

中心服务器相关说明

安装

服务器包说明

以 `master_1.0.0-20200817-15-10-46.tar` 为例，`1.0.0` 为服务器版本号，`20200817-15-10-46` 为发部时间。

安装准备

- https 证书
- 操作系统支持中文
- 开放端口https (443/tcp) 、http端口 (80/tcp) ，媒体中心端口 (6667/tcp) ，masterrpc端口 (6669/tcp)

默认安装步骤

默认安装会安装本地mysql数据库、本地redis。

1. 将某一服务器包 `master_1.0.0-20200817-15-10-46.tar` 移动到需要安装的服务器上
2. 解压服务器包 `tar -xf master_1.0.0-20200817-15-10-46.tar`
3. 进入解压后的cluster目录 `cd cluster`
4. 默认安装，运行 `./deploy install`
5. 根据界面提示进行安装，安装程序将会对安装进行检测，如果不符合条件会有提示信息，比如端口占用，mysql已经存在，ip配置等信息，请根据提示进行调整
6. 等待安装完成，安装过程中的Oracle相关报错客户忽略

```
error: package oracle-instantclient19.10-basic-19.10.0.0.0-1.x86_64 is not installed
error: package oracle-instantclient19.10-devel-19.10.0.0.0-1.x86_64 is not installed
error: package oracle-instantclient19.10-sqlplus-19.10.0.0.0-1.x86_64 is not installed
error: package unixODBC-devel-2.3.1-14.e17.x86_64 is not installed
error: package unixODBC-2.3.1-14.e17.x86_64 is not installed
```

安装过程参考

```
The latest information about MariaDB is available at http://mariadb.org/.
You can find additional information about the MySQL part at:
http://dev.mysql.com
Support MariaDB development by buying support/new features from MariaDB
Corporation Ab. You can contact us about this at sales@mariadb.com.
Alternatively consider joining our community based development effort:
http://mariadb.com/kb/en/contributing-to-the-mariadb-project/

Starting MySQL. SUCCESS!
2021/10/20 08:50:40.685085 [Info] Install Mysql success.
2021/10/20 08:50:40.685175 [Info] Install deps.
2021/10/20 08:50:40.685257 [Info] Check install Oracle client deps now, please wait...
error: package oracle-instantclient19.10-basic-19.10.0.0-1.x86_64 is not installed
error: package oracle-instantclient19.10-devel-19.10.0.0-1.x86_64 is not installed
error: package oracle-instantclient19.10-sqlplus-19.10.0.0-1.x86_64 is not installed
error: package unixODBC-devel-2.3.1-14.el7.x86_64 is not installed
error: package unixODBC-2.3.1-14.el7.x86_64 is not installed
2021/10/20 08:50:50.314830 [Info] Check database connect.
2021/10/20 08:50:50.314993 [Debug] DB connect:root:pass123@tcp(127.0.0.1:3306)?charset=utf8&parseTime=True&loc=Local
2021/10/20 08:50:50.539919 [Info] Check db connect success.
2021/10/20 08:50:50.539972 [Info] Create database.
2021/10/20 08:50:50.542244 [Debug] Create database:Create database avd_cluster DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci
2021/10/20 08:50:50.558144 [Debug] Create database:Create database avd_mqtt DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci
2021/10/20 08:50:50.560952 [Debug] Create database:Create database xconf DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci
2021/10/20 08:50:50.563712 [Debug] Create database:Create database rtm_logic DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci
2021/10/20 08:50:50.564377 [Info] Create db success.
2021/10/20 08:50:50.564444 [Info] Copy files.
2021/10/20 08:51:07.170318 [Info] Copy dir success.
2021/10/20 08:51:07.170340 [Info] Reload config.
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Backend set to file
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Starting confd
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Backend source(s) set to conf/install.yaml
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/admin_api.json out of sync
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/admin_api.json has been updated
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/avdadmin.json out of sync
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/avdadmin.json has been updated
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/configsrv.json out of sync
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/configsrv.json has been updated
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO ./ctl.sh has md5sum 03cb691b699a537d91169b626be6e2a3 should be 2a5ab
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./ctl.sh out of sync
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./ctl.sh has been updated
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config etc/etcd.yml out of sync
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config etc/etcd.yml has been updated
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config etc/etcd.yml has been updated
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO ./nginx/conf/nginx.conf has md5sum 3a0b1f2d0a5734fe3200a48703bafed2
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./nginx/conf/nginx.conf out of sync
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./nginx/conf/nginx.conf has been updated
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/rtm_logic.json out of sync
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/rtm_logic.json has been updated
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/server.json out of sync
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./conf/server.json has been updated
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO ./etc/supervisord.conf has md5sum 1746f8331c638a5cf492c324f44ef7e4
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./etc/supervisord.conf out of sync
2021-10-20T08:51:08Z 1e7383ac63aa ./bin/confd[408]: INFO Target config ./etc/supervisord.conf has been updated
2021/10/20 08:51:08.641924 [Info] Reload config success.
2021/10/20 08:51:08.641967 [Info] Start service.
2021/10/20 08:51:08.746611 [Info] start service success.
[root@1e7383ac63aa cluster]#
[root@1e7383ac63aa cluster]#
```

参数化安装

安装的过程中，往往也需要根据实际情况，自定义安装参数。比如https端口、http端口、安装路径、连接外部数据库、连接外部redis等。进入解压后的cluster目录，通过 `./deploy install -h` 查看。因为参数较多，这里将通过具体的安装例子说明。

```
[root@1e7383ac63aa cluster]# ./deploy install -h
[install]
Usage:
    deploy install [command options] [arguments...]

Options:
  -admin_rpc_port string
                        avdadmin micro port (default "9050")
  -cluster_rpc_port string
                        Cluster Rpc port. (default "6669")
  -cluster_server_id string
                        master server register name. (default "3tee.avd.srv.avdcluster")
  -db_driver string
                        set db driver. (default "mysql")
  -etcd name string
                        etcd server name. (default "etcd01")
  -etcd_client_port string
                        etcd listen client port. (default "2380")
  -etcd_ip string
                        etcd listen ip. (default "127.0.0.1")
  -etcd_peer_port string
                        etcd listen peer port. (default "2379")
  -etcd_server string
```

```
    etcd servers with cluster supported. (default "127.0.0.1:2379")
-etcd_token string
    etcd cluster token.
-gateway_ip string
    micro api gateway port. (default "127.0.0.1")
-gateway_port string
    micro api gateway port. (default "9650")
-h show this help.
-help
    show this help.
-install_dir string
    cluster install dir. (default "/opt/avd_platform/")
-is_master string
    true:cluster server is master false: cluster server is salve (default
"true")
-log_level string
    service log level. (default "release")
-master_port string
    box register to master use port. (default "6667")
-mcu_rpc_port string
    master to master rpc port. (default "6668")
-mqtt_ip string
    mqtt server ip. (default "0.0.0.0")
-mqtt_tcp_port string
    mqtt connect tcp port. (default "1883")
-mqtt_ws_port string
    mqtt connect ws port. (default "1888")
-mysql_admin_db string
    avdadmin:mysql connect db. (default "avd_cluster")
-mysql_ip string
    mysql connect ip. (default "127.0.0.1")
-mysql_logic_db string
    logic:mysql connect db. (default "rtm_logic")
-mysql_mqtt_db string
    mqtt:mysql connect db. (default "avd_mqtt")
-mysql_password string
    mysql connect password. (default "pass123")
-mysql_port string
    mysql connect port. (default "3306")
-mysql_user string
    mysql connect user. (default "root")
-mysql_xconf_db string
    xconf:mysql connect db. (default "xconf")
-nginx_http_port string
    nginx listen http port. (default "80")
-nginx_https_port string
    nginx listen https port. (default "443")
-oracle_db string
    connect oracle db.
-oracle_host string
    connect oracle db host addr.
-oracle_password string
    connect oracle db password.
-oracle_user string
    connect oracle db user.
-other_cluster_rpc_addr string
    other cluster rpc host.
-redis_cluster string
```

```
    redis cluster address.
-redis_db int
    redis connect db.
-redis_host string
    redis server host.
-redis_ip string
    redis connect ip. (default "127.0.0.1")
-redis_password string
    redis connect port.
-redis_port string
    redis connect port. (default "6379")
-redis_type string
    redis server type. (default "alone")
-stdout
    log output to std.
-supervisord_ip string
    supervisord listen ip. (default "127.0.0.1")
-supervisord_port string
    supervisord listen port. (default "9010")
-with_mysql
    install Mariadb. (default true)
-with_mysql_db
    create mysql database. (default true)
[root@1e7383ac63aa cluster]#
```

