银河麒麟服务器迁移运维管理平台 V2.2.1

产品安装手册

麒麟软件有限公司

2024 年 5 月

目录

银河麒麟服务器迁移运维管理平台介绍

欢迎您使用银河麒麟服务器迁移运维管理平台！

麒麟软件主要面向通用和专用领域打造安全创新操作系统产品和相应解决方案，以安全创新操作系统技术为核心，现已形成银河麒麟服务器操作系统、桌面操作系统、嵌入式操作系统、麒麟云、操作系统增值产品为代表的产品线。

银河麒麟服务器迁移运维管理平台是基于国产银河麒麟高级服务器操作系统开发的服务器操作系统迁移和运维管理平台，通过直观清晰易理解的 Web 界面，帮助系统管理员在大规模、集群式服务器主机管理场景中完成高效运维管理工作，打造系统主机管理、配置管理、自定义脚本、漏洞修复、补丁管理、监控告警、迁移管理等多种核心运维场景解决方案。其易用性强、运行稳定，支持高可用与分布式部署方案，实现关键操作全流程闭环管理，能够在坚实的基础之上提升服务器操作系统的可靠性、稳定性、易用性。

银河麒麟服务器迁移运维管理平台基于信创软硬件技术，支持鲲鹏、飞腾、海光、兆芯等主流 CPU 架构，兼容中标麒麟高级服务器操作系统 V7.0、银河麒麟高级服务器操作系统 V10 等主流操作系统版本，支持在物理机，虚拟机和云环境中部署与配置。平台聚焦主机管理、系统配置、补丁升级、监控告警、迁移管理五个领域，实现了大规模主机管理、运维脚本执行、精细化的系统配置管理、精准化智能化的补丁管理与漏洞修复、系统 SP 升级包推送、可视化系统监控告警、一键式系统迁移等功能，对主机迁移、监控、配置、补丁进行全流程闭环管理，有效地解决了批量主机迁移难度高、配置管理困难、补丁包安装繁琐等问题，打破大规模主机运维的瓶颈，助力操作系统运维管理增质提效，为政府、金融、电力、医疗、运输、制造业等行业用户提供了高效、稳定的服务。

银河麒麟服务器迁移运维管理平台具有以下关键产品特性：

- **提供清晰易理解的操作界面**

  使用 Web 界面完成系统运维操作，降低了系统运维的门槛，运维人员无需在终端中逐台操作。

- **支持大规模的服务器操作系统管理**

支持大规模主机批量注册，通过批次对主机分组并实施批量操作，可查看单台主机的软件包列表、软件包变更记录、配置项列表、配置差异以及配置项对比信息。

- **支持自动化的漏洞发现与补丁安装**

提供漏洞修复智能化一站式操作流程，兼容不同主机的漏洞修复差异，降低补丁安装难度。支持补丁安装回退机制，在异常场景下可快速恢复，实现补丁安装全流程闭环管理。

- **支持精细化的系统配置管控**

提供全流程闭环的大规模主机配置变更方案，实现更快的、一致的、可重复的系统配置管理。

- **支持统一的可视化的系统监控告警**

提供统一实时的可视化监控图表，可以随时掌握系统运行的状态，并可进行告警配置，进行自动化告警，其代理端的安装卸载与运维代理进行解耦，保障用户可根据实际需求灵活装卸。

- **支持一键式迁移管理**

针对业务系统新增、扩容和不变场景，提供一键系统评估、迁移、备份功能，实现系统最小风险、最低成本的替换。

- **提供安全稳定、性能高效、弹性部署、兼容性佳的系统迁移运维平台**

支持操作与更改的历史记录查看，以进行故障排除或日志审计。通过加密通信协议、敏感信息加密脱敏处理等方式保障平台数据的安全性。支持万级规模主机管理，响应速度快，支持按需动态扩容。兼容 Intel、海光、兆芯、飞腾、鲲鹏架构生态，以及主流服务器操作系统，如中标麒麟高级服务器操作系统 V7.0、银河麒麟高级服务器操作系统 V10 SPX。

文档约定标识

【界面上的文本】或**【屏幕、窗口中的按钮】**

在 GUI 界面屏幕或窗口中的标题、词汇、或短语、菜单选项等会用全角方括号“【 】”括起来。它用来标明某个 GUI 屏幕或 GUI 屏幕上的某个元素（譬如与复选框或字段相关的文本）。例如：请点击【**确定**】按钮等。

可替代的文字

用在例子中的文本，如使用这种*斜体*方式，表明该文本应被用户提供的数据所代替。

命令

带字符边框的表明是命令。

 窍门：即一些有用的信息、小技巧等；

 重要：提示请您需要格外重视的内容；

 注记：提醒您关注的事项、注释；

 警告：警示信息，告诫您采取或防止哪些操作；

 小心：情况可能稍有复杂，请您谨慎操作。

1 硬件环境

1.1 单机部署

- 服务端：支持 x86_64、aarch64，推荐配置 16CPU 16G 内存，1T 数据盘(/opt)；
- 代理端：支持 x86_64、aarch64，最低配置 CPU >= 4、存储 >= 50G、内存 >= 4G。

1.2 多机部署

- 服务端、代理端均支持 x86_64、aarch64；
- 服务端：

  前端服务(ismp-web)：推荐配置 8CPU 16G 内存，1T 数据盘(/opt)；

  运维服务(ismp-manager)：推荐配置 8CPU 16G 内存，1T 数据盘(/opt)；

  消息队列服务(RocketMQ)：推荐配置 8CPU 16G 内存，500G 数据盘(/opt)；

  数据库服务(PostgreSQL)：推荐配置 8CPU 16G 内存，500G 数据盘(/data)；

  缓存服务（Redis）：推荐配置 8CPU 16G 内存，500G 数据盘(/data)；

  注册与配置中心(ismp-nacos)：推荐配置 8CPU 16G 内存，500G 数据盘(/opt)；

  网关(ismp-gateway)：推荐配置 8CPU 16G 内存，500G 数据盘(/opt)；

  任务管理服务(ismp-job)：推荐配置 8CPU 16G 内存，500G 数据盘(/opt)；

  用户服务服务(ismp-auth)：推荐配置 8CPU 16G 内存，500G 数据盘(/opt)；

  授权服务(ismp-authentication)：推荐配置 8CPU 16G 内存，500G 数据盘(/opt)；

  CentOS 迁移服务(ismp-centos)：推荐配置 8CPU 16G 内存，500G 数据盘(/opt)；

  监控服务(ismp-monitor)：推荐配置 8CPU 16G 内存，500G 数据盘(/opt)，实际规

划时可参考《监控服务的硬件规划补充说明》；

- 代理端：最低配置 CPU >= 4、存储 >= 50G、内存 >= 4G；

1.3 监控服务的硬件规划补充说明

监控服务硬件规划主要关注内存和磁盘两个方面,下表为模拟测试情况供硬件规划参考。

表 1-1 监控服务模拟测试情况

监控代理接入量	CPU 建议	实际内存占用	实际磁盘占用
2000	8C	9G	1.5T
3000	8C	12G	2T
6000	8C	22G	4T
9000	16C	32G	6.5T

注记：以上数据为模拟测试数据，项目在硬件规划时，请充分做好空间预留；另该数据为采集频率 15s、数据保留时长 15 天下的结果，如采集频率和数据保留时长变更，容量需进行线性的增扩和缩减。

2 环境规划与设置

2.1 软件环境

- 服务端：支持银河麒麟高级服务器操作系统 V10（SP1 0711/SP1 0319/SP1 0518/SP2/SP3/兼容版 8.2）；
- 运维代理端：支持中标麒麟高级服务器操作系统 V7、银河麒麟高级服务器操作系统 V10（SP1 0711/SP1 0319/SP1 0518/SP2/SP3/兼容版 8.2/8.8）、银河麒麟云底座操作系统 V10 2309、CentOS 7.0~7.9。
- 监控代理端：支持中标麒麟高级服务器操作系统 V7、银河麒麟高级服务器操作系统 V10（SP1 0711/SP1 0319/SP1 0518/SP2/SP3/兼容版 8.2/8.3）、银河麒麟云底座操作系统 2309。
- 原机迁移代理端：支持 CentOS/RHEL/Oracle Linux 6.0~6.10、Centos/RHEL/Oracle Linux7.0~7.9、CentOS 8.0~8.5、Oracle Linux 8.0~8.2、RHEL 8.0~8.8。

