



迁移运维管理平台项目 V2.3.0 升级方案

麒麟软件有限公司

2025年 8 月13 日

版本说明

日期	版本号	发布说明	编写者
2025.8.13	V0.1	初版编写	王冕
2025.12.4	V0.2	新增高可用升级	王冕
2025.12.24	V0.3	根据测试批注进行修改优化	王冕

1. 方案目标.....	1
2. 影响范围.....	1
2.1. 环境信息.....	1
2.2. 物料信息.....	1
2.3. 操作范围.....	1
2.4. 影响范围.....	1
2.5. 注意事项.....	1
3. 实施步骤.....	1
3.1. 准备工作.....	2
3.2. 单机升级工作.....	2
3.3. 分布式高可用升级.....	4
3.4. 新增微服务（可选安装）.....	16
3.5. 检查验证工作.....	17
3.6. Agent 升级.....	18
4. 重点功能验证.....	18

1.方案目标

本方案为迁移运维管理平台 V2.3.0 版本升级方案，目的在于方便用户更好将迁移运维管理平台从 V2.2.1 升至 V2.3.0。

2.影响范围

2.1.环境信息

节点名称	节点数量
单机	1
分布式高可用	3/6/10/若干

2.2.物料信息

Kylin-MaIOP-V2.3.0-alpha*-General-2026****-X86_64.iso

Kylin-MaIOP-V2.3.0-alpha*-General-2026****-X86_64.sha256

Kylin-MaIOP-V2.3.0-alpha*-General-2026****-ARM64.iso

Kylin-MaIOP-V2.3.0-alpha*-General-2026****-ARM64.sha256

2.3.操作范围

单机部署所含组件

分布式高可用所含组件

2.4.影响范围

无

2.5.注意事项

该文档仅适用于单节点升级和分布式高可用升级

3.实施步骤

此次升级分为 4 个环节，分别为准备工作、升级工作、检查验证工作和 agent 升级工作。

3.1.准备工作

1)准备升级物料，将该方案对应物料上传至系统，然后挂载到/mnt，升级完成后可卸载。

2)确认平台运行正常；

3.2.单机升级工作

升级工作包括数据备份、软件包升级、重启服务、确保服务正常等步骤。

3.2.1.数据导出及备份

3.2.1.1.数据导出

```
# su - postgres
#若使用了运维模块则操作
pg_dump -U postgres -f ismp.sql ismp
#若使用了 centos 迁移模块则操作
pg_dump -U postgres -f ismp_centos.sql ismp_centos
#若使用了监控模块则操作
pg_dump -U postgres -f ismp_monitor.sql ismp_plugin_monitor
exit
```

3.2.1.2.数据备份

```
# systemctl stop ismp-*
systemctl stop mqbroker
systemctl stop mqnamesrv
systemctl stop postgresql.service
rm -rf /data/pgsql/data_b
cp -raf /data/pgsql/data /data/pgsql/data_2.2.1
```

3.2.2.软件包升级

3.2.2.1.版本升级

```
#进入挂载 2.3.0 ISO 的目录，执行 install.sh
# cd /mnt
```

```
./install.sh
```

注意：执行上述脚本只会升级运维 ismp-service 相关包，若需可选组件 centos 迁移、监控等功能根据脚本输出进行选择升级/安装。如下图所示：

```
=====
  可选软件包安装菜单
=====
  1. 仅安装CentOS迁移组件 ismp-centos
  2. 仅安装监控组件 ismp-monitor
  3. 仅安装热补丁组件 ismp-hotpatch
  4. 仅安装OS迁移组件 ismp-os-upgrade
  5. 仅安装安全加固组件 ismp-security-reinforce
  6. 安装所有组件(1+2+3+4+5)
  7. 不安装其它组件,退出可选安装
=====
请选择安装组合 (1-7): 1
```

```
chown
```

```
root:root
```

```
/opt/ismg-manager/html/kylin-manager/{client-last-agent,iso,kylin_ecsm_activation.s
ock,script,monitor} -R
```