自定义nginx端口和安装路径

有时候在部署服务器的时候，web服务器的默认端口是不被允许使用的，这时候就需要更改。默认的安装目录也可以通过安装参数指定。

```
-nginx_http_port string
    nginx listen http port. (default "80")
-nginx_https_port string
    nginx listen https port. (default "443")
-install_dir string
    cluster install dir. (default "/opt/avd_platform/")
```

参数参考

```
[root@1e7383ac63aa cluster]# ./deploy install \  
> -nginx_http_port=7080 \  
> -nginx_https_port=7443 \  
> -install_dir=/home/test/avd_platform \  
> xxx #其他参数
```

连接外部mysql数据库

中心服务器环境，一般对可靠性、稳定性、并发数要求较高，会部署多个master节点并绑定多个媒体服务器，这时候数据库一般会连接单独的mysql数据库。连接mysql数据库需要给予root用户权限。

```
# 连接单独的mysql相关参数
-db_driver string
    set db driver. (default "mysql")
-mysql_admin_db string
    avdadmin:mysql connect db. (default "avd_cluster")
-mysql_ip string
```

```

mysql connect ip. (default "127.0.0.1")
-mysql_logic_db string
    logic:mysql connect db. (default "rtm_logic")
-mysql_mqtt_db string
    mqtt:mysql connect db. (default "avd_mqtt")
-mysql_password string
    mysql connect password. (default "pass123")
-mysql_port string
    mysql connect port. (default "3306")
-mysql_user string
    mysql connect user. (default "root")
-mysql_xconf_db string
    xconf:mysql connect db. (default "xconf")
-with_mysql
    install Mariadb. (default true)
-with_mysql_db
    create mysql database. (default true)

```

参数说明:

- `-db_driver string`, 标识连接的数据库类型, oracle表示Oracle数据库, 默认为mysql
- `-with_mysql`, 连接外部数据库, 本地不安装, 这个参数设置为 `-with_mysql=false`
- `-with_mysql_db`, 会创建master需要用到的数据库, 并初始化, 一般采用默认值
- `-mysql_user`、`-mysql_ip`、`-mysql_port` 为mysql连接信息
- `-mysql_admin_db`、`-mysql_logic_db`、`-mysql_mqtt_db`、`-mysql_xconf_db` 为master用到的数据库, 一般采用默认值, 也可以自定义

安装参数参考

```

[root@1e7383ac63aa cluster]# ./deploy install \
> -with_mysql_db=false \
> -mysql_ip=192.168.3.60 \
> -mysql_port=3306 \
> -mysql_user=root \
> -mysql_password=pass123 \
> -mysql_admin_db=avd_cluster7000 \
> xxx #其他参数

```

另外也支持mysql8的ssl认证连接, 安装参数参考

```

./deploy install \
\
-db_driver=mysql \
-mysql_ip=192.168.3.62 \
-mysql_port=3306 \
-mysql_user=root \
-mysql_password=Admin_3tee \
-mysql_ssl_ca=/root/cert/ca.pem \
-mysql_ssl_cert=/root/cert/client-cert.pem \
-mysql_ssl_key=/root/cert/client-key.pem \
-mysql_admin_db=avd_platform7820 \
-with_mysql=false
# 这里添加了ssl连接证书参数, 客户端证书, 客户端密钥

```

连接外部Oracle数据库

数据库用户需要创建、删除表，创建、删除索引，创建、删除触发器，创建、删除序列，还有最基础的增删查改权限。

master环境，一般对可靠性、稳定性、并发数要求较高，会部署多个master节点并绑定多个媒体服务器，这时候数据库一般会连接单独的数据库。连接Oracle数据库相关参数参考。

```
#连接Oracle数据库相关参数
-db_driver string
    set db driver. (default "mysql")
-oracle_db string
    connect oracle db.
-oracle_host string
    connect oracle db host addr.
-oracle_password string
    connect oracle db password.
-oracle_user string
    connect oracle db user.
-with_mysql
    install Mariadb. (default true)
-with_mysql_db
    create mysql database. (default true)
```

参数说明：

- `-db_driver string`，标识连接的数据库类型，oracle表示Oracle数据库，默认为mysql
- `-with_mysql`，连接Oracle数据库，本地不安装，这个参数设置为 `-with_mysql=false`
- `-with_mysql_db`，会创建master需要用到的数据库，并初始化，一般采用默认值
- `-oracle_user`、`-oracle_password` 为连接Oracle数据库信息
- `-oracle_db`，默认为orcl
- `-oracle_host`，连接的Oracle数据库地址

安装参数参考

```
[root@1e7383ac63aa cluster]# ./deploy install \  
> -with_mysql_db=false \  
> -oracle_host=192.168.3.60:1521 \  
> -oracle_password=pass123 \  
> -oracle_user=C##avd_platform7100 \  
> -db_driver=oracle \  
> -oracle_db=orcl \  
> xxx \  
> #其他参数
```

连接外部postgresql数据库

master环境，一般对可靠性、稳定性、并发数要求较高，会部署多个master节点并绑定多个媒体服务器，这时候数据库一般会连接单独的数据库。连接postgresql数据库相关参数参考。

```
-db_driver string
    set db driver. (default "mysql")
-pgsql_dbname string
    connect pgsql dbname.
-pgsql_ip string
    connect pgsql ip address.
-pgsql_password string
    connect pgsql password.
-pgsql_port string
    connect pgsql port.
-pgsql_user string
    connect pgsql user.
```

参数说明:

- --db_driver=pgsql 标识连接的数据库类型, oracle表示Oracle数据库, pgsql表示postgresql数据库, 默认为mysql
- --pgsql_ip=192.168.3.60 postgresql数据库服务ip地址
- --pgsql_port=5432 postgresql数据库服务端口号
- --pgsql_password=pass123 postgresql数据库服务用户密码
- --pgsql_user=postgres postgresql数据库服务用户名
- --pgsql_dbname=test_cluster_3000 postgresql数据库服务数据库db名称
- -pgsql_schema=test postgresql 数据库非默认模式时, 需要指定schema

安装参数参考

```
./deploy install \  
-with_mysql=false \  
-with_mysql_db=false -db_driver=pgsql -pgsql_ip=192.168.3.60 -pgsql_port=5432 -  
pgsql_user=postgres -pgsql_dbname=avd_plat_postgres -pgsql_password=pass123 \  
\
```

注意: postgresql 数据库非默认模式时, 需要指定schema

```
./deploy install \  
-with_mysql=false \  
-with_mysql_db=false -db_driver=pgsql -pgsql_ip=192.168.3.60 -pgsql_port=5432 -  
pgsql_user=postgres -pgsql_dbname=avd_plat_postgres -pgsql_password=pass123 \  
\ -pgsql_schema=test \  
#数据库非默认模式时, 指定schema
```

连接外部sql server数据库

master也可以连接sql server数据库。连接数据库前, 请先在数据库上创建用于连接的database。注意, 创建数据库的时候, 数据库的排序规则要设置为Chinese_PRC_CI_AS, 否则可能出现乱码。

```
./deploy install \  
\  
-with_mysql=false \  
-db_driver=mssql \  
-mssql_ip=192.168.3.50 \  
-mssql_port=1433 \  
-mssql_user=sa \  
-mssql_password=pass@123 \  
-mssql_dbname=avd_platform \  
\  
*** \  
# 其他参数
```

连接达梦 (dameng) 数据库

注意，达梦数据库地址只支持IP，不支持域名。

master也可以连接达梦数据库，创建好用户，给予dba权限。

安装参数参考

```
./deploy install \  
-with_mysql=false \  
-db_driver=dm \  
-dm_host=192.168.3.65:5236 \  
-dm_user=avd_platform7230 \  
-dm_password=pass123 \  
\  
# 其他安装参数
```

连接达梦oceanbase数据库

master也可以连接oceanbase数据库，oceanbase数据库mysql兼容，采用mysql安装参数

安装参数参考

```
./deploy install \  
-with_mysql=false \  
-db_driver=mysql \  
-mysql_ip=192.168.3.66 \  
-mysql_port=2881 \  
-mysql_user=root \  
-mysql_admin_db=avd_platform7270 \  
-mysql_password=pass123 \  
\  
# 其他安装参数
```

连接翰高(highgo)数据库

20240830的版本已支持翰高数据库，可以通过 `./deploy install -h` 查看相关安装参数，安装的时候，数据库驱动指定为highgo。

```
-highgo_dbname string
    connect highgo dbname.
-highgo_ip string
    connect highgo ip address.
-highgo_password string
    connect highgo password.
-highgo_port string
    connect highgo port.
-highgo_schema string
    highgo schema
-highgo_user string
    connect highgo user.
```