2.2 网络环境

- -必须保证每个网卡有一个固定的 IP 地址，该地址用于对外提供相应的服务；
- -服务端与代理端网络需要能够正常通信；
- -平台后端与其它中间件之间的网络时延应保持在 10ms 以下；
- -平台后端与其他中间件之间的系统时间应保持一致且均为北京时间，建议使用 ntp 保持系统时间同步。

2.3 操作系统环境

- 正确配置服务器系统的网络环境，包括 IP、NetMask、GateWay、DNS 等；
- 建议关闭系统防火墙、关闭系统 SELinux 安全防护措施，如果开启请确认以下端口可以正常访问:

表 1-2 单机部署开放端口要求

服务器角色	开放端口	功能
服务端	9000	平台提供的软件源服务，用于补丁下发、软件包安装升级、SP 升级操作
	443	https 的端口，用于代理端安装注册
	8848	nacos 的反向代理
	9876 10911	mq 服务
	22	上传评估报告
	873	软件包仓库同步
代理端	22	界面注册 ssh 连接
	9100	监控 agent

表 1-3 分布式部署开放端口要求

服务器角色	开放端口	功能
前端服务	9000	平台提供的软件源服务，用于补丁下发、软件包安装升级、SP 升级操作
	22	用于上传迁移评估报告
	80	web 的端口，用于代理端安装注册
	873	软件包仓库同步

	443	https 的端口，用于代理端安装注册
	2049、111、20048	nfs 共享存储
运维服务	8080	http 的端口，用于代理端安装注册
	8899	ismp-job 执行器
数据库服务	5432	数据库连接
缓存服务	6379	redis 连接
消息队列服务	9876 10909 10911 10912	mq 服务
注册与配置中心	8848	web 服务
	7848	Raft-rpc 通信
	9848、9849	grpc 通信
任务管理服务	8083	web 服务
网关	8888	web 服务
用户服务	8085 8086	用户服务
授权服务	8892 8893	授权服务
CentOS 迁移服务	9999 8889	Centos 迁移服务
监控服务	8088 9010	监控服务
代理端	22	界面注册 ssh 连接
	9100	监控 agent

表 1-4 高可用部署开放端口要求

服务器角色	开放端口	功能
前端服务	9000	平台提供的软件源服务，用于补丁下发、软件包安装升级、SP 升级操作
	22	用于上传迁移评估报告
	80	web 的端口，用于代理端安装注册
	443	https 的端口，用于代理端安装注册
	873	软件仓库同步
	8093	ismp-job 的反向代理端口

	8848	nacos 的反向代理
	2049、111、20048	nfs 共享存储
	9848	grpc 通信
运维服务	111	nfs 共享存储
	8080	http 的端口，用于代理端安装注册
	8899	ismp-job 执行器
数据库服务	5432	数据库连接
缓存服务	6379	redis 连接
	26379	哨兵模式
消息队列服务	9876 30911 40911 40912 40913 30909	mq 服务
注册与配置中心	8848	web 服务
	7848	Raft-rpc 通信
	9848、9849	grpc 通信
任务管理服务	8083	web 服务
网关	8888	web 服务
用户服务	8085 8086	用户服务
授权服务	8892 8893 111	授权服务
CentOS 迁移服务	9999 8889	Centos 迁移服务
监控服务	8088 9010	监控服务
代理端	22 9100	界面注册 监控

3 银河麒麟服务器迁移运维管理平台部署

3.1 单机部署

将镜像挂载到一个目录文件下（以/mnt 为例），

```
[root@server1 ~]#mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

进入挂载的目录中，执行 install.sh 命令，运行安装脚本：

```
[root@server1 ~]# cd /mnt
[root@server1 mnt]# ./install.sh
[root@server1 mnt]#vim /etc/rc.local
```

nohup python3 前加上 sleep 30 &&，然后保存退出。

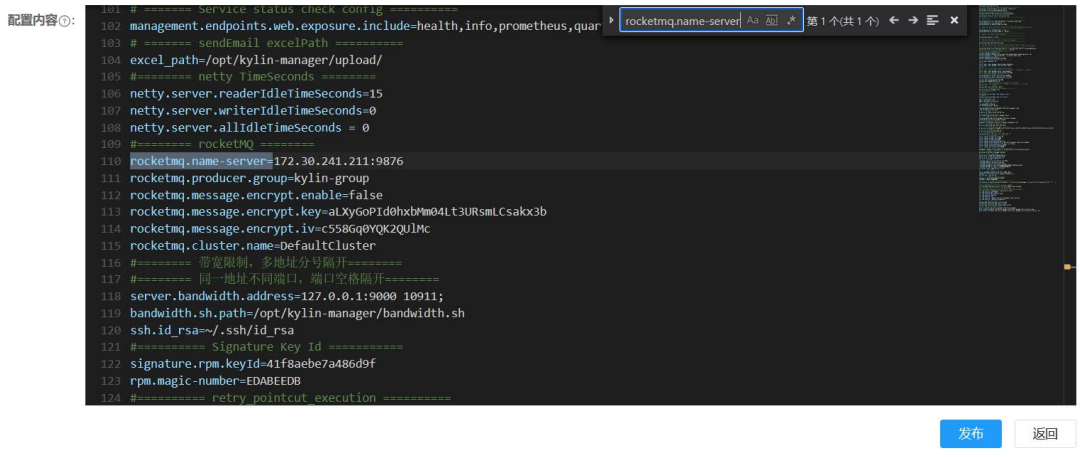
```
[root@localhost repo]# cat /etc/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
sleep 30 && nohup python3 /opt/ismp-manager/scripts/repo_mirr_rsync.py > /opt/ismp-manager/logs/repo_mirr/nohup.out 2>&1 &
sh /opt/ismp-manager/html/kylin-manager/script/nacos_success.sh
```

使用带宽限制功能还需按照 3.3 章节进行配置。

然后使用浏览器访问 http://localhost:8848/nacos，将 localhost 修改为实际部署 ismp-nacos 的 IP 地址。Nacos 平台账号为 nacos，密码默认为 Qwer!234578。若该密码不符合安全要求，用户可根据 3.5 章节进行密码修改。

在配置列表中编辑 ismp-manager 的配置，修改其中的部分配置(鼠标点击配置内容框任意处,然后使用 Ctrl+f 查找,如下图所示,若无反应,可刷新页面重新操作)host.web.ip、shiro-redis.redis-manager.host、spring.redis.host、rocketmq.name-server、server.bandwidth.address 为实际 ip，然后点击发布。修改完成后，重启 ismp-manager。




```
[root@server1 mnt]# systemctl restart ismp-manager
```

若不使用迁移功能可不用修改 ismp-centos 的配置，也无需启动

ismp-centos 服务。

在配置列表中编辑 ismp-centos 的配置，修改其中的部分配置(鼠标点击配置内容框任意处,然后使用 Ctrl+f 查找,如下图所示,若无反应,可刷新页面重新操作)host.web.ip、rocketmq 下的 name-server、redis 下的 host 为实际 ip，然后点击发布。修改完成后，重启 ismp-centos。