3.2.3.数据导入

```
# systemctl daemon-reload
systemctl stop ismp-*
su - postgres
#若使用了运维模块则操作
dropdb -U postgres ismp
createdb -U postgres ismp
psql -U postgres -d ismp -f ismp.sql
#若使用了 centos 迁移模块则操作
dropdb -U postgres ismp_centos
createdb -U postgres ismp_centos
psql -U postgres -d ismp_centos -f ismp_centos.sql
#若使用了监控模块则操作
dropdb -U postgres ismp_plugin_monitor
createdb -U postgres ismp_plugin_monitor
psql -U postgres -d ismp_plugin_monitor -f ismp_monitor.sql
#退出
exit
```

3.2.4.启动服务

```
systemctl start mqnamesrv
```

```
systemctl start mqbroker
```

```
systemctl restart ismp-nacos-standalone
```

```
tail -n10 /opt/ismp-nacos/logs/start.out|grep "Nacos started successfully" #输
```

出 successfully 则代表 nacos 启动成功，执行后续命令，反之等待大约 30 秒再次执行

```
[root@localhost client-last-agent]# tail -n10 /opt/ismp-nacos/logs/start.out|grep "Nacos started successfully"
2025-08-13 16:28:57,098 INFO Nacos started successfully in stand alone mode. use external storage
```

```
systemctl restart ismp-gateway
```

```
systemctl restart ismp-job
```

```
systemctl restart ismp-auth
```

```
systemctl restart ismp-authentication
```

```
systemctl restart kylin-ecsm-server
```

```
systemctl restart ismp-manager
```

#若用到了 centos 迁移则重启

```
systemctl restart ismp-centos
```

#若用到了监控则重启

```
systemctl restart ismp-monitor
```

3.3.分布式高可用升级

前提条件：

- 最新版本 ISO 已挂载到待升级节点的某个目录下，假设为/mnt；
- 平台包存放路径 x86 与 arm 不同，下文以 x86 为例，arm 路径为 KylinManager/packages/v10sp1/aarch64/。

3.3.1.停止服务

在 ansible 安装节点执行命令远程停止：

```
# cd /opt/ansible/ismp-ansible-playbook/
```

```
ansible gateway -i hosts-high-available -m shell -a 'systemctl stop ismp-gateway'
```

```
ansible auth -i hosts-high-available -m shell -a 'systemctl stop ismp-auth'
```

```
ansible authentication -i hosts-high-available -m shell -a 'systemctl stop
```

```
ismp-authentication && systemctl stop kylin-ecsm-server'  
    ansible job -i hosts-high-available -m shell -a 'systemctl stop ismp-job'  
    ansible nacos -i hosts-high-available -m shell -a 'systemctl stop  
ismp-nacos-cluster'  
    ansible manager -i hosts-high-available -m shell -a 'systemctl stop ismp-manager'  
    ansible monitor -i hosts-high-available -m shell -a 'systemctl stop ismp-monitor'  
#如果部署了监控则执行  
    ansible centos -i hosts-high-available -m shell -a 'systemctl stop ismp-centos'
```

#如果部署了 centos 迁移则执行

3.3.2.数据库操作

3.3.2.1.数据导出

数据库主节点操作：

```
# su - postgres  
#若使用了运维模块则操作  
pg_dump -U postgres -f ismp.sql ismp  
#若使用了 centos 迁移模块则操作  
pg_dump -U postgres -f ismp_centos.sql ismp_centos  
#若使用了监控模块则操作  
pg_dump -U postgres -f ismp_monitor.sql ismp_plugin_monitor  
# exit //退出
```

3.3.2.2.数据备份

数据库主从节点均操作，先从后主：

```
systemctl stop postgresql.service  
rm -rf /data/pgsql/data_b  
mv /data/pgsql/data /data/pgsql/data_2.2.1
```

3.3.2.3.导入数据

数据库主节点操作：

```
# cd /mnt  
cp -raf KylinManager/files/data /data/pgsql/
```

```
chown postgres:postgres /data/pgsql/data -R && chmod 700 /data/pgsql/data -R
//赋予权限
```

生成主从数据同步所需证书

```
# sh KylinManager/scripts/outkey.sh
systemctl start postgresql
sudo -u postgres psql
postgres=# CREATE ROLE ismp WITH REPLICATION LOGIN;
postgres=# \q
su - postgres
#若使用了运维模块则操作
dropdb -U postgres ismp
createdb -U postgres ismp
psql -U postgres -d ismp -f ismp.sql
#若使用了 centos 迁移模块则操作
dropdb -U postgres ismp_centos
createdb -U postgres ismp_centos
psql -U postgres -d ismp_centos -f ismp_centos.sql
#若使用了监控模块则操作
dropdb -U postgres ismp_plugin_monitor
createdb -U postgres ismp_plugin_monitor
psql -U postgres -d ismp_plugin_monitor -f ismp_monitor.sql
# exit //退出
```