安装实例参数参考

```
./deploy install \
\
-with_mysql=false \
-db_driver=highgo \
-highgo_ip=192.168.3.71 \
-highgo_port=5866 \
-highgo_user=sysdba \
-highgo_dbname=master9310 \
-highgo_password=password \
-highgo_schema=test \
\
# -with_mysql=false 指不安装本地的数据库，centos7环境默认会安装mysql数据库
# -highgo_schema=test 这里数据库创建了一个test模式，如果不创建，默认使用public模式
```

连接外部redis

注意，媒体服务器卸载重装且用同一个redis，redis需要清空下数据。

master部署，一般部署多个master，各个master连接同一个数据库，redis也不例外，需要连接相同的redis。连接redis支持单点和redis cluster集群，相关参数参考。

```
# redis相关参数
-redis_cluster string
    redis cluster address.
-redis_db int
    redis connect db.
-redis_host string
    redis server host.
-redis_ip string
    redis connect ip. (default "127.0.0.1")
-redis_password string
    redis connect port.
-redis_port string
    redis connect port. (default "6379")
-redis_type string
    redis server type. (default "alone")
```

参数说明

- `-redis_cluster`，连接redis cluster集群地址，兼容老版本，已不使用
- `-redis_db`，连接默认的db，也可指定。redis数据库切换。redis集群不支持数据库，redis单机和哨兵支持数据库，默认为0。
- `-redis_host`，连接redis地址，例如-
`redis_host=127.0.0.1:6379,127.0.0.1:6380,127.0.0.1:6381`
- `-redis_ip`，连接redis的地址，连接单点时用到
- `-redis_password`，redis的连接密码
- `-redis_port`，连接redis端口
- `-redis_type`，链接的redis服务的模式，单机为alone，哨兵为sentinel，集群为cluster。如果是单机默认，需要配置redis_ip和redis_port两个配置项。如果是哨兵和集群，则配置redis_host配置项

连接redis cluster集群相关参数参考

连接redis集群的时候，需要首先指定redis类型，这里选择`-redis_type=cluster`。

```
[root@1e7383ac63aa cluster]# ./deploy install \
> -redis_type=cluster \
> -
redis_host=192.168.3.60:6380,192.168.3.60:6381,192.168.3.60:6382,192.168.3.60:6383,192.168.3.60:6384,192.168.3.60:6385 \
> -redis_password=pass123 \
> xxx \ #其他参数
```

连接redis单点相关参数参考

连接redis单点的时候，需要首先指定redis类型，这里选择`-redis_type=alone`。

```
[root@1e7383ac63aa cluster]# ./deploy install \
> -redis_type=alone \
> -redis_ip=192.168.3.60 \
> -redis_port=6379
> -redis_password=pass123 \
> xxx \ #其他参数
```

连接redis哨兵相关安装参数

连接redis哨兵的时候，需要首先指定redis类型，这里选择`-redis_type=sentinel`

相关安装参数参考：

```
./deploy install \
-with_redis=false \
-redis_type=sentinel \
-redis_host=192.168.3.60:27000,192.168.3.60:27001,192.168.3.60:27002 \
-redis_db=2 \
-redis_password=pass123 \
# 其他安装参数
```

如果redis哨兵集群，哨兵也有密码验证，则需要添加`-redissentinel_password=`、`-sentinel_name=`参数。`-sentinel_name`指redis哨兵在配置的时候，指定的master的名称，由`sentinel monitor mymaster 192.168.0.117 7501 2`指定，默认情况下`-sentinel_name=mymaster`。

相关安装参数参考：

```

./deploy install \
\
-redis_host=192.168.3.60:26400,192.168.3.60:26401,192.168.3.60:26402 \
-redis_password=pass123 \
-redis_type=sentinel \
-with_redis=false \
-
redis_sentinel_password=ed4c39b015b0e46f074dbfd0a9a4ab278f63340a6d640999f25c68a932fef815 \
-sentinel_name=mymaster \
\
# 其他安装参数

```

如果已经安装好的服务器需要修改配置，在安装目录 `conf/install.yaml` 修改。

```

redis:
  ip: 127.0.0.1
  port: "6379"
  password: pass123
  sentinel_password:
    "ed4c39b015b0e46f074dbfd0a9a4ab278f63340a6d640999f25c68a932fef815"
  sentinel_name: mymaster
  db: 1
  cluster: ""
  type: sentinel
  host: 192.168.3.60:26400,192.168.3.60:26401,192.168.3.60:26402

```

多master相关部署

注意：部署的时候，要保障2台服务器的时间一致。

在部署中心服务器的时候，往往需要安装多个master。这里将介绍2个master的安装部署参数说明。

```

#多master相关部署参数
-cluster_rpc_port string
    cluster rpc port. (default "6669")
-cluster_server_id string
    master server register name. (default "3tee.avd.srv.avdcluster")
-is_master string
    true:cluster server is master false: cluster server is slave (default
"true")
-other_cluster_rpc_addr string
    other cluster rpc host.
# 6669/tcp端口是需要开放的

```

参数说明

- `-cluster_rpc_port`，中心服务器cluster模块rpc服务端口，默认6669
- `-other_cluster_rpc_addr`，多master模式时其它master的cluster模块的rpc服务地址，例如192.168.1.168:6669
- `-cluster_server_id`，多master的时候，各个master的值要不一样，比如3tee.avd.srv.avdcluster61和3tee.avd.srv.avdcluster62
- `-is_master`，是否是主的master节点，默认为true，多master环境的备用master为false

部署实例

部署2个maser，其他A为主，B为备用。A的IP为192.168.3.61，B的IP为192.168.3.60。连接Oracle数据和redis集群。

A机器的安装参数为：

```
[root@1e7383ac63aa cluster]# ./deploy install \  
> -cluster_rpc_port=6669 \  
> -other_cluster_rpc_addr=192.168.3.60:6669 \  
> -cluster_server_id=3tee.avd.srv.avdcluster61 \  
> -is_master=true \  
> xxx #其他参数  
  
# 3tee.avd.srv.avdcluster61客户自定义，保障两台master的该参数不同即可
```

B机器的安装参数为：

```
[root@1e7383ac63aa cluster]# ./deploy install \  
> -cluster_rpc_port=6669 \  
> -other_cluster_rpc_addr=192.168.3.61:6669 \  
> -cluster_server_id=3tee.avd.srv.avdcluster60 \  
> -is_master=false \  
> -with_mysql_db=false \  
> xxx # 其他参数  
  
# -with_mysql_db=false 因为连接同一个数据库，这里就不需要初始化数据库了
```

普通用户安装

从2021.12.15以后的中心服务器安装包，开始支持普通用户安装。安装注意项：

- 普通用户安装因为没有root权，数据库不能安装，连接外部数据库。
- 普通用户不能使用1024前的端口，http和https端口要大于1024。
- 普通用户默认安装目录要自定义
- 普通用户安装，没有开机自启动（开机自启动脚本拷贝权限不足），需要用户单独处理

安装前，要先运行安装目录下的 `source setenv.sh` 脚本，然后安装。（`source setenv.sh` 而不是 `./setenv.sh`）

安装参数：

```
./deploy install \  
-with_mysql=false \  
-mysql_admin_db=cluster30 \  
-mysql_ip=192.168.3.60 \  
-nginx_http_port=1080 \  
-nginx_https_port=1443 \  
-install_dir=/home/shanks/avd_platform \  
xxx \  
# 其他参数
```

等待安装完成，然后设置开机自启动。开机自启动设置需要sudo权限。

这个通过配置 `/etc/rc.d/rc.local` 文件的方法，开机启动设置方法：

1. `sudo vi /etc/rc.d/rc.local`，然后添加

```
su shanks -c "cd /home/shanks/avd_platform;./ctl.sh start"
# shanks 为普通用户
# -c 后面跟的是要执行的脚本，切换目录和执行启动脚本
```

2. `sudo chmod +x /etc/rc.d/rc.local` ,赋予文件可执行权限

3. 最后重启机器，检查是否设置生效。

数据库、redis密码加密安装

在解压后的cluster目录，运行 `./bin/encryptiontool -h`，可以查看帮助说明。

- `-h` show this help. 展示帮助说明
- `-help`
show this help. 展示帮助说明
- `-key string`
des加密密钥,key的长度只能是8位 (default "12345678")
- `-password string`
需要进行加密的密码，密码不能带冒号(:)

以下为具体的安装实例

首先对数据库的密码进行加密，密码为 `pass123`，通过工具加密后的密码为：
`9RpUvSGQt9g=:3Tee_123`

```
[root@localhost cluster]# ./bin/encryptiontool -password pass123 -key 3Tee_123
des加密密钥key: 3Tee_123
des加密密码password: pass123
des加密生成加密串: 9RpUvSGQt9g=:3Tee_123
回车键退出。。。
```