```
#systemctl restart ismp-centos
```

如需卸载请执行：

```
[root@server1 mnt]# ./uninstall.sh
```

3.2 分布式部署

3.2.1 主机配置

分布式多机单台部署每个组件一台机器，共 12 台：

缓存 redis

数据库 postresql

消息队列 rocketmq

注册中心和配置中心 ismp-nacos

网关 ismp-gateway

任务管理 ismp-job

用户 ismp-auth

授权 ismp-authentication

前端 ismp-web

监控 ismp-monitor

迁移 ismp-centos

后端 ismp-manager

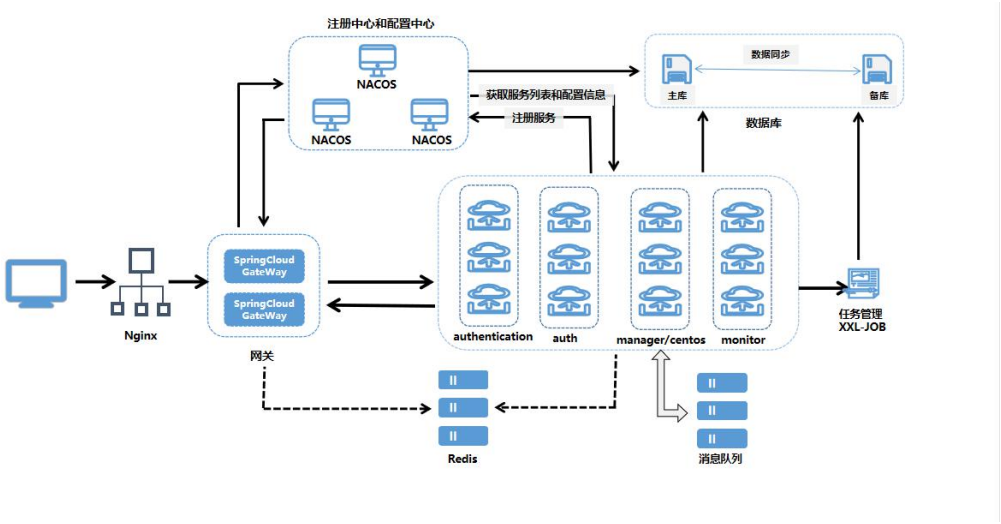


图 2-1 架构图

3.2.2 Redis 部署

注记：如果需要配置 redis 主从，请按照第 4 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行下面命令：

```
# cd /mnt
# sh KylinManager/scripts/redis.sh
```

如需卸载 redis 请执行：

```
#systemctl stop redis
#yum remove redis -y
```

3.2.3 PostgreSQL 部署

注记：如果需要配置 postgresql 主备，请按照第 5 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行下面命令：

```
# cd /mnt/
# sh KylinManager/scripts/postgresql.sh
#systemctl restart postgresql
```

如需卸载数据库请执行：

```
#systemctl stop postgresql
#yum remove postgresql-server postgresql -y
```

3.2.4 RocketMQ 部署

注记：如果需要配置 rocketmq 高可用，请按照第 6 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名，执行以下命令：

```
# cd /mnt/
# sh KylinManager/scripts/rocketmq.sh
```

如需卸载 rockermq 请执行：

```
#systemctl stop mqbroker
#systemctl stop mqnamesrv
#yum remove rocketmq -y
```

3.2.5 ismp-nacos 部署

注记：如果需要配置 ismp-nacos 集群模式，请按照第 7 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名；执行以下命令，其中 psql_ip 为数据库 ip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-nacos.sh -p psql_ip
```

然后使用浏览器访问 http://localhost:8848/nacos，将 localhost 修改为实际部署 ismp-nacos 的 IP 地址。Nacos 平台账号为 nacos、密码为 Qwer!234578。若该密码不符合安全要求，用户可根据 3.5 章节进行密码修改。

进入【配置管理】-【配置列表】，依次编辑下列配置,修改完成后点击发布。(全部修改完成以后再进行后面的部署)

- ismp-gateway 中所有的 127.0.0.1 换成实际部署的 redis(2 处)、rocketmq(1 处)的 ip 地址；
- ismp-job 中的 localhost 换成数据库 IP 地址；
- ismp-auth 中的 127.0.0.1、localhost 换成 redis(2)、postgresql(1)、ismp-job(1)的 ip；
- ismp-authentication-service 中的 127.0.0.1、localhost 换成 redis(1)、psql(2)、ismp-job(1)的 ip；
- ismp-centos 中的 127.0.0.1、localhost 换成 redis(1)、job(1)、mq(1)、web(1)、psql(2)的 ip；
- ismp-manager 中的 127.0.0.1、localhost 换成 redis(2)、job(1)、mq(1)、web(1)、psql(2)的 ip；

如需卸载 ismp-nacos 请执行：

```
#systemctl stop ismp-nacos-standalone
#yum remove ismp-nacos -y
```

3.2.6 ismp-gateway 部署

注记：如果需要配置 ismp-gateway 集群模式，请按照第 9 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行以下命令，其中 nacos_ip 为部署 ismp-nacos 的 ip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-gateway.sh -n nacos_ip
```

如需卸载 ismp-gateway 请执行：

```
#systemctl stop ismp-gateway
#yum remove ismp-gateway -y
```

3.2.7 ismp-job 部署

注记：如果需要配置 ismp-job 集群模式，请按照第 10 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行以下命令，其中 nacos_ip 为部署 ismp-nacos 的 ip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-job.sh -n nacos_ip
```

如需卸载 redis 请执行：

```
#systemctl stop ismp-job
#yum remove ismp-job -y
```

3.2.8 ismp-auth 部署

注记：如果需要配置 ismp-auth 集群模式，请按照第 11 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行以下命令，其中 nacos_ip 为部署 ismp-nacos 的 ip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-auth.sh -n nacos_ip
```

如需卸载请执行：

```
# systemctl stop ismp-auth
# yum remove ismp-auth -y
```

3.2.9 ismp-web 部署

注记：如果需要配置前端 nginx 高可用，请按照第 8 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行以下命令,其中 gateway_ip 为部署 ismp-gateway 的 ip：

```
# cd /mnt
# sh KylinManager/scripts/web.sh -g gateway_ip
#vim /etc/rc.local
nohup python3 前加上 sleep 30 &&
```

```
[root@localhost repo]# cat /etc/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
sleep 30 && nohup python3 /opt/ismp-manager/scripts/repo_mirr_rsync.py > /opt/ismp-manager/logs/repo_mirr/nohup.out 2>&1 &
sh /opt/ismp-manager/html/kylin-manager/script/nacos_success.sh
```

限制挂载前端共享目录的 IP 地址：

```
# echo "/opt/ismp-manager/html/kylin-manager/ 10.1.110.97(rw,sync,no_root_squash) 10.1.110.98(rw,sync,no_root_squash) 10.1.110.99(rw,sync,no_root_squash)" > /etc/exports
# exportfs -r
```

注记：上面命令中的 ip 地址要更改为实际部署 ismp-manager、ismp-centos、ismp-authentication 的 IP 地址，顺序随机。

然后可以在三台机器中随机选择一台进行测试，使用下面命令验证是否可以正常挂载：

```
# mount -t nfs web_ip:/opt/ismp-manager/html /opt/ismp-manager/html/
# df -h  //查看共享目录是否挂载
```

注记：web_ip 要替换成实际的前端服务器 ip。

重要：如果使用 centos 迁移模块功能，请注释掉 /etc/nginx/nginx.conf 中的 56、57、58 行内容，然后重启 nginx。

```
# systemctl restart nginx
```

如需卸载请执行：

```
# systemctl stop nginx
# yum remove ismp-web nginx -y
```

3.2.10 ismp-authentication 部署

注记：如果是高可用部署，请按照第 12 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行以下命令，其中 web_ip、nacos_ip 为分别为部署前端、ismp-nacos 的 ip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-authentication.sh -w web_ip -n nacos_ip
```

如需卸载请执行：

```
# systemctl stop ismp-authentication
# yum remove ismp-authentication -y
```

3.2.11 ismp-monitor 部署(可选安装)

注记：如果需要配置监控 monitor 高可用，请按照第 13 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行以下命令，其中 psql_ip、nacos_ip、redis_ip 分别为数据库、ismp-nacos、redis 的 ip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-monitor.sh -p psql_ip -n nacos_ip -r redis_ip
```

3.2.12 ismp-CentOS 部署(可选安装)

注记：如果需要配置迁移 CentOS 高可用，请按照第 14 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行以下命令，其中 web_ip、nacos_ip 为分别为前端、ismp-nacos 的 ip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-centos.sh -w web_ip -n nacos_ip
```

3.2.13 ismp-manager 部署

注记：如果需要配置后端高可用，请按照第 15 章节操作。

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行以下命令，其中 web_ip、nacos_ip 为前端、ismp-nacos 的 ip：

```
# cd /mnt
# sh KylinManager/scripts/manager.sh -w web_ip -n nacos_ip
```

如需卸载请执行：

```
# systemctl stop ismp-manager
# yum remove ismp-service -y
```

3.3 配置后端与前端免密(带宽限制，仅首次部署时需要操作)