3.3.2.4.恢复主从同步

数据库主节点操作：

```
# systemctl stop postgresql
# vim /data/pgsql/data/pg_hba.conf
在文件的最后一行添加如下，绿色字体需替换成实际数据库网段
hostssl replication ismp 10.44.43.0/24 cert clientcert=1
# 启动数据库服务
# systemctl start postgresql
systemctl status postgresql
```

数据库从节点操作:

```
# cd /data/pgsql/
rm -rf /data/pgsql/data
rsync -azvP root@10.44.43.174:/data/pgsql/.ssl .
```

注意: 10.44.43.174 需换成实际的主节点 ip

切换到 postgresql 用户

```
# su - postgres
```

```
# pg_basebackup -h 10.44.43.173 -p 5432 -U ismp -D /data/pgsql/data -v -P
--checkpoint=fast -d "sslmode=verify-ca sslcert=/data/pgsql/.ssl/client.crt
sslkey=/data/pgsql/.ssl/client.key sslrootcert=/data/pgsql/.ssl/ca.crt"
```

#10.44.43.173 需换成实际的主节点 ip

编辑 recovery.conf 文件, 写入下列框中内容, 其中绿色字体 ip 需更换成实际主节点 ip

```
# vim /data/pgsql/data/recovery.conf
```

```
standby_mode = 'on' # 指明从库身份
primary_conninfo = 'host=10.44.43.173 port=5432 user=ismp sslmode=verify-ca
sslcert=/data/pgsql/.ssl/client.crt sslkey=/data/pgsql/.ssl/client.key
sslrootcert=/data/pgsql/.ssl/ca.crt'
recovery_target_timeline = 'latest' # 同步到最新数据
```

```
# exit
```

```
systemctl restart postgresql
```

```
systemctl status postgresql
```

3.3.2.5. 查看主从同步状态

```
# ps -ef|grep -v grep|grep wal
```

主节点如下图: 存在 sender 进程且状态为 streaming

```
[root@localhost ~]# ps -ef|grep -v grep |grep wal
postgres 38733 38728 0 14:43 ? 00:00:00 postgres: wal writer process
postgres 39022 38728 0 14:43 ? 00:00:01 postgres: wal sender process ismp 10.44.179.131(45356) streaming 0/C14DFE8
```

从节点如下图: 存在 receiver 进程且状态为 streaming

```
[root@localhost ~]# ps -ef|grep -v grep |grep wal
postgres 37075 37067 0 14:43 ? 00:00:04 postgres: wal receiver process streaming 0/C150A78
```

3.3.3.缓存操作

3.3.3.1.停止服务

```
# systemctl stop redis-sentinel
systemctl stop redis
```

3.3.3.2.版本升级

```
# mv /etc/redis /etc/redis221
# sh KylinManager/scripts/redis.sh -a 2 -m redis1
```

注意：redis1 为三节点中的节点 1 的 ip

3.3.4.消息队列 EMQX

3.3.4.1.安装 emqx

三节点安装 emqx，若原环境高可用是 6 节点，emqx 节点不可与前端节点共用；若比 6 节点多优先选择部署到 rocketmq 节点。

```
# sh KylinManager/scripts/emqx.sh 本机 ip
```

3.3.4.2.构建集群

在 emqx1,2 上分别加入 emqx3，构建 emqx 集群，加入集群命令如下，ip 为 emqx3 的 ip:

```
# /opt/emqx/bin/emqx_ctl cluster join emqx@emqx3_ip
```

查看集群状态:

```
# /opt/emqx/bin/emqx_ctl cluster status
```

3.3.5.配置与注册中心 Nacos

备份:

```
# cd /opt/ismp-nacos/conf
# cp -ra cluster.conf cluster.confbak
# cp -ra application.properties application.propertiesbak
# cd /mnt
```

nacos 包升级，java 版本升级:

```
# rpm -Uvh KylinManager/packages/ismp/ismp-nacos* --force
```

```
rpm -Uvh KylinManager/packages/ismp/java-17-openjdk-*
```

```
openjdk=`rpm -qa|grep java-17-openjdk-17|head -n1`
```

```
alternatives --set java /usr/lib/jvm/$openjdk/bin/java
```

```
alternatives --set javac /usr/lib/jvm/$openjdk/bin/javac
```