安装参数举例：

```
./deploy install \  
***** \  
-with_redis=false \  
-redis_type=cluster \  
-  
redis_host=192.168.3.60:6290,192.168.3.60:6291,192.168.3.60:6292,192.168.3.60:6293,192.168.3.60:6294,192.168.3.60:6295 \  
-redis_password=9RpUvSGQt9g=:3Tee_123 \  
\  
-with_mysql=false \  
-db_driver=mysql \  
-mysql_user=root \  
-mysql_password=9RpUvSGQt9g=:3Tee_123 \  
-mysql_ip=192.168.3.61 \  
-mysql_admin_db=avd_plat_mysql \  
# 这里采用mysql数据库和redis集群，分别对数据库密和redis密码进行加密，密码为：pass123，加密key为3Tee_123
```

添加restApi请求前缀

在部署的时候，因安全等要求，媒体服务器不直接暴露，需要通过代理。运维已经做了代理，比如我们通过 `https://proxyServer/path/` 来访问部署在隔离网段的中心服务器 `https://clusterServer`，则需要添加restApi请求前缀，即 `/path/`。

1. 通过安装参数添加

我们在安装的时候，可以通过 `./deploy install -h` 获取，`-rest_api_prefix_path string` 可指定restApi请求前缀，`-local_ip_and_domain string` 来指定服务器内网地址。

```
./deploy install \  
-rest_api_prefix_path=/js/zhyt/cluster/ \  
xxx \  
# 其他参数  
  
# /js/zhyt/cluster/ 是restApi请求前缀  
# https://proxyServer/js/zhyt/cluster/ 则可以访问到部署在隔离网段的服务器  
https://clusterserver
```

2. 通过配置文件添加

服务器的安装目录下，打开配置文件conf/install.yaml，修改 `rest_api_prefix_path` 项。

```
cluster:  
  help: false  
  install_dir: /data/avd_platform/  
  with_mysql_db: true  
  with_mysql: false  
  with_redis: false  
  db_driver: mysql  
  redirect_proto_type: ""  
  rest_api_prefix_path: /js/zhyt/cluster/ #已添加了restAPI路径  
  local_ip_and_domain: ""  
  supervisord:  
    ip: 127.0.0.1  
    port: "9010"  
  
# local_ip_and_domain: 192.168.1.124 本机ip和域名，多个以 ， 隔开
```

3. `enable_use_inner_api_prefix: false` 此参数主要用于需要从内部网络、外部网络同时访问master服务器的情况，内部网络和外部网络地址不一致，且添加了访问前缀。master会分配任务到媒体服务器（`media_server`），当我们从内部网络访问master的时候，master会返回给我们 `media_server`的内部网络地址（`media_server`后台配置的内网地址）。如果 `media_server`后台配置的内网本机地址，我们希望返回的 `media_server`地址不带前缀，需要 `enable_use_inner_api_prefix: false`。如果 `media_server`后台配置的代理服务器地址，则我们希望返回的 `media_server`地址带上前缀。则需要 `enable_use_inner_api_prefix: true`。
4. 在需要区分内外网的时候，master地址设置，要配置“master公网地址”、“master内网地址”等。配置路径为：中心服务器管理系统---平台管理---系统设置--集群设置---集群地址设置。服务器的https地址和http地址，不带协议头，比如 `cd.nice2meet.cn:7790;cd.nice2meet.cn:7792`(公网) `192.168.3.65:7790;192.168.3.65:7792` (内网)

注意：20230228 后面的版本以下参数已废弃

- `local_ip_and_domain`: 已废弃

所有请求被重定向为https

服务器在安装的时候，可以强行把请求重定向，安装参数 `-redirect_proto_type=https`，表示master的http请求强转为https。

注意：`redirect_proto_type`: 重定向协议头，例如配置为https。master重定向的时候优先使用https。

如果是已经安装好的环境。需要修改master的配置文件

1. 打开conf/install.yaml文件,
2. redirect_proto_type: "" 修改为 redirect_proto_type: "https", 然后保存。
3. reload服务器。 ./ctl.sh reload
4. 最后重启服务器, ./ctl.sh stop ./ctl.sh start

安全相关安装参数说明

- -enable_restrict_download=true #只能下载录制服务配置的录制存放目录下的文件
- -enabling_safe_mode=true #严格模式文件下载的时候验证token
- value_same_as_box=false #在调用 /room/mix_record/stop、room/get_record 接口时; 根据配置文件参数value_same_as_box的值返回不同的下载地址。value_same_as_box为true时返回media_server的下载地址, 为false时返回master的下载地址。
- attachment_safe_mode: false # 设置为ture时, 后台隐藏下载按钮, 下载只能用post方法。

也可以在conf/install.yaml中对相关参数进行配置, 配置完成后重新加载并重启服务器。

```
./ctl.sh reload #重新加载配置
./ctl.sh stop # 停止
./ctl.sh start # 启动
```

master多中心相关配置

20230423后的版本, master支持部署在多个中心, 比如杭州部署2个master、成都也部署2个master。既支持以前的部署方式(部署2台master), 也支持同时部署多台master(超过2台)。比如, 我们部署了4台master, 其中2台在成都的机房, 另外2台在杭州的机房。配置文件新装参数cluster_group, 用于master分组。

情况一, 部署2台master, 服务器采用默认分组, 或者通过参数自定义分组名。服务器参数设置方面, 我们可以采用以前的方式, othercluster_rpc_addr设置为另外一台机器的cluster_rpc_addr; 我们也可以采用新的方式, 直接设置 -cluster_rpc_port=服务器IP:6669, 忽略掉 -other_cluster_rpc_addr。

传统安装方式:

```
#cluster1
./deploy install \
\
-master_port=6667 \
-cluster_rpc_port=6669 \
-other_cluster_rpc_addr=cluster2_server_IP:6669 \ #cluster2_server_IP 另外一台服务器IP
-cluster_server_id=cluster1 \
-is_master=true \
\
... # 其他安装参数

#cluster2
./deploy install \
\
-master_port=6667 \
-cluster_rpc_port=6669 \
-other_cluster_rpc_addr=cluster1_server_IP:6669 \ #cluster1_server_IP 另外一台服务器IP
```

```
-cluster_server_id=cluster2 \  
-is_master=false \  
\   
-with_mysql_db=false \  
... # 其他安装参数
```

最新的安装方式

```
#cluster1  
./deploy install \  
\   
-master_port=6667 \  
-cluster_rpc_port=server_ip:6669 \  
#可以是本机IP或本机映射后的IP, 另外一台服务器能访问即可  
-cluster_server_id=cluster1 \  
-is_master=true \  
\   
... # 其他安装参数  
  
#cluster2  
./deploy install \  
\   
-master_port=6667 \  
-cluster_rpc_port=server_ip:6669 \  
#可以是本机IP或本机映射后的IP, 另外一台服务器能访问即可  
-cluster_server_id=cluster2 \  
-is_master=false \  
\   
-with_mysql_db=false \  
... # 其他安装参数
```

情况二，部署超过2台master，我们可以分组，也可以不分组。分组，主要针对多中心，彼此无法访问对方，比如部署在杭州是媒体服务器没办法直接访问在chengdu机房的服务器。-cluster_group 我们可通过这个参数指定分组。

```
./deploy install -cluster_group=chengdu \  
其他参数
```

部署超过2master，我们必须在安装的时候指定本机的rpc地址，通过 -cluster_rpc_port=server_ip:6669 指定，可以是本机IP或本机映射后的IP，同组服务器能访问即可。主master设置为 -is_master=true，其他master设置为 -is_master=false。这里以双中心，部署4个master为例，列举安装参数参考。

```
#分了2组, chengdu 和 hangzhou  
#主master设置为-is_master=true, 其他master设置为-is_master=false  
#除了第一个master, 其他master都不初始化数据库-with_mysql_db=false  
  
#cluster1  
./deploy install -cluster_group=chengdu \  
\
```

```

-master_port=6667 \
-cluster_rpc_port=server_ip:6669 \ #可以是本机IP或本机映射后的IP，同组服务器能访问即可
-cluster_server_id=cluster1 \
-is_master=true \
\
... # 其他安装参数

#cluster2
./deploy install -cluster_group=chengdu \
\
-master_port=6667 \
-cluster_rpc_port=server_ip:6669 \ #可以是本机IP或本机映射后的IP，同组服务器能访问即可
-cluster_server_id=cluster2 \
-is_master=false \
\
-with_mysql_db=false \
... # 其他安装参数

#cluster3
./deploy install -cluster_group=hangzhou \
\
-master_port=6667 \
-cluster_rpc_port=server_ip:6669 \ #可以是本机IP或本机映射后的IP，同组服务器能访问即可
-cluster_server_id=cluster3 \
-is_master=false \
\
-with_mysql_db=false \
... # 其他安装参数

#cluster4
./deploy install -cluster_group=hangzhou \
\
-master_port=6667 \
-cluster_rpc_port=server_ip:6669 \ #可以是本机IP或本机映射后的IP，同组服务器能访问即可
-cluster_server_id=cluster4 \
-is_master=false \
\
-with_mysql_db=false \
... # 其他安装参数