在后端机器操作，如果有多台后端，在所有后端机器均需要操作：

```
# ssh-keygen -t rsa -N '' -f /root/.ssh/id_rsa -q
# mkdir -p /home/kylin/.ssh/ && chmod 755 /home/kylin/.ssh
# cp /root/.ssh/id_rsa* /home/kylin/.ssh/
# chown kylin:kylin /home/kylin/.ssh -R
# ssh-keygen -p -f /home/kylin/.ssh/id_rsa -m pem
# ssh-copy-id -f -i /home/kylin/.ssh/id_rsa.pub 前端 ip
```

3.4 端口修改

如果客户要求修改默认的 443、80、8080 端口，则修改下面文件中对应的端口。

含 80、443 端口的文件：

/etc/nginx/nginx.conf

#systemctl restart nginx

只含 443 端口的文件：

/opt/ismp-manager/html/kylin-manager/script/register.sh

/opt/ismp-manager/html/kylin-manager/script/migrate_register.sh

/opt/ismp-manager/html/kylin-manager/script/kyreplace_install.sh

/opt/ismp-manager/html/kylin-manager/script/get_register.sh

Nacos 界面配置管理->配置列表 ismp-manager、ismp-centos 中的 443 端口

#systemctl restart nginx　　//修改完成后重启 nginx

#systemctl restart imsp-manager

#systemctl restart ismp-centos

只含 8080 对应的文件:

/opt/ismp-manager/bootstrap.properties

#systemctl restart ismp-manager　　//重启服务

访问 web 界面时需带上修改后的端口，假设为 444，则浏览器输入 ip:444 即可。

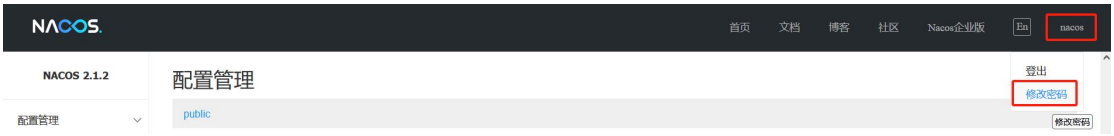
3.5 平台默认密码修改

平台有 pgsql、redis、nacos 密码

3.5.1 Nacos 登录密码修改

单机、分布式、高可用通用

默认为 Qwer!234578，登录后进行修改，如下图所示：



然后在 server 端修改配置文件中的 nacos 密码：

#grep 'Qwer!234578' /opt/ -r|grep -v ismp-nacos

共有 7 个文件需要更改 nacos 对应的密码，且密码不以特殊字符开头

```
[root@localhost opt]# grep 'Qwer!234578' /opt/ -r|grep -v ismp-nacos
/opt/ismp-manager/bootstrap.properties:spring.cloud.nacos.password=Qwer!234578
/opt/ismp-auth/bootstrap.properties:spring.cloud.nacos.password=Qwer!234578
/opt/ismp-authentication/bootstrap.properties:spring.cloud.nacos.password=Qwer!234578
/opt/ismp-centos/bootstrap.yml:      password: Qwer!234578
/opt/ismp-gateway/bootstrap.yml:      password: Qwer!234578
/opt/ismp-plugin-monitor/config.yaml:  password: Qwer!234578
/opt/ismp-plugin-monitor/config.yaml:  password: Qwer!234578
/opt/ismp-job/bootstrap.properties:spring.cloud.nacos.password=Qwer!234578
```

修改完成后需重启所有的服务

systemctl restart ismp-gateway

systemctl restart ismp-job

systemctl restart ismp-auth

systemctl restart ismp-authentication

systemctl restart ismp-monitor

systemctl restart ismp-centos

systemctl restart ismp-manager

3.5.2 Pgsql、redis 密码修改

默认均为 Qwer!234578，如需修改按以下步骤进行

修改 redis 配置文件中的密码：

/etc/redis/redis.conf

/etc/redis/redis-sentinel.conf　　//仅高可用部署需要修改

使用 grep 命令查找需要修改的行：

grep 'Qwer!234578' /etc/redis -rn

修改完成后需重启服务：

systemctl restart redis

systemctl restart redis-sentinel　　//仅高可用部署重启

修改数据库用户 kylin 默认密码，高可用部署时在主节点操作即可：

#sudo -u postgres psql

#\password kylin

#\q

在 nacos 节点修改连接数据库的密码

vim /opt/ismp-nacos/conf/application.properties

修改 kylin 的密码

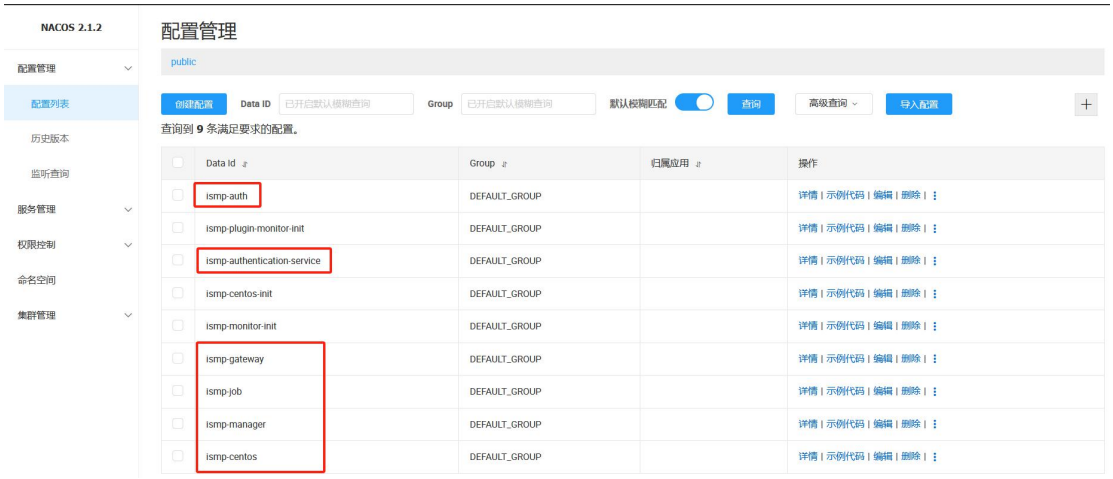
```
37 db.num=1
38 ### Connect URL of DB:
39 db.url.0=jdbc:postgresql://localhost:5432/nacos_config
40 db.user=kylin
41 db.password=Qwer!234578
42
```

修改完成后重启服务

systemctl restart ismp-nacos-standalone　　//单节点

systemctl restart ismp-nacos-cluster　　//高可用

然后登录 nacos 界面，点击配置管理->配置列表，修改配置文件中的默认 pgsql、redis 的默认密码，将 Qwer!234578 全部换成自己修改的密码，然后重启服务即可。如下图所示共有 6 个配置文件需要修改。



如使用监控功能还需在监控所在节点端修改监控的配置文件：

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

修改完成后需重启服务

systemctl restart ismp-monitor

3.5.3 配置文件加密

如果需要对配置文件中涉及的 ip 及端口进行加密，只需要使用加密工具将加密的内容进行转换，加密后的格式为 ENC(xxxxxxxxx)，如下图所示。加密工具已放在平台 iso 中，挂载平台 iso 进入 tools 目录即可使用，使用方式参见 tools/README.txt。

bootstrap.yml 中的部分内容

```
server:
  port: 8888

spring:
  cloud:
    nacos:
      #username: nacos
      username: ENC(7D0/B+K8ruWLVExtd+mrS8FeEl3/ZzFLbQlFIW2CBRj0t10pd3LfI1IrzX+C47DI)
      #password: Qwer!234578
      password: ENC(6JUTg1KV4Xx0KyEN0n3gVyGx3in63/PIl+inTjiqMvilZrdjOFpOWCuVsb+rR5cx)
      config:
        namespace: public
        group: DEFAULT_GROUP
        file-extension: yaml
        #Nacos 地址
        server-addr: 127.0.0.1:8848
        enabled: true
      discovery:
        #Nacos 地址
        server-addr: 127.0.0.1:8848
  application:
    name: ismp-gateway
```

