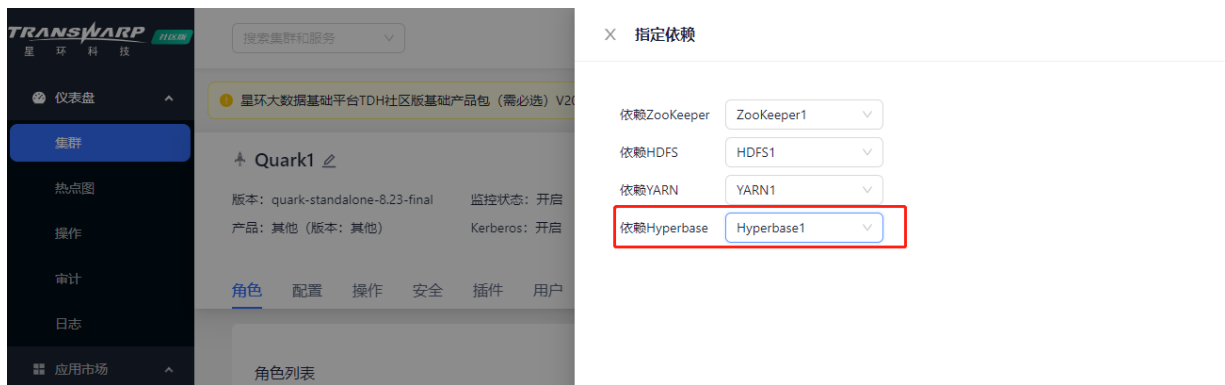
3.2. 使用核心组件Hyperbase的操作设置



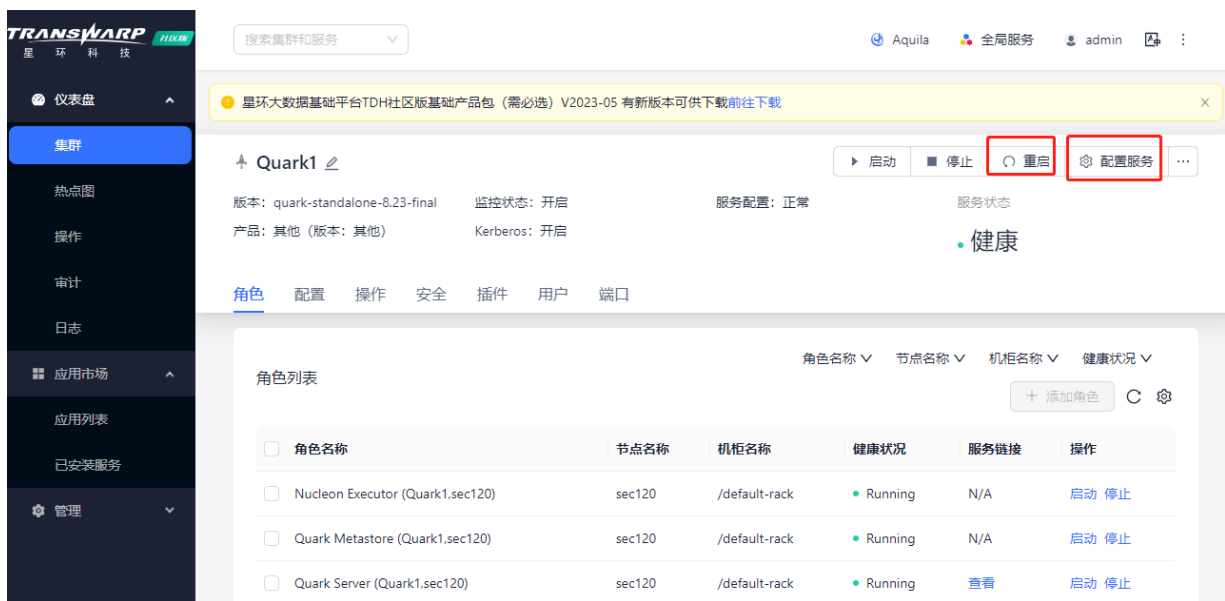
使用hyperbase前需要进行依赖配置，如果开启安全还需要进行赋权操作。

1. 安装结束后进入Quark服务详情页，更新quark依赖关系；



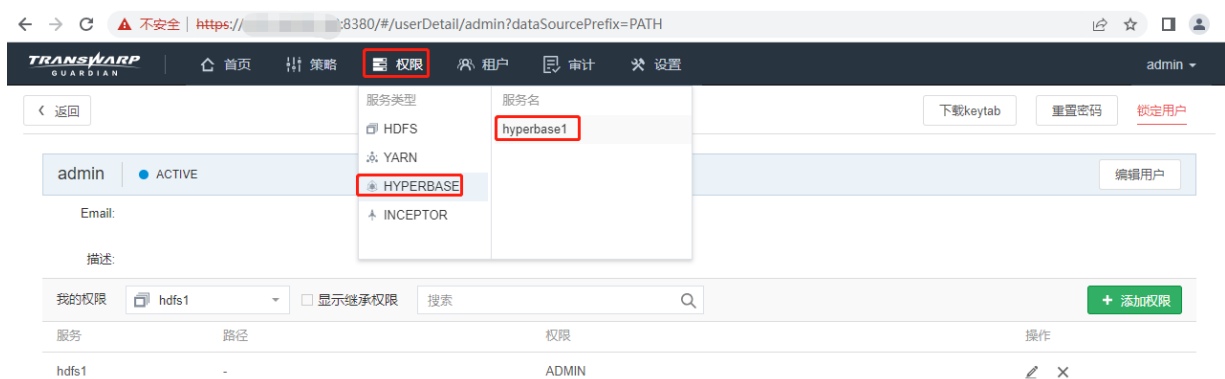


2. 点击配置服务，配置完成后进行重启服务；

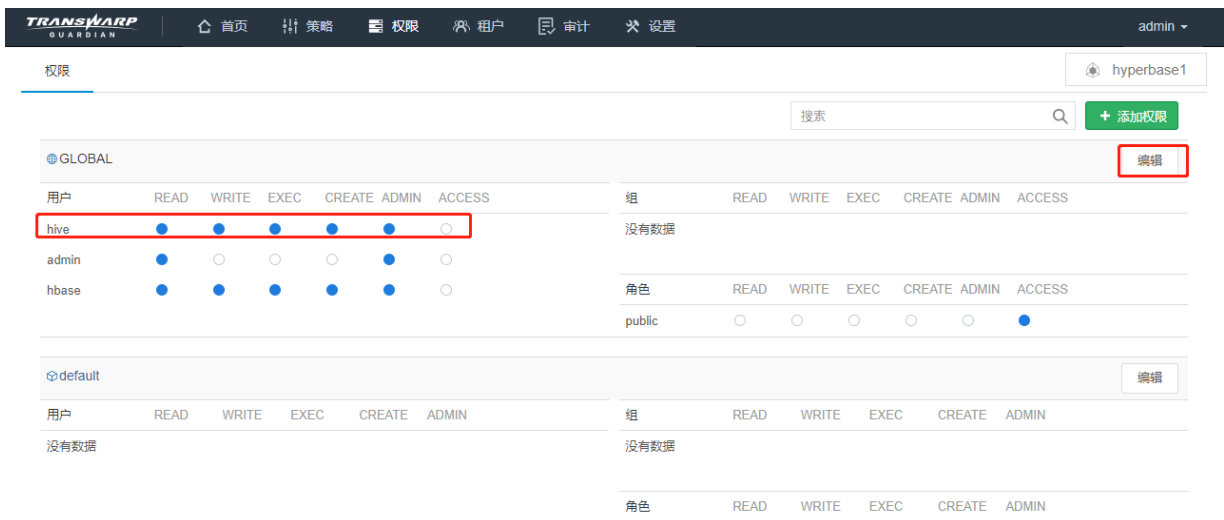