```

redis key 前缀

安装master服务器，`./deploy install -h`可以查询到`-redis_key_prefix`参数，比如`-redis_key_prefix=tee3`，可以在所有redis key前面添加tee3，一般应用在master服务器和其他应用共用一个redis库的时候，避免出现key冲突。tee3也可以是其他标识符。

```

./deploy install -redis_key_prefix=tee3 \
xxx \ #其他参数

```

已经安装好的服务器，升级到20231207后的版本，需要修改配置文件才能生效。

打开安装目录的配置文件(一般默认/opt/avd_platform)，在配置文件conf/install.yaml中修改`redis_key_prefix: tee3`配置项，修改完成后，reload服务器，然后重启整个服务器。

```
./ctl.sh reload
./ctl.sh stop
./ctl.sh start
```

允许跨域访问配置

参数`nginx_origin_addrs`，配置允许跨域的域名，可以配置多个，中间用逗号隔开。注意，协议头、域名都需要，比如

`nginx_origin_addrs=https://cd.nice2meet.cn,https://avd.nice2meet.cn:9889`，端口如果是非443和80，也是必须要加上。

安装的时候参数设置：

```
./deploy install -
nginx_origin_addrs=https://cd.nice2meet.cn,https://avd.nice2meet.cn:9889 \
xxx # 其他参数
```

20231207以前的版本升级，需要修改安装目录`conf/install.yaml`中`nginx_origin_addrs: https://xxx`配置项，然后`reload`服务器，并重启。

```
./ctl.sh reload
./ctl.sh stop
./ctl.sh start
```

设置数据库连接保持时长

新增安装参数和配置，用于设置数据库连接保持时长，控制数据库使用长或短连接。参数名为`database_conn_max_lifetime`，单位秒，0为默认值，连接会长期保持。另外也可以设置为其他整数值，比如120，2分钟后会重新建立新的连接。安装参数设置参考：

```
./deploy install \
-database_conn_max_lifetime=0 \
# 这里采用默认值，保持长连接
```

如果更新，该参数会默认添加，该参数为服务器20240816版本添加，更新后请检查`conf/install.yaml`配置文件`database_conn_max_lifetime: "0"`配置项，根据需求修改配置项，如果没有该配置，请手动添加。修改配置后，需要`reload`服务器配置，然后重启整个服务器。

```
./ctl.sh reload
./ctl.sh start
./ctl.sh stop
```

配置

配置ssl证书

安装完成以后，会默认启动`nginx` web服务，网关服务，`master`服务。为了支持`https`，一般还需要手动配置证书。叁体采用的`nginx` web服务器，`ssl`证书也需要`nginx`格式。叁体媒体服务器更新会覆盖原目录，所以证书路径一般为安装目录的同级目录，比如`/opt/cert/ssl.crt`。

1. 将`ssl`证书上传到服务器，以`/opt/cert/ssl.crt`、`/opt/cert/ssk.key`为例
2. 在安装目录下，打开配置文件`confd/templates/nginx.conf.tpl`，安装目录默认为`/opt/avd_platform`。在`http`模块，`server`配置项，找到以下配置选项

```
ssl_certificate    ssl.crt;
ssl_certificate_key  ssl.key;
```

修改为:

```
ssl_certificate    /opt/cert/ssl.crt; #以颁发的证书为准, 不一定crt结尾
ssl_certificate_key  /opt/cert/ssl.key;
```

3. 修改完成后, 保存退出
4. 在安装目录, 默认为 /opt/avd_platform 重启nginx, 运行 `./ctl.sh restart_service nginx` 完成重启。

数据库、redis密码加密配置

在安装目录, 运行 `./bin/encryptiontool -h`, 可以查看帮助说明。

- -h show this help. 展示帮助说明
- -help
show this help. 展示帮助说明
- -key string
des加密密钥,key的长度只能是8位 (default "12345678")
- -password string
需要进行加密的密码, 密码不能带冒号(:)

首先对数据库的密码进行加密, 密码为 `pass123`, 通过工具加密后的密码为:
`9RpUvSGQt9g=:3Tee_123`

```
[root@localhost cluster]# ./bin/encryptiontool -password pass123 -key 3Tee_123
des加密密钥key: 3Tee_123
des加密密码password: pass123
des加密生成加密串: 9RpUvSGQt9g=:3Tee_123
回车键退出。。。
```

然后修改配置文件 `vim conf/install.yaml`

```
pgsql:
  ip: 192.168.3.65
  port: "5432"
  dbname: avd_platform7510
  user: postgres
  password: rbyHZQJQlNI=:yue_123A
# 这里以postgresql为例, 修改密码为: rbyHZQJQlNI=:yue_123A, redis以及其他数据库类似
```

修改完成后保存, 然后重新加载配置, 重新启动服务器。

```
./ctl.sh reload
./ctl.sh stop
./ctl.sh start
```

nginx proxy_set_header host 取值可配置

新增参数说明:

- `http_hosts_mod`, 默认值为`$http_host`, 支持内外网端口不一致; 如果漏洞扫描不让使用`$http_host`可配置`$host:$server_port`, 但是必须保证内外网端口一致。可打开配置文件`conf/install.yaml`进行配置。

```
nginx:
  http_port: "7331"
  https_port: "7330"
  http_hosts_mod: $host:$server_port
  http_hosts_whitelist: ""

# 或者http_hosts_mod: $http_host 可以不配, 默认为$http_host
```

- 参数`http_hosts_whitelist`, 默认为空, 如果需要配置白名单, 则需要根据`http_hosts_mod`值来配置。如果`http_hosts_mod`为`$http_host`时, 则白名单的格式为`http_hosts_whitelist: "192.168.3.66:7310|cd.nice2meet.cn:7310|127.0.0.1:7310"`的格式, 即IP或域名加端口, 如ip:port, 有多个的话中间用"`|`"分隔开。如果`http_hosts_mod`为`$host:$server_port`时, 则白名单的格式为`http_hosts_whitelist: "192.168.3.66|cd.nice2meet.cn|127.0.0.1"`的格式, 即单格式IP或者域名, 不能携带端口, 有多个的话中间用"`|`"分隔开。

注: 如果需要配置白名单的时候, 127.0.0.1必须要填, nginx会检查请求进来的host和白名单的host是否匹配, 不配会返回403, 禁止访问。

```
# 示例测试环境:
# 内网地址:192.168.3.66
# 公网地址: cd.nice2meet.cn

# http_hosts_mod配置为$http_host时, 没有配置默认值为$http_host
# 配置的白名单格式
nginx:
  http_port: "7351"
  https_port: "7350"
  http_hosts_mod: $http_host
  http_hosts_whitelist:
    "192.168.3.66:7350|192.168.3.66:7351|127.0.0.1:7350|127.0.0.1:7351|cd.nice2meet.cn:7351|cd.nice2meet.cn:7351"
# 配置白名单的时候, 127.0.0.1必须要填, 上面的配置允许公网和内网访问, 需要限制不同的网络只需要去掉对应的地址即可, 如需要限制公网, 去掉上边的公网地址, 需要限制内网, 去掉上边的内网地址即可(127.0.0.1保留)。

# http_hosts_mod配置为$host:$server_port时
# 配置的白名单格式
nginx:
  http_port: "7351"
  https_port: "7350"
  http_hosts_mod: $host:$server_port
  http_hosts_whitelist: "192.168.3.66|127.0.0.1|cd.nice2meet.cn"
# 配置白名单的时候, 127.0.0.1必须要填, 上面的配置允许公网和内网访问, 需要限制不同的网络只需要去掉对应的地址即可, 如需要限制公网, 去掉上边的公网地址, 需要限制内网, 去掉上边的内网地址即可(127.0.0.1保留)。
```

配置方法参考, 在服务器安装目录打开配置文件 `conf/install.conf`, 在nginx配置项添加配置。

```
# 例如
nginx:
  http_port: "7351"
  https_port: "7350"
  http_hosts_mod: $host:$server_port
  http_hosts_whitelist: "192.168.3.66|127.0.0.1|cd.nice2meet.cn"
```

然后reload服务器，重启服务器。

```
./ctl.sh reload
./ctl.sh stop
./ctl.sh start
```

安全相关配置

20230830以后的master，安装完成后，默认开启安全模式。如果是升级的，且以前没有开启安全模式，需要手动开启。如果需要开启安全模式，master（或称呼为中心服务器）和媒体服务器（或称呼为media_server）都需要开启，上传和下载白名单都需要单独配置。