Nacos 界面中 ismp-gateway 中的部分配置示例

```
redis:
  password: Qwer!234578
  maxWaitMillis: 10
  #哨兵模式配置方式      mastername@ip1:port1,ip2:port2,ip3:port3
  #host: mymaster@1.1.1.1:26379,1.1.1.2:26379,1.1.1.3:26379
  #单点模式参数配置方式        ip:port
  #host: 127.0.0.1:6379
  host: ENC(XDlLBDJknhkOq0mMxQvehx+qMfFB+C2tKPwWqy7yUMB0Rczse8a6nxUmQeYYN5VA)
```

Nacos 界面中 ismp-authentication-service 中的部分配置示例

```
# ============================================================
# = Data Source
# ============================================================
#spring.datasource.url=jdbc:postgresql://localhost:5432/ismp
spring.datasource.url=ENC(9Qt1yfPB3P5WzugtH+FpqftlLPvPljyfAFeDgm1T2m002hE0aiz6nFkKM5QCkxq2ak6G10xsapuMzg4IqGttqD1UrMS4441ZHNV/rcBZvlY)
#spring.datasource.username=kylin
spring.datasource.username=ENC(/2DEqLDFzdHdMxyvc3EoZDbs2mjryJ44qAh9FGteX4YCTanRmwCjPe+buBDB3eMp)
#spring.datasource.password=Qwer!234578
spring.datasource.password=ENC(zTWGWYQgq7UxrBCwhlqoIhjFCMuyfBDQx3uo+xeYznel8eNdyJyqC5Vw0X91kZUD)
```

4 Redis 哨兵模式集群环境部署(3 台)

主机配置示例：

172.30.201.51 node1　　//假设 node1 为主

172.30.201.52 node2　　//假设 node2 为备

172.30.201.53 node3　　//假设 node3 为备

4.1 安装 Redis

在三台服务器上安装和配置缓存服务器软件 redis：

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名。

执行以下命令，其中 redis_ipm 为 redis 主节点 ip：

```
# cd /mnt
# sh KylinManager/scripts/redis.sh -a 2 -m redis_ipm
```

4.2 查看集群状态

注记：可在任意一台部署 redis 的机器上查看集群某个节点的信息。

```
# redis-cli -c -h 10.44.43.201 -p 6379 -a 'Qwer!234578' info replication
```

```
[root@localhost mnt]# redis-cli -c -h 10.44.43.201 -p 6379 -a 'Qwer!234578' info replication
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
# Replication
role:master
connected_slaves:2
slave0:ip=10.44.43.203,port=6379,state=online,offset=95195,lag=1
slave1:ip=10.44.43.202,port=6379,state=online,offset=95334,lag=1
master_failover_state:no-failover
master_replid:1e5028b8e7b63af69bd9c4c261a20ef7ed945eb0
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:95334
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1
repl_backlog_histlen:95334
```

图 **3-1** 主节点信息

```
[root@localhost mnt]# redis-cli -c -h 10.44.43.202 -p 6379 -a 'Qwer!234578' info replication
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
# Replication
role:slave
master_host:10.44.43.201
master_port:6379
master_link_status:up
master_last_io_seconds_ago:1
master_sync_in_progress:0
slave_read_repl_offset:99103
slave_repl_offset:99103
slave_priority:100
slave_read_only:1
replica_announced:1
connected_slaves:0
master_failover_state:no-failover
master_replid:1e5028b8e7b63af69bd9c4c261a20ef7ed945eb0
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:99103
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1
repl_backlog_histlen:99103
```

图 **3-2** 从节点信息

5 PostgreSQL 高可用(2 台)

主机配置示例：

172.30.201.160 master　　//假设为主

172.30.201.198 standby　　//假设为备

172.30.201.112　vip　　//此 ip 为漂移 ip

5.1 磁盘调优

lsblk　　//确认/data 目录所在的磁盘标识符，图中/data 目录所在为 sda

```
[root@localhost ~]# lsblk
NAME            MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda               8:0    0  21.8T  0 disk /data
sdb               8:16   0 446.6G  0 disk
├─sdb1            8:17   0   600M  0 part /boot/efi
├─sdb2            8:18   0     1G  0 part /boot
└─sdb3            8:19   0   445G  0 part
  ├─klas-root   253:0    0   391G  0 lvm  /
  ├─klas-swap   253:1    0     4G  0 lvm  [SWAP]
  └─klas-backup 253:2    0    50G  0 lvm
```

若磁盘类型为 NVMe，则不支持调优。

#vim /etc/redis/redis-tuning.sh　　将下面两行内容写入脚本

echo deadline >　/sys/block/sda/queue/scheduler

echo 2048 > /sys/block/sda/queue/nr_requests

注意：将上面命令中的 sda 换成实际的磁盘标识符

#chmod 755 /etc/redis/redis-tuning.sh

nohup sh /ect/redis/redis-tuning.sh &

设置 pg-tuning.sh 脚本的开机执行

chmod +x /etc/rc.d/rc.local

echo "nohup sh /etc/redis/redis-tuning.sh & " >> /etc/rc.d/rc.local

5.2 安装 PostgreSQL

在主备服务器上安装和配置数据库软件 postgresql：

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注：此处 iso 名为代称，部署时需替换成实际 iso 名，执行以下命令：

```
# cd /mnt
# sh KylinManager/scripts/postgresql.sh
#systemctl restart postgresql
```

5.3 创建复制角色

注意：以下 master 为主节点上的操作，standby 为备节点上的操作

```
[root@master ~]# su - postgres
[postgres@master ~]$ createuser --replication -P -e ismp     //此处角色为 ismp，设置 ismp 的密码
```

为新角色输入的口令:

再输入一遍:

```
SELECT pg_catalog.set_config('search_path', '', false)
CREATE ROLE kylin PASSWORD 'md59713c21061910f0a4776c40625668430'
NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN REPLICATION;
```

注记：在以上命令中，-P 标志提示输入新角色的密码，-e 回显 createuser 生成并发送到数据库服务器的命令。

5.4 角色授权

```
[postgres@master ~]$ vim /data/pgsql/data/pg_hba.conf
```

在文件的最下面加上如下条目

```
host    replication     ismp            10.1.1.0/24         md5
```

重要：10.1.1.0 要修改为实际的数据库网段

5.5 启动主数据库

```
[postgres@master ~]$ exit
[root@master ~]# systemctl restart postgresql
```

5.6 配置备数据库

5.6.1 将主数据库数据备份到备机

```
[root@standby ~]# systemctl stop postgresql
[root@standby ~]# tar -zcvf /data/pgsql/data.tar.gz /data/pgsql/data # 做备份
[root@standby ~]# rm -rf /data/pgsql/data/*
[root@standby ~]# su - postgres
[postgres@standby ~]$ pg_basebackup -h 172.30.201.160 -D /data/pgsql/data -U ismp -P -v -R -X stream //输入 5.2 中设置的密码，使用创建的角色 ismp 同步数据
```

```
Password:
pg_basebackup: initiating base backup, waiting for checkpoint to complete
pg_basebackup: checkpoint completed
pg_basebackup: write-ahead log start point: 0/4000028 on timeline 1
pg_basebackup: starting background WAL receiver
23788/23788 kB (100%), 1/1 tablespace
pg_basebackup: write-ahead log end point: 0/40000F8
pg_basebackup: waiting for background process to finish streaming ...
pg_basebackup: base backup completed
```

5.6.2 配置 recovery.conf

```
[postgres@standby ~]$ vim /data/pgsql/data/recovery.conf
```

修改 qwer1234 为 4.2 章节中设置的密码

```
standby_mode = 'on'            # 指明从库身份
primary_conninfo = 'user=ismp password=qwer1234 host=172.30.201.160 port=5432' # 连接到主库信息
recovery_target_timeline = 'latest'          # 同步到最新数据
```

5.6.3 备机重启数据库

```
[postgres@standby ~]$ exit
[root@standby ~]# systemctl restart postgresql
```