还原配置:

```
# cd /opt/ismp-nacos/conf
```

```
mv cluster.confbak cluster.conf
```

```
mv application.propertiesbak application.properties
```

启动服务:

```
# systemctl daemon-reload
```

```
systemctl start ismp-nacos-cluster
```

```
systemctl status ismp-nacos-cluster
```

3.3.6. 网关操作

注意: 下面命令中的 vip 为前端节点 vip

备份:

```
# cd /opt/ismp-gateway
```

```
cp -ra bootstrap.yml bootstrap.ymlbak
```

```
cp -ra gateway-2.2.1-18.ky10.jar gateway-2.2.1-18.ky10.jarbak
```

gateway 包, java 版本升级:

```
# cd /mnt 进入 iso 挂载目录, 假设为/mnt
```

```
rpm -Uvh KylinManager/packages/ismp/ismp-gateway*
```

```
rpm -Uvh KylinManager/packages/ismp/java-17-openjdk-*
```

```
openjdk=`rpm -qa|grep java-17-openjdk-17|head -n1`
```

```
alternatives --set java /usr/lib/jvm/$openjdk/bin/java
```

```
alternatives --set javac /usr/lib/jvm/$openjdk/bin/javac
```

修改配置, 启动服务:

```
# sed -i 's#127.0.0.1#vip#g' /opt/ismp-gateway/bootstrap.yml
```

任选一台网关节点执行配置修改且推送到 nacos:

```
# cp -ra KylinManager/files/nacos_configs/ismp-gateway.yaml /tmp
```

```
vim /tmp/ismp-gateway.yaml
```

修改其中的 ip 地址、账号、密码等，与之前保持一致即可

```
# sh KylinManager/scripts/change_nacos_config.sh vip ismp-gateway
```

启动所有网关节点服务：

```
# systemctl start ismp-gateway
```

```
systemctl status ismp-gateway
```

等待大约 30S 再次查看日志无报错则启动正常，反之异常。

3.3.7.用户操作

注意：下面命令中的 vip 为前端节点 vip

备份：

```
# cd /opt/ismp-auth
```

```
cp -ra bootstrap.properties bootstrap.propertiesbak
```

```
cp -ra ismp-auth-2.2.1-18.ky10.jar ismp-auth-2.2.1-18.ky10.jarbak
```

auth 包，java 版本升级：

```
# cd /mnt 进入 iso 挂载目录，假设为/mnt
```

```
rpm -Uvh KylinManager/packages/ismp/ismp-auth-*
```

```
rpm -Uvh KylinManager/packages/ismp/java-17-openjdk-*
```

```
openjdk=`rpm -qa |grep java-17-openjdk-17 |head -n1`
```

```
alternatives --set java /usr/lib/jvm/$openjdk/bin/java
```

```
alternatives --set javac /usr/lib/jvm/$openjdk/bin/javac
```

修改配置，启动服务：

```
# sed -i 's#127.0.0.1#vip#g' /opt/ismp-auth/bootstrap.properties
```

任选一台用户管理节点执行配置修改且推送到 nacos：

```
# cp -ra KylinManager/files/nacos_configs/ismp-auth.properties /tmp
```

```
vim /tmp/ismp-auth.properties
```

修改其中的 ip 地址、账号、密码等，与之前保持一致即可

```
# sh KylinManager/scripts/change_nacos_config.sh vip ismp-auth
```

启动所有用户节点服务：

```
# systemctl start ismp-auth
```

```
systemctl status ismp-auth
```