如何进行配置呢？首先，登录“master管理系统”，选择“平台管理”，然后在“系统设置-集群设置”页面，选择“文件上传白名单”，勾选对应选项，配置文件上传路径白名单，然后保存。注意，文件上传路径白名单，一定要有权限访问。即安装叁体服务的用户，有访问该路径的权限。



媒体服务器，需要登录媒体服务器后台，进行配置。

版本回退

从 master_1.0.1-20230620-15-38-43.tar 以前的版本升级到该版本或者更新的版本后，需要回退服务器，需要执行下面的操作。

1. 首先关闭服务器，在安装目录运行 `./ctl.sh stop`。服务器关闭后，备份下当前安装目录，默认安装目录为 `/opt/avd_platform`，我们可以将其备份为 `/opt/bak`。
2. 叁体服务器升级的时候，会在安装目录的同级目录生成备份文件，默认为 `avd_platform_20240119105314/` 这样的格式。我们将备份目录还原回去，这里将 `avd_platform_20240119105314/` 重命名为 `avd_platform`。
3. 还原数据库，我们解压 `master_1.0.1-20230620-15-38-43.tar` 或者你更新的版本的包，在解压的 `master/service/avd_admin/conf/reverse/` 目录下，有用于还原的sql文件。mysql.sql针对

mysql数据库, oracle.sql针对Oracle数据库, postgresql.sql针对postgresql数据库, sqlserver.sql针对sql server数据库。用数据库客户端连接数据库, 在对应数据库执行下该sql文件即可还原数据库到回退版本。

4. 最后, 重启服务器即可 `./ctl.sh start` 。

更新执行增量sql

服务器更新, 如果所给数据库用户没有权限, 导致更新数据库失败, 需要手动执行更新包新增的sql文件。以mysql为例, 在解压后的 `master/service/avd_admin/conf/db/` 目录下, 新增文件都是按顺序命名的。对比两个版本的db目录, 新增了1.3.*.sql之类的文件, 只需要执行新增文件即可。需要注意的是, mysql数据库, 如果有 `master/service/avd_admin/conf/db_new` 目录, 也需要对比, 执行除去1.0.0.sql的新增文件 (db_new中的1.0.0.sql千万别手动执行) 。

其他数据库有单独的目录用于存储sql文件, 分别为:

- `service/avd_admin/conf/pgsqldb/` 为postgresql 数据库
- `service/avd_admin/conf/oracledb/` 为Oracle数据库
- `service/avd_admin/conf/sqlserverdb/` 为sql server 数据库

初始化

服务器安装好后, 需要对服务器进行初始化。master的初始化地址为

`https://server_address:port/server_admin/`, `server_address` 为master的域名或者ip, `port` 为https端口号, `server_admin` 为访问路径。

🔒 不安全 | https://192.168.3.60:20000/cluster_admin/#/init

管理员初始化

初始化完成后登录后台，进行服务器基本配置。我们可以在右上角看到**管理按钮**，分为**会议管理**、**企业管理**和**平台管理**。多个master的环境，只需要初始化主要master即可，因为都是连接的同一个数据库。下面将对基本设置进行介绍。



授权

服务器安装完成后，需要对服务器进行授权，才可以使用。对服务器授权的方式有**文件授权**、**网络授权**、**硬件授权**。文件授权、硬件授权可用于离线环境。网络授权，需要服务器能够访问参体的注册服务器，注册服务器的地址为 `register.3tee.cn`。

网络授权

要求：服务器能够访问公网，即能连接参体注册服务器，。注册服务器地址为：`register.3tee.cn`，操作如下：

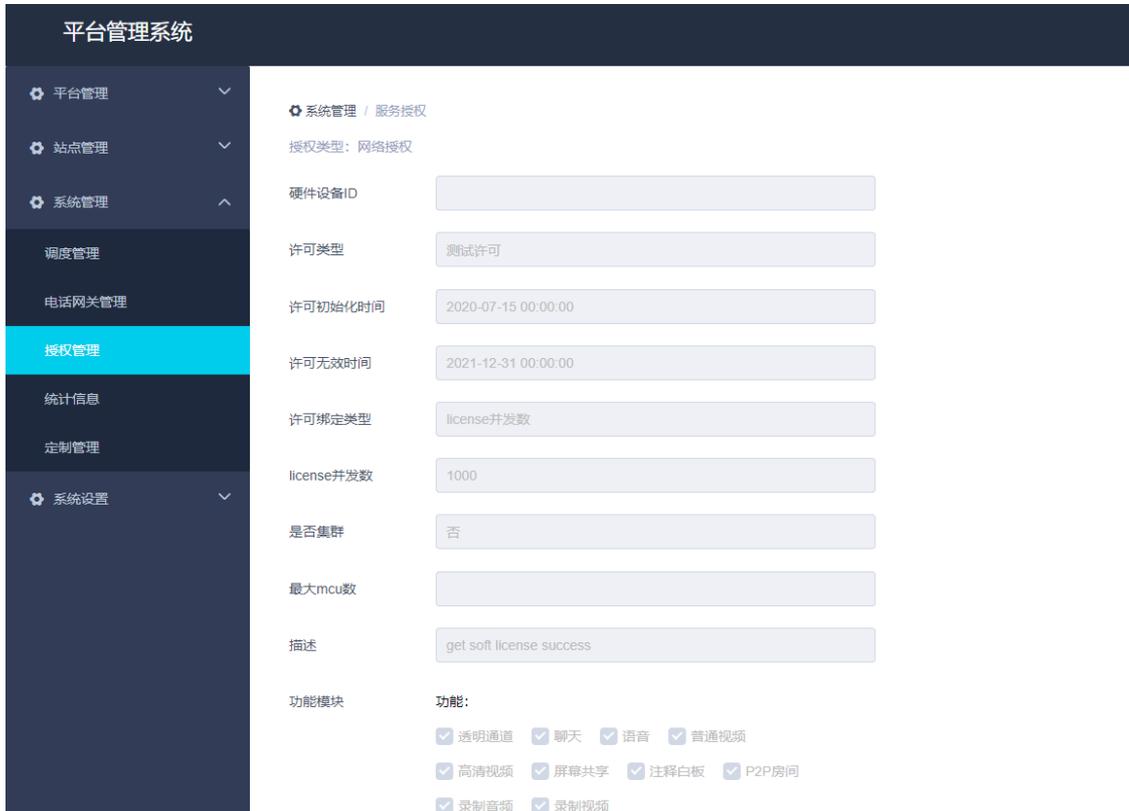
1. 需要向参体工作人员申请**网络授权**，参体会返回**网络授权文件**（一张图片），包含了授权相关记录。其中**permitAccess**对应用户名称、**permitSecret**对应用户密码，用于配置到master平台管理系统的授权页面，完成授权。以下截图为参体提供的**网络授权文件**



2. 登陆平台管理系统，点击右上角的**管理按钮**，选择**平台管理**，在**系统管理--授权管理**页面，点击**申请网络授权**，注册服务器地址不用变更，填入**用户名**、**用户密码**，保存即可生效。