如果开启安全，需要进入guardian server页面，为hive用户赋予hyperbase权限。

1. 点击Guardian服务详情页，进入Guardian Server权限页面进行编辑；



2. 编辑后保存；



3.3. 容器重启后的操作

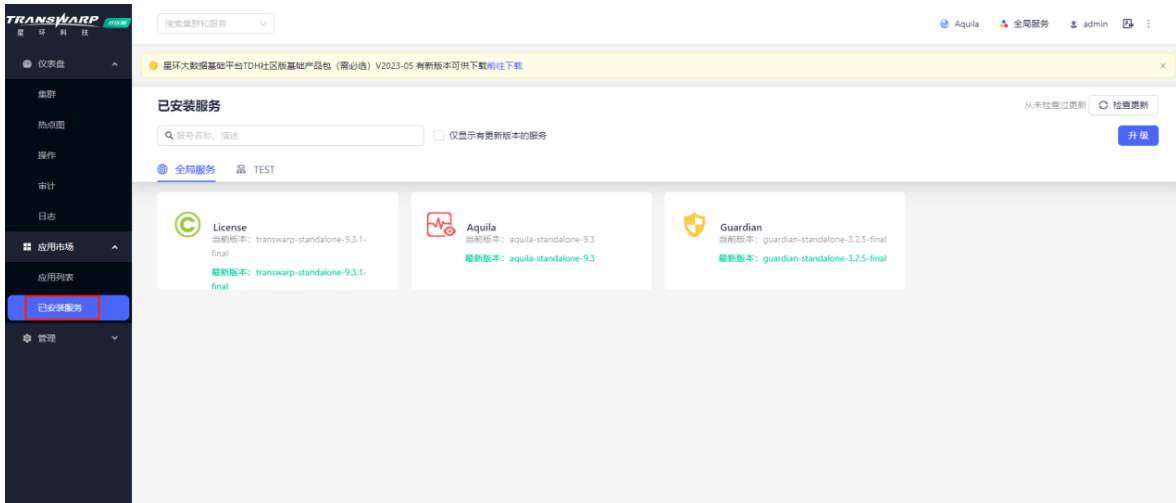


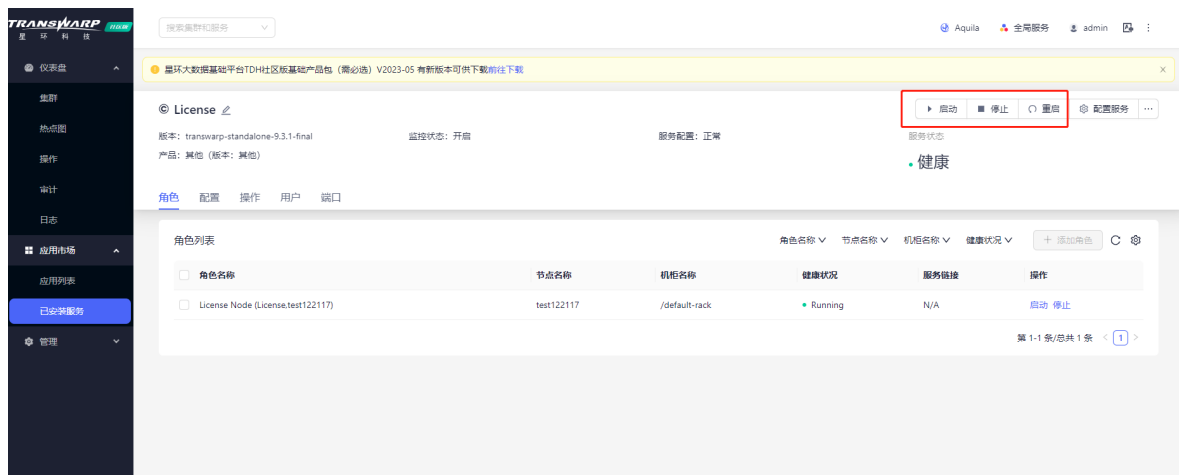
在Docker容器重启（宿主机掉电、更换TDH版本等）后，需要在manager上对TDH各服务进行重启操作。

```
[root@test122117 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
ac4b2966df18      tdh-standalone:2023.5  "/usr/bin/boot.sh"  22 minutes ago     Up 22 minutes     0.0.0.0:54321->0.0.0.0:54321  stupefied_darwin
[root@test122117 ~]# docker restart ac4b2966df18
```