5.6.4 查看主服务(WAL 发送器进程)状态

```
[postgres@master ~]$ psql -U postgres -c "\x" -c "SELECT * FROM pg_stat_replication;"
```

```
[postgres@pnode1 ~]$ psql -c "\x" -c "SELECT * FROM pg_stat_replication;"
扩展显示已打开。
-[ RECORD 1 ]----+------------------------------
pid              | 1168854
usesysid         | 16386
usename          | icbc
application_name | walreceiver
client_addr      | 172.30.201.198
client_hostname  |
client_port      | 59782
backend_start    | 2021-11-17 14:14:27.143362+08
backend_xmin     |
state            | streaming
sent_lsn         | 0/B1F4C70
write_lsn        | 0/B1F4C70
flush_lsn        | 0/B1F4C70
replay_lsn       | 0/B1F4C70
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
```

图 **4-1** 主数据库状态

5.6.5 查看从服务(WAL 接收器进程)状态

```
[postgres@standby ~]$ psql -U postgres -c "\x" -c "SELECT * FROM pg_stat_wal_receiver;"
```

```
[postgres@pnode2 ~]$ psql -c "\x" -c "SELECT * FROM pg_stat_wal_receiver;"
扩展显示已打开。
-[ RECORD 1 ]---------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
pid                   | 1121672
status                | streaming
receive_start_lsn     | 0/B000000
receive_start_tli     | 9
received_lsn          | 0/B1F4DC8
received_tli          | 9
last_msg_send_time    | 2021-11-17 15:17:02.16854+08
last_msg_receipt_time | 2021-11-17 15:17:01.893358+08
latest_end_lsn        | 0/B1F4DC8
latest_end_time       | 2021-11-17 15:17:02.16854+08
slot_name             |
conninfo              | user=icbc password=******** dbname=replication host=172.30.201.160 port=5432 fallback_application_name=walrece
iver sslmode=prefer sslcompression=1 krbsrvname=postgres target_session_attrs=any
```

图 **4-2** 从数据库状态

5.7 keepalived 部署

5.7.1 修改 keepalived.conf

主备节点均需要修改配置。

```
[root@master ~]# vim /etc/keepalived/keepalived.conf
```

```
! Configuration File for keepalived
global_defs {
```

```
    router_id master
}

vrrp_script check_pg {
    script "/etc/keepalived/check_pg.sh"      #监控脚本
    interval 2
    weight -20      #keepalived 部署了两台，所以设为 20，如果三台就设为 30
}

vrrp_instance VI_1 {
    state BACKUP          #两台主机都设为 backup 非抢占模式
    interface ens3          #网卡名写自己的，不要照抄
    virtual_router_id 51
    priority 100          #master 设为 100，backup 设为 80，反正要比 100 小
    unicast_src_ip  1.1.1.1          #本机实际 ip
    unicast_peer {
        2.2.2.2                      #对端实际 ip
    }
advert_int 1
    nopreempt                  #设置为非抢占模式必须要该参数
    authentication {
        auth_type PASS
        auth_pass 1111
    }

    track_script {
        check_pg
    }

    virtual_ipaddress {
        10.1.110.101        #虚拟 ip
    }
}
```

```
[root@standby ~]# vim /etc/keepalived/keepalived.conf
```

```
! Configuration File for keepalived
```

```
global_defs {
   router_id master
}

vrrp_script check_pg {
    script "/etc/keepalived/check_pg.sh"      #监控脚本
    interval 2
    weight -20      #keepalived 部署了两台，所以设为 20，如果三台就设为 30
}

vrrp_instance VI_1 {
    state BACKUP          #两台主机都设为 backup 非抢占模式
    interface ens3          #网卡名写自己的，不要照抄
    virtual_router_id 51
    priority 80          #master 设为 100，backup 设为 80，反正要比 100 小
unicast_src_ip   1.1.1.1          #本机实际 ip
    unicast_peer {
        2.2.2.2                      #对端实际 ip
    }
advert_int 1
    nopreempt                  #设置为非抢占模式必须要该参数
    authentication {
        auth_type PASS
        auth_pass 1111
    }

    track_script {
        check_pg
    }

    virtual_ipaddress {
            10.1.110.101          #虚拟 ip
    }
}
```

5.7.2 修改 check_pg.sh

注记：主备均需要修改，只需要将脚本中的默认漂移 ip8.8.8.8 改为

实际漂移 ip；

```
[root@master ~]# vim /etc/keepalived/check_pg.sh
```

```
#!/bin/bash
# 判断 pg 是否活着
A=`ps -C postmaster --no-header | wc -l`

# 判断 vip 浮到哪里，需要更改成自己设置的 vip
B=`ip a | grep 8.8.8.8 | wc -l`

# 判断是否是从库处于等待的状态
C=`ps -ef | grep postgres | grep 'startup process' | wc -l`

# 判断从库链接主库是否正常
D=`ps -ef | grep postgres | grep 'receiver' | wc -l`

# 判断主库连接从库是否正常
E=`ps -ef | grep postgres | grep 'sender' | wc -l`

# 如果 pg 死了，将消息写入日志并且关闭 keepalived
if [ $A -eq 0 ];then
    echo "`date "+%Y-%m-%d--%H:%M:%S"` postgresql stop so vip stop " >> /etc/keepalived/check_pg.log
    systemctl stop keepalived
else

    # 判断出主库挂了，vip 漂移到了从库，提升从的地位让它可读写
    if [ $B -eq 1 -a $C -eq 1 -a $D -eq 0 ];then
        su - postgres -c "pg_ctl promote -D /data/pgsql/data/"    #重新加载 pgsql 使其可写
        echo "`date "+%Y-%m-%d--%H:%M:%S"` standby promote " >> /etc/keepalived/check_pg.log
    fi

    # 判断出自己是主并且和从失去联系
    if [ $B -eq 1 -a $C -eq 0 -a $D -eq 0 -a $E -eq 0 ];then
```

```
            echo "`date "+%Y-%m-%d--%H:%M:%S"` can't find standby " >>
/etc/keepalived/check_pg.log
        fi
    fi
```

5.7.3 启动 keepalived

注记：主备配置好后，均启动服务

```
[root@master ~]# systemctl start keepalived
[root@master ~]# systemctl status keepalived    //查看服务是否正常启动
[root@master ~]# ip a
```

```
[root@pnode1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:f3:cf:eb brd ff:ff:ff:ff:ff:ff
    inet 172.30.201.160/24 brd 172.30.201.255 scope global noprefixroute ens3
       valid_lft forever preferred_lft forever
    inet 172.30.201.112/24 scope global secondary ens3
       valid_lft forever preferred_lft forever
    inet6 fe80::a95f:bea2:834b:6967/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:c1:4c:88:c3 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```

```
[root@pnode2 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:5a:21:b6 brd ff:ff:ff:ff:ff:ff
    inet 172.30.201.198/24 brd 172.30.201.255 scope global noprefixroute ens3
       valid_lft forever preferred_lft forever
    inet6 fe80::2312:952b:fba:73c9/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:b0:9a:3c:85 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```

可以看到漂移 ip 已在主节点上，备节点没有

5.8 故障测试

5.8.1 模拟主库宕机

```
[root@master ~]# systemctl stop postgresql
[root@standby ~]# ip a
```

```
[root@pnode2 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:5a:21:b6 brd ff:ff:ff:ff:ff:ff
    inet 172.30.201.198/24 brd 172.30.201.255 scope global noprefixroute ens3
       valid_lft forever preferred_lft forever
    inet 172.30.201.112/24 scope global secondary ens3
       valid_lft forever preferred_lft forever
    inet6 fe80::2312:952b:fba:73c9/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:b0:9a:3c:85 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```

可以看到此时 vip 已漂移到备机上，迁移运维管理系统管理平台可继续操作,/data/pgsql/data/recovery.conf 文件已经自动变成 recovery.done 文件

5.8.2 原主机恢复

```
[root@master ~]# systemctl stop postgresql.service
[root@master ~]# su - postgres
[postgres@primary ~]$ vim /data/pgsql/data/recovery.conf
```

```
standby_mode='on'
recovery_target_timeline = 'latest'
primary_conninfo='host=172.30.201.198 port=5432 user=ismp password=qwer1234'
```

注记：host 需换成实际原备机 ip，passwd 换成 5.2 中设置的密码