等待大约 30S 再次查看日志无报错则启动正常，反之异常。

3.3.8.授权操作

注意：下面命令中的 vip 为前端节点 vip

备份：

```
# cd /opt/ismp-authentication
```

```
cp -ra bootstrap.properties bootstrap.propertiesbak
```

```
cp -ra ismp-authentication-service-2.2.1-18.ky10.jar
```

```
ismp-authentication-service-2.2.1-18.ky10.jarbak
```

authentication 包，java 版本升级：

```
# cd /mnt 进入 iso 挂载目录，假设为/mnt
```

```
rpm -Uvh KylinManager/packages/ismp/ismp-authentication*
```

```
rpm -Uvh KylinManager/packages/ismp/java-17-openjdk-*
```

```
openjdk=`rpm -qa|grep java-17-openjdk-17|head -n1`
```

```
alternatives --set java /usr/lib/jvm/$openjdk/bin/java
```

```
alternatives --set javac /usr/lib/jvm/$openjdk/bin/javac
```

```
cp -ra KylinManager/scripts/auth_check_mount.sh /opt/ismp-authentication/
```

```
sed -i 's#1.1.1.1#vip#g' /opt/ismp-authentication/auth_check_mount.sh
```

修改配置，启动服务：

```
# sed -i 's#127.0.0.1#vip#g' /opt/ismp-authentication/bootstrap.properties
```

```
cp -ra KylinManager/files/nacos_configs/ismp-authentication-service.properties
```

```
/tmp
```

```
# vim /tmp/ismp-authentication-service.properties
```

修改其中的 ip 地址、账号、密码等，与之前保持一致即可

```
# sh KylinManager/scripts/change_nacos_config.sh vip
```

```
ismp-authentication-service
```

```
systemctl start kylin-ecsm-server
```

```
systemctl start ismp-authentication
```

```
systemctl status ismp-authentication
```

等待大约 30S 再次查看日志无报错则启动正常，反之异常。

3.3.9.任务操作

注意：下面命令中的 vip 为前端节点 vip

备份:

```
# cd /opt/ismp-job
cp -ra bootstrap.properties bootstrap.propertiesbak
cp -ra ismp-job-admin-2.2.1-3.ky10.jar ismp-job-admin-2.2.1-3.ky10.jarbak
```

job 包, java 版本升级:

```
# cd /mnt    进入 iso 挂载目录, 假设为/mnt
rpm -Uvh KylinManager/packages/ismp/ismp-job*
rpm -Uvh KylinManager/packages/ismp/java-17-openjdk-*
openjdk=`rpm -qa|grep java-17-openjdk-17|head -n1`
alternatives --set java /usr/lib/jvm/$openjdk/bin/java
alternatives --set javac /usr/lib/jvm/$openjdk/bin/javac
```

修改配置, 启动服务:

```
# sed -i 's#127.0.0.1#vip#g' /opt/ismp-job/bootstrap.properties
```

任选一台任务管理节点执行配置修改且推送到 nacos:

```
# cp -ra KylinManager/files/nacos_configs/ismp-job.properties /tmp
vim /tmp/ismp-job.properties
```

修改其中的 ip 地址、账号、密码等, 与之前保持一致即可

```
# sh KylinManager/scripts/change_nacos_config.sh vip ismp-job
```

所有任务管理节点启动服务:

```
systemctl start ismp-job
systemctl status ismp-job
```

等待大约 30S 再次查看日志无报错则启动正常, 反之异常。

3.3.10.运维后端

注意: 下面命令中的 vip 为前端节点 vip

备份:

```
# cd /opt/ismp-manager
cp -ra bootstrap.properties bootstrap.propertiesbak
cp -ra ismp-manager-service-2.2.1-18.ky10.jar
ismp-manager-service-2.2.1-18.ky10.jarbak
```

manager 包, java 版本升级:

```
# cd /mnt    进入 iso 挂载目录，假设为/mnt
rpm -Uvh KylinManager/packages/ismp/ismp-service*
rpm -Uvh KylinManager/packages/ismp/java-17-openjdk-*
openjdk=`rpm -qa|grep java-17-openjdk-17|head -n1`
alternatives --set java /usr/lib/jvm/$openjdk/bin/java
alternatives --set javac /usr/lib/jvm/$openjdk/bin/javac
cp -ra KylinManager/scripts/auth_check_mount.sh /opt/ismp-manager/
sed -i 's#1.1.1.1#vip#g' /opt/ismp-manager/auth_check_mount.sh
```

修改配置，启动服务：

```
# cd /opt/ismp-manager/
sed -i 's#127.0.0.1#vip#g' bootstrap.properties
```