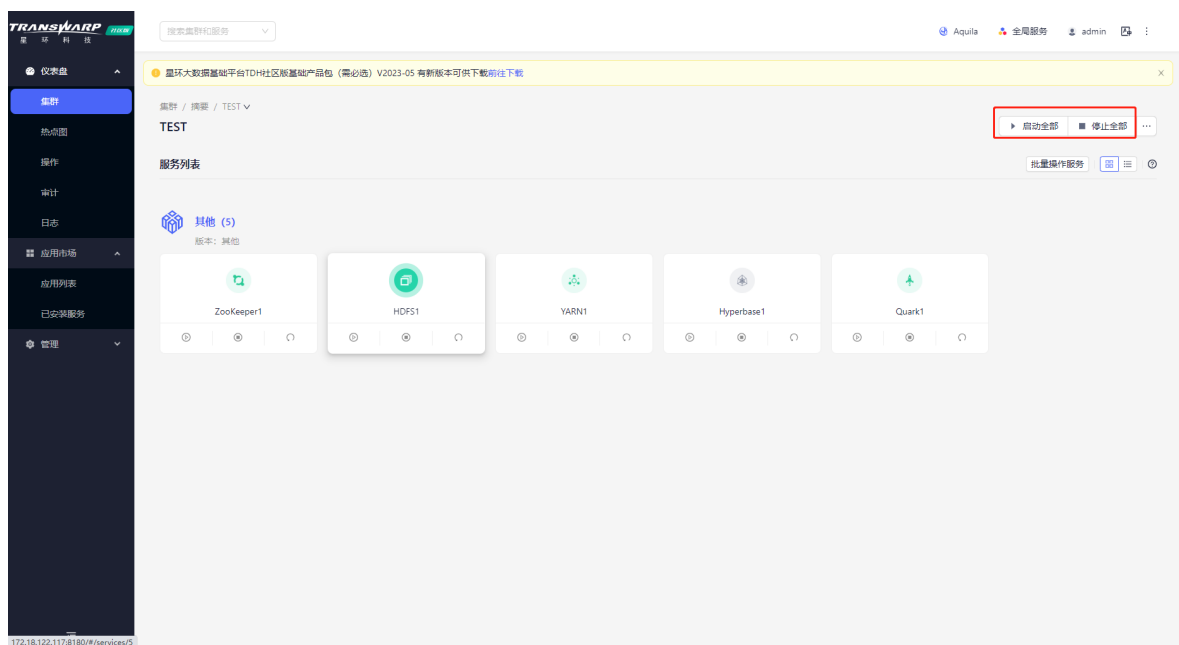
- TDH各服务的重启顺序为：

首先在“已安装服务”页面中，挨个重启License node、Aquila(若使用)、Guardian(开安全时) 服务。





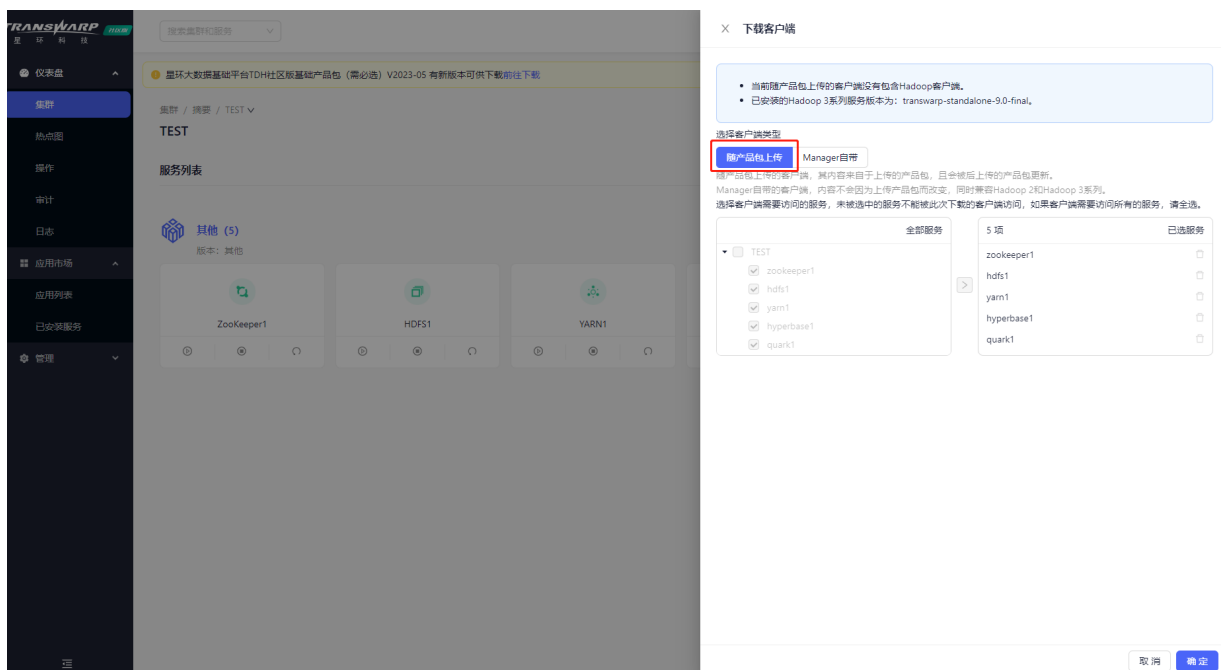
重启完成后返回集群首页，选择一键停止全部服务，再一键启动全部



3.4. 快速下载TDH-Client

Manager为用户提供了两种下载tdh-client的方式，其中选择manager自带tdh-client下载可能会比较慢，本章节将介绍如何进行快速下载。

1. Manager页面下载客户端，选择“随产品包上传”下载tdh-client。



2. 将下载后的tdh-client.tar上传至服务器，并解压。

```
tar -xvf tdh-client.tar
```

3. 拷贝容器内的bundled_tdh_client目录到本地。此处假设客户端解压到了目录/root下，命令为：

```
docker cp 830805a9f95a:/var/lib/transwarp-manager/master/content/resources/bundled_tdh_client /root
```

4. 复制bundled_tdh_client目录下的内容到解压后的TDH-Client目录下。

```
cp -nr /root/bundled_tdh_client/* /root/TDH-Client
```

5. 完成后即可使用TDH-Client。

6. 使用TDH-Client前还需设置JAVA_HOME环境变量，设置示例：

a. 找到java安装路径

```
[root@sec120 ~]# which java
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.362.b08-1.el7_9.x86_64/jre/bin/java