```
[root@master ~]# exit
[root@master ~]# systemctl start postgresql
[root@primary ~]# systemctl restart keepalived
```

此时宕机的主机已完全恢复，原主库成为新备库，漂移 ip 在原备库上，再次发生故障宕机时，按照同上恢复方法即可恢复。多次恢复后 recovery.done 可直接变更为 recovery.conf。

6 RocketMQ 集群模式(3 台)

主机配置示例：

IP	角色	架构模式	对应配置文件

1.1.1.1	nameserver1	master	broker-n0.conf
2.2.2.2	nameserver2	salve1	broker-n1.conf
3.3.3.3	nameserver3	salve2	broker-n2.conf

6.1 安装 rocketmq

在服务器上安装 rocketmq 服务器：

```
#mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名,执行以下命令，其中-n 有 0、1、2 三个值，当在主节点部署时-n 后跟 0,在备 1 节点部署时-n 后跟 1，在备 2 节点部署时-n 后跟 2，-i 后跟三节点 ip：

```
# cd /mnt/
# sh KylinManager/scripts/rocketmq.sh -a 2 -n 0 -i "172.30.23.1 172.30.23.2 172.30.23.3"
```

当三台机器均执行完脚本后，依次启动 mqbroker 服务

```
# systemctl start mqbroker.service
# systemctl status mqbroker     //查看服务是否正常启动
```

如需卸载 rockermq 请执行：

```
#systemctl stop mqbroker
#systemctl stop mqnamesrv
#yum remove rocketmq -y
```

7 ismp-nacos 高可用(3 台)

nacos 高可用部署需要 3 台服务器，需要开放 8848、7848、9848、9849 端口的对外访问权限，以下步骤在 3 台服务器上依次执行。

7.1 安装 ismp-nacos

在服务器上安装和配置 ismp-nacos：

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名，执行下面命令，psql_ip 为数据库 vip，-i 后跟三节点 ip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-nacos.sh -a 2 -p psql_ip -i "172.30.23.1 172.30.23.2 172.30.23.3"
```

如需卸载 ismp-nacos 请执行：

```
#systemctl stop ismp-nacos-cluster
#yum remove ismp-nacos -y
```

7.2 检查服务状态

然后使用浏览器访问 http://localhost:8848/nacos，将 localhost 修改为实际部署 nacos 的一个 IP 地址。Nacos 平台账号为 nacos，密码为 Qwer!234578。

在“集群管理-节点列表”中查看 3 台服务的状态均为“UP”则启动成功。

进入【配置管理】-【配置列表】，依次修改文件中的参数

ismp-gateway 中 redis、rocketmq 部分的 127.0.0.1 换成实际部署的 redis(2 处)、rocketmq(1 处)的 ip 地址

ismp-job 中数据库部分的 localhost 换成实际部署的数据库 IP(1 处)地址

ismp-auth 中 redis、数据库、job 部分的 127.0.0.1、localhost 换成实际部署的 redis(2 处)、postgresql(2 处)、ismp-job(1 处)的 ip

ismp-authentication-service 中 redis、数据库、job 部分的 127.0.0.1、

localhost 换成实际部署的 redis(1 处)、psql(1 处)、ismp-job(1 处)的 ip

ismp-centos 中 redis、job、mq、web、pgsql 部分的 127.0.0.1、localhost 换成实际部署的 redis(1 处)、job(1 处)、mq(1 处)、web(1 处)、psql(2 处)的 ip

ismp-manager 中 redis、job、mq、web、pgsql 部分的 127.0.0.1、localhost 换成实际部署的 redis(2 处)、job(1 处)、mq(1 处)、web(1 处)、psql(2 处)的 ip

8 前端高可用(2 台)

8.1 配置主备免密传输文件

主机 A 配置免密：

```
# ssh-keygen -t rsa -N '' -f /root/.ssh/id_rsa -q
# ssh-copy-id -i /root/.ssh/id_rsa.pub   root@备机 B_ip
```

同理在备机 B 也进行相同的操作配置免密，并将文件拷贝到主机 A。

```
# ssh-keygen -t rsa -N '' -f /root/.ssh/id_rsa -q
# ssh-copy-id -i /root/.ssh/id_rsa.pub   root@主机 A_ip
```

8.2 安装前端环境

将镜像挂载到一个目录文件下（以/mnt 为例），

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

 注记：此处 iso 名为代称，部署时需替换成实际 iso 名，执行以下命令：

```
# cd /mnt
# mkdir -p /opt/ismp-manager && cp -ra KylinManager/files/web_ip.conf /opt/ismp-manager/     //创建目录拷贝配置文件，内容如下，修改 ip1，ip2，ip3,最后三行 ip 之间要用逗号隔开
```

```
[root@localdomain files]# cat web_ip.conf
master_ip=ip1                   #前端本机实际ip地址
backup_ip=ip2                   #前端对端实际ip地址
vip=ip3                         #前端漂移ip地址

job_ip=ip1,ip2                      #部署ismp-job的IP地址,至少两台
nacos_ip=ip1,ip2,ip3                #部署ismp-nacos的IP地址,三台
gateway_ip=ip1,ip2                  #部署ismp-gateway的IP地址,至少两台
```

确认修改无误后再运行下面命令：

```
# sh KylinManager/scripts/web.sh -a 2
# systemctl status nginx
```

使用浏览器访问 http://localhost:8848/nacos，将 localhost 修改为实际部署 nginx 的 ip 地址，Nacos 平台账号为 nacos，密码为 Qwer!234578。可以访问则前端与 nacos 通信正常。

配置 nfs 共享目录：

```
# echo "/opt/ismp-manager/html/kylin-manager/ 10.1.110.99(rw,sync,no_root_squash) 10.1.110.97(rw,sync,no_root_squash) 10.1.110.98(rw,sync,no_root_squash)" > /etc/exports
# exportfs -r
```

注记：当有多个 nfs 客户端时上述命令中的 IP 地址要换成实际的 manager、authentication、centos 地址。

重要：如果需要使用 centos 迁移模块功能，请注释掉 /etc/nginx/nginx.conf 中的 60、61、62 行内容，然后重启 nginx。

```
51      server {
52          #listen       80;
53          #listen       [::]:80;
54          server_name  _;
55          listen       443 ssl;
56          access_log /var/log/nginx/access.log main if=$loggable;
57
58          ssl_certificate      ssl/server.crt;
59          ssl_certificate_key  ssl/server.key;
60  #        ssl_protocols TLSv1.2;
61  #        ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
62  #        ssl_prefer_server_ciphers off;
63
64          location / {
65              root  /opt/ismp-manager/html;
66              try_files $uri $uri/ /index.html;
67          }
68
```

```
# systemctl restart nginx
```

8.3 修改主备 keepalived 配置文件

```
# vim /etc/keepalived/keepalived.conf(修改下面标红的地方)
```

```
! Configuration File for keepalived

global_defs {
   router_id master
}

vrrp_script chk_nfs {
    script "/opt/ismp-manager/scripts/nfs_check.sh"      #监控脚本
    interval 2
    weight -20      #keepalived 部署了两台，所以设为 20，如果三台就设为 30
}

vrrp_instance VI_1 {
    state BACKUP          #两台主机都设为 backup 非抢占模式
    interface ens3          #网卡名写自己的，不要照抄
    virtual_router_id 60
    priority 100        #master 设为 100，backup 设为 90，反正要比 100 小
    unicast_src_ip   1.1.1.1    #本机实际 ip
    unicast_peer {
        2.2.2.2                 #对端实际 ip
    }
advert_int 1
    nopreempt                   #设置为非抢占模式必须要该参数
    authentication {
        auth_type PASS
        auth_pass ismp!234
    }

    track_script {
        chk_nfs
    }

    virtual_ipaddress {
        10.1.110.101        #虚拟 ip
    }
}
```

Slave 节点的 keepalived.conf 配置需将 priority 参数项修改为 90，unicast_src_ip 和 unicast_peer 调换位置，两节点均修改完成后启动 keepalived 服务。

```
# systemctl start keepalived
# systemctl status keepalived
```

8.4 启动脚本(主备传输文件)

```
# nohup sh /opt/ismp-manager/scripts/vip_monitor.sh &
```

设置 vip_monitor.sh 脚本的开机自启动