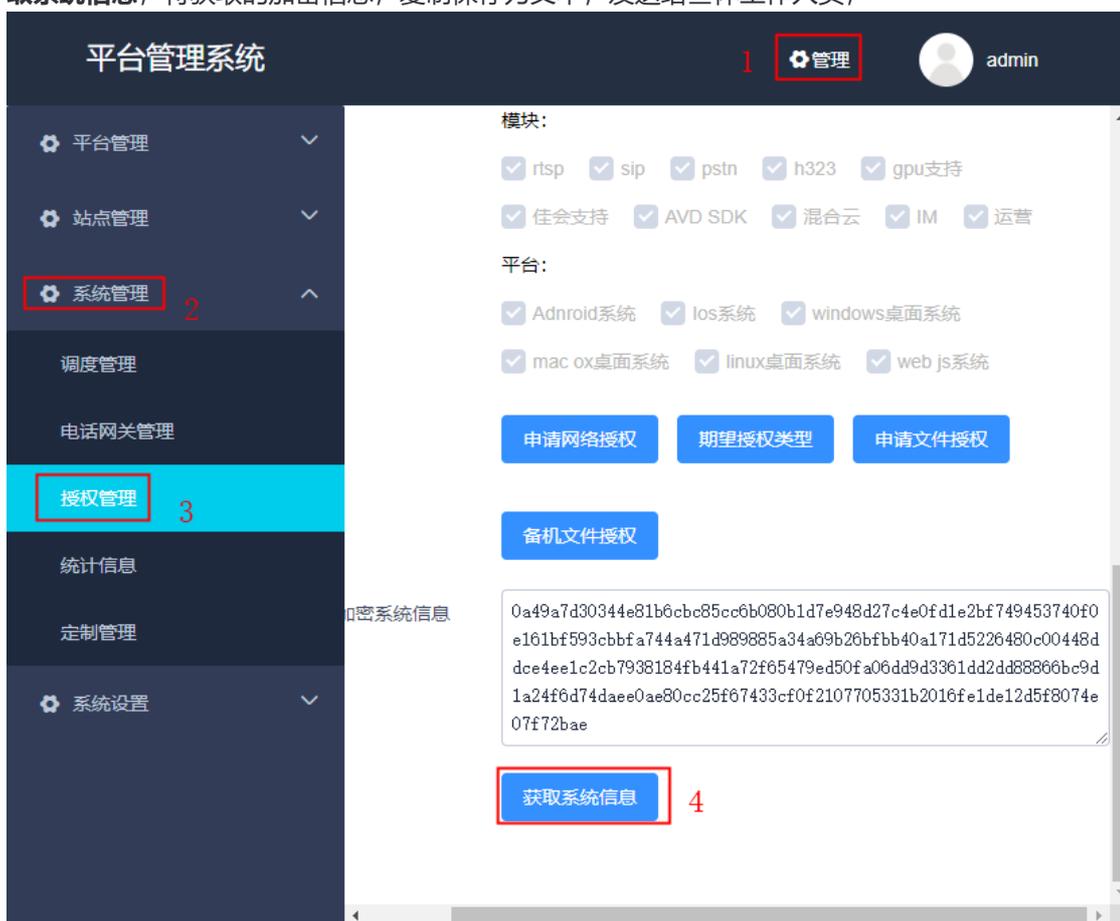
3. 授权成功后，可以在**授权管理**页面查看具体授权信息，包括许可初始化时间、许可无效时间、license并发数、功能模块等。授权成功的截图如下：



文件授权

文件授权，一般用于离线环境，比如服务器部署在物理隔离Internet的环境里面。授权的操作方法如下：

1. 登陆平台管理系统，点击右上角的**管理**按钮，选择**平台管理**，在**系统管理--授权管理**页面，点击**获取系统信息**，将获取的加密信息，复制保存为文本，发送给参体工作人员；



2. 参体会返回“文件授权文件”（加密串，文本文件），在**系统管理--授权管理**页面，点击**申请文件授权**，将**加密串**粘贴到**文件授权证书**栏，点击“确定”，即可生效。在主备两个master的请客下，主服务器的文件授权点击“**申请文件授权**”，备服务器点击“**备机文件授权**”
其中，文件授权具体参考如下：

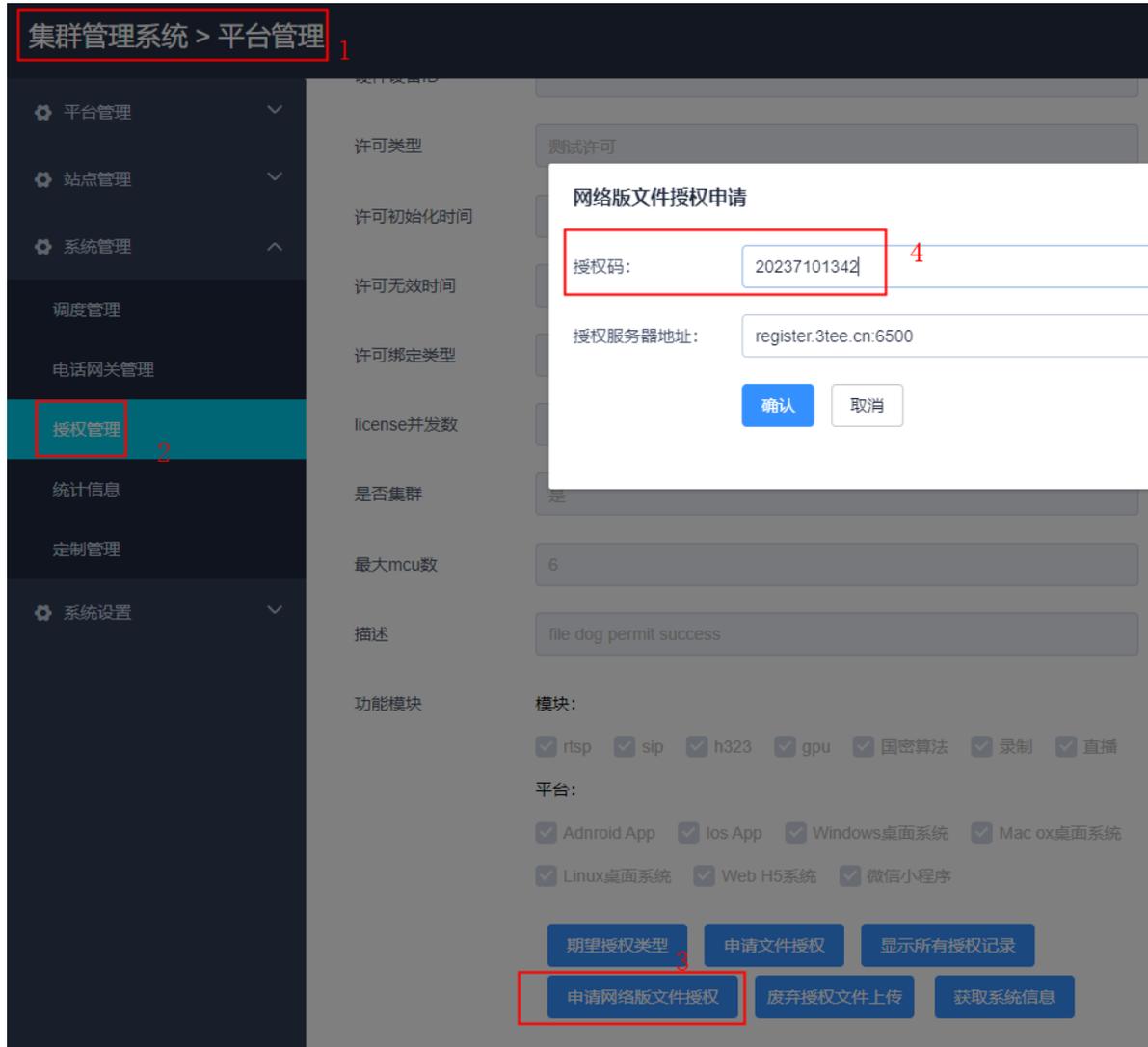
```
94fa0beaa78c1f8f1392888e44f089e2a1159385ef220f509573957aba0fa67947c2d0a2ec34
96b21e2467551d548e1dabc5ccaa7c51dd40e14273732ca62c95f526058ec426fdcf86c51dd7
c40b4d9135a18f697aa447550ab3c98c250cb40a77f363b5ebd3695a70d7f7ce61d9f668a5e1
24d723b5e866c6a47246f0683e49$23692fd4a085bd9281d09d6adcbf03f4764a1d05938a7a6
c5c531a793560977b5af72340a331be55dcfc2982d8d3eb49f69b5ddbc4aa4bd227ef5a9c3e4
d585dc71c1bab1b2a5bc08a157067486c5d9d5cc97a05ad286cc254446a77cc3058d0835775c
8dcc0147d9b7da3d8fcc96aba17ef45e3bdf2d23867f659fe5ab24eeb$5536bd222e5c76e4c7
d242d779a8d6e77a9eeec4ad13a7df3e3155b6c43f3c514ee3c914a5c3b6ede9ea2942293a37
48c16e666d490723c8a52b2adc50122091cfff58e545915752d4bf4f3dbfc9c87c51d8aeb3
71f1ea1ca166d10f22c48d322f66c52dad9bd7ff9679c0058e6ea1b464222475bdd93d09058b
a08d8f2d82$73faa314990f95e3f878a8eb08d158a65a481d0c06362ced594d1fa15335c4066
5275d8a55315e04f9f080ee45ff0f8f977f1c3e9c636ca339f9452c07b5301df618dd4d2611a
baca73b1be87dc7f147496af3e8ee02dcee816f643a89bfc2cf63e8eb7c14d54e8502a2265ea
90b4cc9dd498847f69cc81f21b1c204f0b37fa7
```



3. 授权成功后，可以在**授权管理**页面查看具体授权信息，包括许可初始化时间、许可无效时间、license并发数、功能模块等。

网络版文件授权

网络版文件授权，授权的时候需要能够联网，即能够访问叁体的注册服务器register.3tee.cn:6500，授权完成后可以离线使用。需要先向叁体工作人员申请授权码，授权通过后，叁体会返回一个授权码（类似20237101xxxx）。然后，登录master管理系统，在右上角“管理”按钮选择平台管理，在“系统管理-授权管理”页面，选择申请网络版文件授权，填写授权码（类似20237101xxxx），点击“确定”，完全授权。