[root@sec120 ~]# ll /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.362.b08-1.el7_9.x86_64/jre/bin/java
-rwxr-xr-x. 1 root root 8984 Jan 26 01:03 /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.362.b08-1.el7_9.x86_64/jre/bin/java
[root@sec120 ~]#
```

b. 修改配置文件/etc/profile，在该文件中添加环境变量

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.362.b08-1.el7_9.x86_64/jre
export PATH=$JAVA_HOME/bin:$PATH
```

c. 输入指令使变量生效

```
source /etc/profile
```



注意：TDH切换安全开关后均需要重新下载tdh-client.tar。

3.5. 集群外使用hadoop client

首先您需要保证JAVA_HOME设置正确。已准备好TDH-Client（TDH-Client文件夹下有init.sh），下方假定TDH-Client存储路径为：/root

1. 使用zookeeper-client

在guardian上下载zookeeper租户的keytab，假定路径是/tmp/zookeeper.keytab，然分别执行

```
# export ZOOKEEPER_KEYTAB=/tmp/zookeeper.keytab
# source /root/TDH-Client/init.sh
```

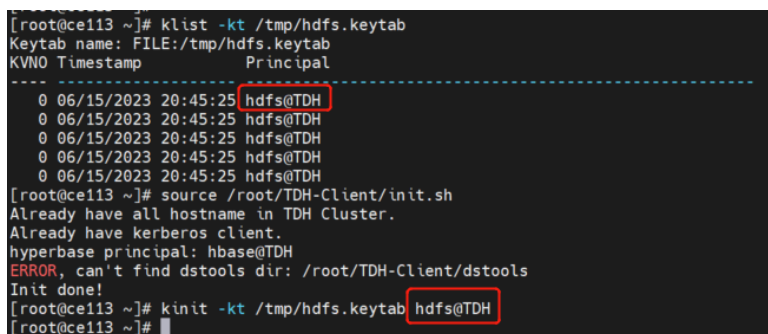
2. 使用hadoop/hdfs

在guardian上下载hdfs租户的keytab，假定路径是/tmp/hdfs.keytab，使用下方命令确定Keytab文件下的principal：