下面 sed 替换仅在第 2 台后端节点执行：

```
# sed -i 's#managerA#managerB#g' bootstrap.properties
```

任选一台运维后端节点执行配置修改且推送到 nacos：

```
# cd /mnt/KylinManager/files/nacos_configs/
cp -ra ismp-manager.properties /tmp
vim /tmp/ismp-manager.properties
```

修改其中的 ip 地址、账号、密码等，rocketmq 也需根据实际情况修改

```
# cd /mnt
sh KylinManager/scripts/change_nacos_config.sh vip ismp-manager
```

所有运维后端启动服务：

```
systemctl start ismp-manager
systemctl status ismp-manager
```

等待大约 30S 再次查看日志无报错则启动正常，反之异常。

3.3.11.前端

rpm 包升级：

```
# cd /mnt    进入 iso 挂载目录，假设为/mnt
rpm -Uvh KylinManager/packages/ismp/ismp-web*
```

如果部署了监控模块则执行命令升级监控前端包：

```
# rpm -Uvh KylinManager/packages/ismp-plugin-monitor-web*
```

如果部署了 CentOS 迁移模块则执行命令升级迁移前端包:

```
# rpm -Uvh KylinManager/packages/ismp-centos-web*
```

确认是否存在同步服务:

```
# systemctl status vip-repo-sync.service
```

先确认是否有上面服务运行, 如没有则执行下面命令, 反之不执行:

kill V2.2 版本运行的同步进程:

```
# ps -ef|grep vip_monitor.sh|grep -v grep|awk '{print $2}'|xargs kill -9
```

```
# ps -ef|grep rsync_inotify.sh|grep -v grep|awk '{print $2}'|xargs kill -9
```

确保同步进程已完全 kill:

```
# ps -ef|grep vip_monitor.sh
```

```
# ps -ef|grep rsync
```

```
# cp -ra KylinManager/scripts/vip_repo_sync.sh /opt/ismp-manager/scripts/
```

```
# #cp -ra KylinManager/scripts/repo_mirr_rsync.py /opt/ismp-manager/scripts/
```

```
#cp -ra KylinManager/files/vip-repo-sync.service /lib/systemd/system/
```

```
# systemctl daemon-reload
```

```
# systemctl restart vip-repo-sync
```

```
# systemctl status vip-repo-sync
```

拷贝 agent 介质, 仅在主节点操作即可:

```
# cd /opt/ismp-manager/html/kylin-manager/iso
```

```
mkdir bak
```

```
mv * bak/
```

```
cp -ra /mnt/KylinManager/client/client-last-* .
```

```
cp -ra /mnt/KylinManager/client/centos-last-x86_64.iso .
```

```
cp -ra /mnt/KylinManager/client/kyreplace-* . #若使用 centos 迁移则执行
```

```
cd ../client-last-agent
```

```
mkdir bak
```

```
mv * bak/
```

```
cp -ra /mnt/KylinManager/client/ismp-agent-last-* .
```

```
cd ../script
```

```
mkdir bak
```

```
mv * bak/
```

```
cp -ra /mnt/KylinManager/scripts/register.sh .
```

```
cp -ra /mnt/KylinManager/scripts/kyreplace_install.sh . #若使用 centos 迁移  
则执行
```

```
cp -ra /mnt/KylinManager/scripts/migrate_register.sh . #若使用 centos 迁移则  
执行
```

```
cp -ra /mnt/KylinManager/files/emqx_stream.conf /etc/nginx/conf.d/stream/  
vim /etc/nginx/conf.d/stream/emqx_stream.conf
```

修改其中的 ip 地址为实际 emqx 节点的 ip

然后重启 nginx 服务

```
# systemctl restart nginx
```

```
systemctl status nginx
```

3.3.12. 监控后端（可选升级）

注意：下面命令中的 vip 为前端节点 vip

备份：

```
# cd /opt/ismp-plugin-monitor/  
cp -ra config.yaml config.yamlbak
```

monitor 包版本升级：

```
# cd /mnt 进入 iso 挂载目录，假设为/mnt
```

```
rpm -Uvh KylinManager/packages/ismp/ismp-plugin-monitor-2*
```