```
# chmod +x /etc/rc.d/rc.local
# echo "nohup sh /opt/ismp-manager/scripts/vip_monitor.sh & " >> /etc/rc.d/rc.local
```

9 ismp-gateway 高可用部署(2 台)

9.1 部署 ismp-gateway 服务

在服务器上安装和配置 ismp-gateway：

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名，执行以下命令，其中 web_ip 为前端 vip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-gateway.sh -a 2 -w web_vip
#systemctl status ismp-gateway
```

如需卸载请执行：

```
#systemctl stop ismp-gateway
#yum remove ismp-gateway -y
```

9.2 检查服务状态

登录 Nacos 界面，在“服务管理-服务列表”中查看服务状态，查找服务名为 gateway 的服务，其健康实例数与实际启动的服务数量一致即可。点击详情可以查看各节点的健康状态。

IP	端口	临时实例	权重	健康状态	元数据
10.1.167.160	8888	true	1	true	preserved.register.source=SPRING_CLOUD
10.1.167.162	8888	true	1	true	preserved.register.source=SPRING_CLOUD

10 ismp-job 高可用部署(2 台)

10.1 部署多个 ismp-job 服务

在服务器上安装和配置 ismp-job：

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名，执行以下命令，其中 web_ip 为前端 vip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-job.sh -a 2 -w web_ip
```

如需卸载请执行：

```
#systemctl stop ismp-job
#yum remove ismp-job -y
```

10.2 检查服务状态

登录 Nacos 界面，在“服务管理-服务列表”中查看服务状态，查找服务名

为 ismp-job 的服务，其健康实例数与实际启动的服务数量一致即可。点击详情可以查看各节点的健康状态。

IP	端口	临时实例	权重	健康状态	元数据
10.1.167.160	8083	true	1	true	preserved.register.source=SPRING_CLOUD
10.1.167.161	8083	true	1	true	preserved.register.source=SPRING_CLOUD

11 ismp-auth 高可用部署(2 台)

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名，执行以下命令，其中 web_ip 为前端 vip：

```
# cd /mnt
# sh KylinManager/scripts/ismp-auth.sh -a 2 -w web_ip
```

如需卸载请执行：

```
# systemctl stop ismp-auth
# yum remove ismp-auth -y
```

12 ismp-authentication 高可用部署(1 台)

将镜像挂载到一个目录文件下（以/mnt 为例），

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名，执行以下命

令，其中 web_ip 为前端 vip。

```
# cd /mnt
# sh KylinManager/scripts/ismp-authentication.sh -a 2 -w web_ip
```

如需卸载请执行：

```
# systemctl stop ismp-authentication
# yum remove ismp-authentication -y
```

13 ismp-monitor 高可用部署(2 台)

将镜像挂载到一个目录文件下（以/mnt 为例），

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名，其中 psql_ip、web_ip、redis_ip 分别为数据库 vip、前端的 vip、缓存 ip。

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

14 ismp-CentOS 高可用部署(2 台)

将镜像挂载到一个目录文件下（以/mnt 为例），

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名；

执行以下命令，其中 web_ip 为前端 vip：

```
[root@server1 ~]# cd /mnt
[root@server1 mnt]# sh KylinManager/scripts/ismp-centos.sh -a 2 -w web_ip
```

15 ismp-manager 高可用部署(2 台)

将镜像挂载到一个目录文件下（以/mnt 为例），

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
```

注记：此处 iso 名为代称，部署时需替换成实际 iso 名，执行以下命令，其中 web_ip 为前端 vip。

```
# cd /mnt
# sh KylinManager/scripts/manager.sh -a 2 -w web_ip
# df -h   //查看是否挂载成功,如果显示未挂载，可手动挂载：
# mount -t nfs 3.3.3.3:/opt/ismp-manager/html /opt/ismp-manager/html/
```

注记：3.3.3.3.要替换成第 8 章中实际的前端服务器 vip。

16 迁移运维管理平台快捷部署

本方案为迁移运维管理平台自动部署方案，使用 ansible 工具实现平台的快捷部署。基本思路是通过在 ansible 管理节点上运行部署脚本，向其他目标节点部署对应的模块，因此本方案需要预先准备一台主机安装 ansible 工具。

16.1 安装 Ansible

注意：关闭所有节点防火墙

#systemctl stop firewalld

将镜像挂载到一个目录文件下（以/mnt 为例），

```
# mount -o loop Kylin-Easyclick-V2.2.iso /mnt
# cd /mnt
# sh KylinManager/scripts/ansible.sh
# ssh-copy-id -o StrictHostKeyChecking=no ip     //ip 为所有的目标节点，拷贝管理节点 ssh 公钥到所有目标节点上
#scp Kylin-Easyclick-V2.2.iso ip:/root   //将 iso 拷贝到所有目标节点
```

16.2 部署迁移运维管理平台

16.2.1 单机部署

修改 playbook 中的参数：

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

修改 hosts-standalone 文件中的 ip 为实际部署迁移运维管理平台的主机 ip：

vim standalone/hosts-standalone

修改 group_vars/all 文件中变量：

vim group_vars/all

iso_path 为迁移运维管理平台镜像文件在目标节点上的实际绝对路径；

mount_path 为镜像文件在目标节点上的挂载目录；

port1 修改默认 443

port2 修改默认 80

port3 修改默认 8080

进入 standalone 目录下执行部署命令：

cd standalone

ansible-playbook -i hosts-standalone standalone.yml

16.2.2 多机部署

修改 playbook 中的参数：

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

修改 hosts-multi-machine 文件:所有 ip 为对应模块待部署节点 ip;

#vim hosts-multi-machine

修改 group_vars/all 文件:

vim group_vars/all

1. iso_path 为迁移运维管理平台镜像文件在目标节点上的实际绝对路径；
2. mount_path 为镜像文件在目标节点上的挂载目录；
3. port1 修改默认 443
4. port2 修改默认 80
5. port3 修改默认 8080
6. ismp_centos_install 是否安装使用 ismp-centos 组件，True 或者 False
7. ismp_monitor_install 是否安装使用 ismp-monitor 组件，True 或者

False

8. nginx_machine_password 是部署 nginx 的主机密码；

ansible-playbook -i hosts-multi-machine multi-machine.yml

16.2.3 高可用部署

修改 playbook 中的参数：

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

修改 hosts-high-available 文件:所有 ip 为对应模块待部署节点 ip;

#vim hosts-high-available

修改 group_vars/all 文件:

vim group_vars/all

1. iso_path 为迁移运维管理平台镜像文件在目标节点上的实际绝对路径；
2. mount_path 为镜像文件在目标节点上的挂载目录；
3. port1 修改默认端口 443
4. port2 修改默认端口 80
5. port3 修改默认端口 8080
6. ismp_centos_install 是否安装 ismp-centos 组件，True 或者 False
7. ismp_monitor_install 是否安装 ismp-monitor 组件,True 或者 False
8. psql_vip: 192.168.59.160 数据库 vip
9. psql_ismp_password: "QWer1234$#@!" # 数据库用户 ismp 密码
10. psql_segment: 192.168.59.0/24 # 数据库主机所在网段
11. postgresql_mas_network_card_name: ens33 # 部署数据库主机网卡名称
12. postgresql_bak_network_card_name: ens33 # 部署数据库备机网卡名称
13. nginx_mas_network_card_name: ens33 # 部署前端 nginx 主机网卡名称
14. nginx_bak_network_card_name: ens33 # 部署前端 nginx 备

机网卡名称

15. nginx_vip: 192.168.59.161 # 前端 nginx 的 vip

16. nginx_mas_machine_password: "QWer1234$#@!" # 部署前端 nginx 主机密码

17. nginx_bak_machine_password: "QWer1234$#@!" # 部署前端 nginx 备机密码

ansible-playbook -i hosts-high-available high-available.yml

17 漏洞解决方案

17.1 目标主机 showmount -e 信息泄露(CVE-1999-0554)

解决方法：

在 nfs 服务端(所有前端)机器上开启防火墙，只对 nfs 客户端(所有后端)主机开放。

在前端机器上操作:

```
# systemctl start firewalld
# firewall-cmd --new-zone=mountd --permanent
#firewall-cmd --zone=mountd --add-service=mountd --permanent
# firewall-cmd --zone=mountd --add-source=10.1.110.97 --permanent
# firewall-cmd --reload
```

注记：当开启防火墙时，该机器所跑的服务对应的 2.3 中涉及到的端口需要开放。