```
# klist -kt /tmp/hdfs.keytab
```

确认后执行下方命令方可使用。

```
# source /root/TDH-Client/init.sh
# kinit -kt /tmp/hdfs.keytab <Principal>@TDH
```



```
[root@ce113 ~]# klist -kt /tmp/hdfs.keytab
Keytab name: FILE:/tmp/hdfs.keytab
KVNO Timestamp Principal
-----
0 06/15/2023 20:45:25 hdfs@TDH
0 06/15/2023 20:45:25 hdfs@TDH
0 06/15/2023 20:45:25 hdfs@TDH
0 06/15/2023 20:45:25 hdfs@TDH
0 06/15/2023 20:45:25 hdfs@TDH
[root@ce113 ~]# source /root/TDH-Client/init.sh
Already have all hostname in TDH Cluster.
Already have kerberos client.
hyperbase principal: hbase@TDH
ERROR, can't find dstools dir: /root/TDH-Client/dstools
Init done!
[root@ce113 ~]# kinit -kt /tmp/hdfs.keytab hdfs@TDH
[root@ce113 ~]#
```

3. 使用hbase shell

在guardian上下载hbase租户的keytab，假定路径是/tmp/hbase.keytab，使用下方命令确定Keytab文件下的principal：

```
# klist -kt /tmp/hbase.keytab
```

确认后执行下方命令方可使用。

```
# export HYPERBASE_KEYTAB=/tmp/hbase.keytab
# source /root/TDH-Client/init.sh
# kinit -kt /tmp/hbase.keytab <Principal>@TDH
```

```
[root@ce113 ~]# klist -kt /tmp/hbase.keytab
Keytab name: FILE:/tmp/hbase.keytab
KVNO Timestamp Principal
-----
0 06/15/2023 21:37:30 hbase@TDH
0 06/15/2023 21:37:30 hbase@TDH
0 06/15/2023 21:37:30 hbase@TDH
0 06/15/2023 21:37:30 hbase@TDH
0 06/15/2023 21:37:30 hbase@TDH
[root@ce113 ~]# export HYPERBASE_KEYTAB=/tmp/hbase.keytab
[root@ce113 ~]# source /root/TDH-Client/init.sh
Already have all hostname in TDH Cluster.
Already have kerberos client.
hyperbase principal: hbase@TDH
ERROR, can't find dstools dir: /root/TDH-Client/dstools
Init done!
[root@ce113 ~]# kinit -kt /tmp/hbase.keytab hbase@TDH
[root@ce113 ~]#
```

3.6. 如何查看资源占用

- 查看容器占用资源量