修改配置，启动服务：

```
# vim /opt/ismp-plugin-monitor/config.yaml
```

任选一台监控后端节点执行配置修改且推送到 nacos：

```
# cp -ra KylinManager/files/nacos_configs/ismp-monitor.yaml /tmp  
vim /tmp/ismp-monitor.yaml
```

修改其中的 ip 地址、账号、密码等，与之前保持一致即可

```
# sh KylinManager/scripts/change_nacos_config.sh vip ismp-monitor
```

所有监控后端启动服务：

```
systemctl start ismp-monitor
```

```
systemctl status ismp-monitor
```

等待大约 30S 再次查看日志无报错则启动正常，反之异常。

3.3.13.CentOS 迁移后端（可选升级）

注意：下面命令中的 **vip** 为前端节点 **vip**

备份：

```
# cd /opt/ismp-centos/
```

```
cp -ra bootstrap.yml bootstrap.ymlbak
```

centos 包、java 版本升级：

```
# cd /mnt 进入 iso 挂载目录，假设为/mnt
```

```
rpm -Uvh KylinManager/packages/ismp/ismp-centos-2*
```

```
rpm -Uvh KylinManager/packages/ismp/java-17-openjdk-*
```

```
openjdk=`rpm -qa|grep java-17-openjdk-17|head -n1`
```

```
alternatives --set java /usr/lib/jvm/$openjdk/bin/java
```

```
alternatives --set javac /usr/lib/jvm/$openjdk/bin/javac
```

修改配置，启动服务：

```
# sed -i 's#127.0.0.1#vip#g' /opt/ismp-centos/bootstrap.yml
```

任选一台迁移后端节点执行配置修改且推送到 nacos：

```
# cp -ra KylinManager/files/nacos_configs/ismp-centos.yaml /tmp
```

```
vim /tmp/ismp-centos.yaml
```

修改其中的 **ip** 地址、账号、密码等，与之前保持一致即可

```
# sh KylinManager/scripts/change_nacos_config.sh vip ismp-centos
```

所有迁移后端启动服务：

```
# systemctl start ismp-centos
```

```
systemctl status ismp-centos
```

等待大约 **30S** 再次查看日志无报错则启动正常，反之异常。

3.4.新增微服务（可选安装）

2.3 版本新增 OS 升级、热补丁两个微服务模块，用户可根据自身需求选择对应模块安装，下面将介绍两种模块的安装方法。

下面命令中的 **web_ip** 为前端 **vip**、**psql_vip** 为数据库 **vip**、**redis_ip123** 为 **redis** 三节点 **ip**。

3.4.1.前端安装

在 2 台前端节点操作：根据需求安装对应前端包

```
# cd /mnt
# OS 升级前端包
rpm -Uvh KylinManager/packages/ismp/ismp-os-upgrade-web*
# 热补丁前端包
rpm -Uvh KylinManager/packages/ismp/ismp-hotpatch-web*
# 配置 nfs 挂载
vim /etc/exports
```

根据已有内容格式，将 OS 升级(若使用)ip 写入，然后保存退出。

```
exportfs -r
```

3.4.2.OS 升级安装

```
# cd /mnt
sh KylinManager/scripts/ismp-os-upgrade.sh -a 2 -w web_ip -p psq1_vip -r
"redis_ip1 redis_ip2 redis_ip3"
```

3.4.3.热补丁安装

```
# cd /mnt
sh KylinManager/scripts/ismp-hotpatch.sh -a 2 -w web_ip -p psq1_vip -r
"redis_ip1 redis_ip2 redis_ip3"
```

前端主节点操作：

```
mkdir -p /opt/ismp-manager/html/kylin-manager/hotpatch/
cp -rf /mnt/KylinManager/client/kylin-warm*
/opt/ismp-manager/html/kylin-manager/hotpatch/
```

3.5.检查验证工作

退出原登录界面，重新登录 web 界面，可正常登录进入且界面无报错，然后可随机选择一台状态连接的主机进行立即扫描，如成功则功能正常。



3.6.Agent 升级

界面访问 <https://ip>，待界面可以访问时，通过界面 agent 升级将 agent 升级至最新版本 2.3.0-*。升级后 agent 可能短暂掉线，等待大约 2 分钟左右 agent 会显示连接。

4.重点功能验证

登入平台后随机选择一台状态连接的主机，点击操作->立即扫描，若扫描成功则功能正常。