硬件授权

硬件授权由叁体工作人员邮寄，u盘的形式，插到服务器上即可完成授权，一般用于正式授权。

卸载

到服务器安装目录，运行 `./deploy uninstall` 即可完成卸载

更新

媒体服务器会不断更新，更新也相当简单，运行更新脚本即可。解压后的master安装包，运行 `./deploy update -install_dir=/opt/avd_platform`。其中 `-install_dir` 为默认安装目录，您所安装的位置可能与此不同。注意，如果原目录存在master目录，解压前需要先删除master目录，然后再解压。

master节点服务及端口说明

服务说明

服务名称	功能	日志目录	备注	说明
etcd	master节点内部服务发现使用	安装工目录/logs/etcd/		
nginx	web服务器,提供反向代理的功能	安装工目录/logs/Nginx/		
gateway	所有restapi路由调度分发网关	安装工目录/logs/gateway/		
avdadmin	restapi 逻辑处理服务	安装工目录/logs/avdadmin/		
mqtt	mqtt信令服务器,用户邀请其他用户加入房间	安装工目录/logs/mqtt/		
avdcluster	media_server注册到mastercluster等相关业务逻辑处理	安装工目录/logs/avdcluster/		
clean	日志清理服务,清除本地目录log日志文件	安装工目录/logs/clean/		
redis	存储服务	安装工目录/logs/redis/		如果客户自己提供此服务,这个可以不启动

端口说明

端口说明主要针对,服务器内外网端口,内部端口一般安装都采用默认值。外部端口,可以在安装的时候自定义。**nginx https请求端口**、**nginx http请求端口**这两个端口用于支持基本的master入口,一般必需开启。

外部端口(需要开放防火墙或者映射到公网的端口)

端口	协议类型	说明	防火墙转发层级	是否有证书
443	https	nginx https请求端口	7层	需要证书
80	http	nginx http请求端口	7层	需要证书
6667	tcp	media_server注册到cluster使用的端口,如果media_server和cluster不在同一个局域网,则需要防火墙开放.如果在同一个局域网则无需防火墙开放	4层	无
6668	tcp	提供其他cluster节点rpc的接口,如果多个master不在同一个局域网,则需要防火墙开放.如果在同一个局域网则无需防火墙开放	4层	无

内部端口(不需开放防火墙或者映射到公网的端口)

9650	tcp	gateway监听接收nginx反向代理的请求的端口
80	http	nginx http请求端口
6669	tcp	avdcluster服务接收avdadmin rpc请求的端口
9050	tcp	avd_admin接收rest api请求使用的端口
9010	tcp	supervisord 本地服务管理程序监听端口
2379		etcd监听端口,用于服务注册和服务发现

master管理系统配置

绑定media_server

授权成功后,才可以绑定media_server进行。master可以绑定多个media_server,每个media_server的数据库和录制文件存放地址是独立的,master起调度作用。media_server的安装请参考[安装运维说明](#)。

操作步骤:

1. 登陆平台管理系统,点击右上角的**管理按钮**,选择**平台管理**,在**平台管理--机器管理**页面,点击**“新增机器”**,可以添加需要绑定的媒体服务器



2. 名字、box编号、密码均可以自定义

新增机器 ×

* 名字

* box编号

* 密码

标签

描述

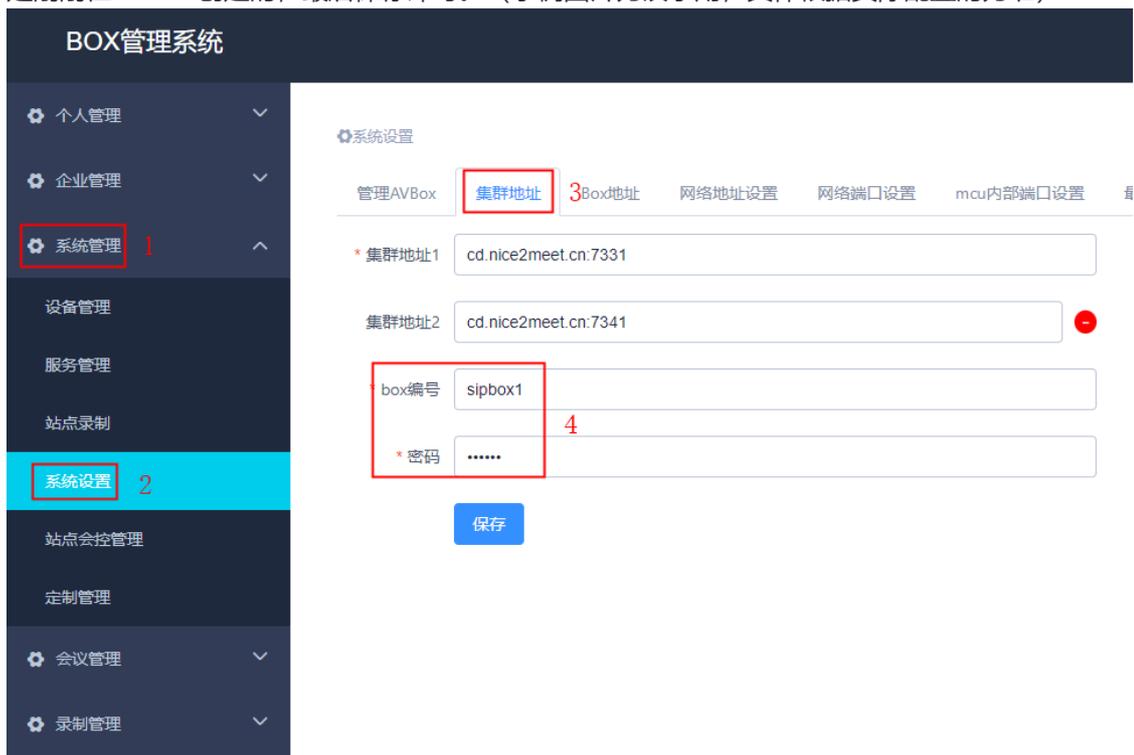
在线状态 是 否

3. 完成绑定后，master的绑定配置就ok了，需要去media_server机器配置。media_server和master的通信端口是6667/tcp，需要开放。

4. 绑定master的media_server，不需要单独授权。登录media_server管理系统，在**系统管理--系统设置**页面，在**管理AVmedia_server**页面，将**是否为单机版**的选项去掉，然后保存。



5. 然后，在**master地址**页面，添加master的连接地址，一般是http端口，比如这里的7331即是master的http端口。多master环境，一般两个master的地址都需要配置。然后填写box编号，就是刚刚在master创建的，最后保存即可。（示例图片为演示用，具体根据实际配置的为准）



6. 然后在**系统管理--服务管理**页面，停止所有服务。

7. 启动link服务，然后在启动mcu和其他服务，这样media_server绑定就完成了。可绑定多个media_server，其他绑定操作也一样。区别box编号即可。
8. master的佳会地址为 `https://server_address:port/r/`，可有通过web进行音视频会议测试下，能进行音视频说明master部署成功。（需要支持webrtc的浏览器，推荐chrome73以上版本）。

小程序支持

小程序支持的配置和media_server版本的一样，登录**平台管理系统**，点击右上角的**管理**按钮，选择**企业管理**，在**系统管理--直播管理**页面，点击**新增**进行配置。



SDK站点管理

点击右上角“管理”按钮，选择“平台管理”，在“**站点管理-sdk站点管理**”页面，可以新增连接服务器凭证。



cpu阈值设置

点击右上角“管理”按钮，选择“平台管理”，在“**系统设置-集群设置-cpu阈值设置**”页面，可以调整cpu阈值，资源占用达到相应阈值后加会、混屏/桌面录制、直播、设备邀请、创建播放器等功能会提示CPU使用率高，拒绝服务。这个设置针对media_server的。

集群切集群切换部署换部署

集群（录制、直播）功能配置

常见异常排除和处理方案

常见问题

1. media_server master的存储数据库及视频文件 是共享存放还是每个都是单独的
2. media_server 和avd_platform 都需要初始化是吧
3. 这个能调整成不需要root吗?
答：不能。原因有以下几点：1.安装数据库驱动(mysql/oracle/pgslq)等相关依赖；2.拷贝master服务本身依赖的动态库(主要是安装包的lib64目录下)；3.创建开机自启动服务需要root权限

更新说明

- master_1.0.1-20210930-21-02-49.tar
 - 修复master链接Oracle数据库和mysql数据库，文件授权不能使用的问题；
- [master/master_1.0.1-20211015-15-43-20.tar](#)
 - master支持国密
 - master支持多maser部署，即master可以部署2个节点，用于保障可靠性
- master_1.0.1-20211216-16-28-43.tar
 - 修复cluster非root用户安装的问题
 - 修复http请求重定向https的问题