```
docker stats ${containerId}
```

```
[root@test122117 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS               NAMES
84cff9d42ccf      tdh-standalone:2023.5  "/usr/bin/boot.sh"    5 hours ago       Up 5 hours         0 B / 0 B          flamboyant_edison
[root@test122117 ~]# docker stats 84cff9d42ccf
CONTAINER ID        CPU %               MEM USAGE / LIMIT   MEM %               NET I/O             BLOCK I/O           PIDS
84cff9d42ccf      7.10%              5.815 GiB / 11.58 GiB  50.23%             0 B / 0 B           14 GB / 5.07 GB    1346
CONTAINER ID        CPU %               MEM USAGE / LIMIT   MEM %               NET I/O             BLOCK I/O           PIDS
84cff9d42ccf      53.42%             5.817 GiB / 11.58 GiB  50.25%             0 B / 0 B           14 GB / 5.07 GB    1364
CONTAINER ID        CPU %               MEM USAGE / LIMIT   MEM %               NET I/O             BLOCK I/O           PIDS
84cff9d42ccf      53.42%             5.817 GiB / 11.58 GiB  50.25%             0 B / 0 B           14 GB / 5.07 GB    1364
CONTAINER ID        CPU %               MEM USAGE / LIMIT   MEM %               NET I/O             BLOCK I/O           PIDS
84cff9d42ccf      293.73%            5.815 GiB / 11.58 GiB  50.24%             0 B / 0 B           14 GB / 5.07 GB    1349
^C
[root@test122117 ~]#
```

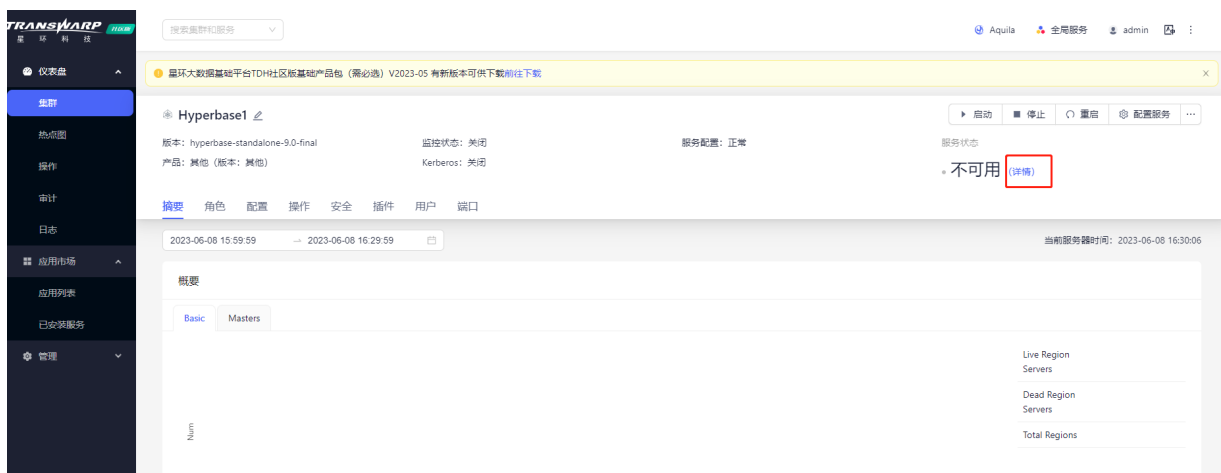
4. 常见的问题排查方法

- 对服务安装/重启等操作出现异常时，查看出错服务组件组件的日志即可。

操作步骤中可查看日志：



- 服务状态不正常时，可查看服务状态详情：



- 后台进入到挂载的本地磁盘下的log目录，其下有TDH各组件的日志：

```
[root@test122117 log]# pwd
/home/tdh/log
[root@test122117 log]# ls -al
total 4
drwxr-xr-x. 11 root root 152 Jun  8 10:14 .
drwxr-xr-x.  5 root root  41 Jun  8 10:14 ..
drwxr-xr-x.  8 root root 104 Jun  8 10:24 aquila
drwxr-xr-x.  2 root root 155 Jun  8 10:20 guardian
drwxr-xr-x.  2 1004 1004 123 Jun  8 10:21 hdfs1
drwxr-xr-x.  2 1006 1006 167 Jun  8 10:22 hyperbase1
drwxr-xr-x.  2 1000 1000  27 Jun  8 10:17 license
drwxr-xr-x.  3 1002 1002 4096 Jun  8 16:17 quark1
drwxr-xr-x.  4 root root  33 Jun  8 10:14 transwarp-manager
drwxr-xr-x.  2 1001 1001 199 Jun  8 10:21 yarn1
drwxr-xr-x.  2 1000 1000  27 Jun  8 10:20 zookeeper1
[root@test122117 log]#
```

- 数据库的操作出现错误时，可根据返回的错误日志信息判断是使用问题还是产品问题。
若为TDH产品问题，则查看quark的日志或有异常的组件日志。

客户服务

技术支持

感谢你使用星环信息科技（上海）股份有限公司的产品和服务。如您在产品使用或服务中有任何技术问题，可以通过以下途径找到我们的技术人员给予解答。

Email: support@transwarp.io

技术支持热线电话: 4008 079 976

技术支持QQ专线: 3221723229, 3344341586

官方网址: <http://www.transwarp.cn/>

社区支持: <http://community.transwarp.cn/>

意见反馈

如果你在系统安装，配置和使用中发现任何产品问题，可以通过以下方式反馈：

Email: support@transwarp.io

感谢你的支持和反馈，我们一直在努力！