

iModel 企业级数据分析平台部署手册

银河麒麟 V11（2503）· x86-64 · 单机部署

适用系统	银河麒麟高级服务器操作系统 V11（2503）
处理器架构	x86-64（不支持 ARM 架构）
Java 运行环境	OpenJDK 1.8.x
元数据库	MySQL 8.0
反向代理	Nginx（麒麟系统仓库版本）
手册版本	v3.0 · 2026 年 9 月

深圳市卓越方达科技有限公司 · www.imodel.org.cn · 400-8568-196

目录

1. 部署前须知
2. 一、基础环境
3. 二、MySQL 8.0 元数据库
4. 三、主服务首次初始化
5. 四、Nginx 反向代理
6. 五、注册开机自启动
7. 六、进程守护
8. 七、部署验证
9. 八、日常运维
10. 九、常见问题

部署前须知

本手册适用于在一台银河麒麟 V11（2503）x86-64 服务器上，单机部署 iModel 企业级数据分析平台：元数据库、主服务、执行器与反向代理均安装在同一台服务器。数据库与应用分开部署时，请按各节中的「异机部署」提示调整。

全部命令以 `root` 账号执行。第二章下载 MySQL 安装包需要服务器可以访问互联网；无法联网的环境，请提前把安装包拷贝到服务器上。

组件与端口

组件	端口	作用	是否需要对外开放
Nginx 反向代理	9000	用户浏览器访问入口	是
iModel 主服务	8080	应用服务，由 Nginx 转发	否
实时推送服务	51020	WebSocket 连接，由 Nginx 转发	否
FTP 文件服务	4646	平台内部文件传输	否
Executor 执行器	50000	workflow 执行	否
Modern 服务	59100	workflow 运行环境	否
MySQL 元数据库	3306	平台元数据存储	否（异机部署时仅对应用服务器开放）

说明：端口号为软件包默认值。若在配置文件中调整过端口，以实际配置为准；部署完成后可用 `ss -tlnp | grep java` 查看各 Java 进程实际监听的端口。

部署顺序

1. 第一章 基础环境：系统确认、Java、防火墙、SELinux、解压软件包
2. 第二章 MySQL 元数据库：安装、初始化、导入表结构、配置连接
3. 第三章 主服务首次初始化：补充缺失的配置文件（首次部署必做）
4. 第四章 Nginx 反向代理
5. 第五章 注册开机自启动
6. 第六章 进程守护
7. 第七章 部署验证
8. 第八章 日常运维；第九章 常见问题

一、基础环境

1.1 确认操作系统与处理器架构

```
cat /etc/os-release | grep -E '^(NAME|VERSION)='  
uname -m
```

bash

预期结果： `uname -m` 输出 `x86_64`。如果输出 `aarch64`，说明是 ARM 服务器，iModel 目前不支持，请停止部署。

1.2 安装 OpenJDK 1.8

```
dnf install -y java-1.8.0-openjdk java-1.8.0-openjdk-devel  
  
# 如系统预装了 JDK 11 或 17，切换默认版本为 1.8  
alternatives --config java  
  
# 配置系统级环境变量  
cat > /etc/profile.d/java.sh << 'EOF'  
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk  
export PATH=$JAVA_HOME/bin:$PATH  
EOF  
source /etc/profile.d/java.sh  
  
java -version  
jar 2>&1 | head -1
```

bash

预期结果： `java -version` 显示 `1.8.0_xxx`；`jar` 命令可用（第三章会用到）。

1.3 配置防火墙

用户只通过 Nginx 的 9000 端口访问平台，其余端口仅在本机组件之间通信，不需要对外开放。

```
firewall-cmd --zone=public --permanent --add-port=9000/tcp  
firewall-cmd --reload  
firewall-cmd --zone=public --list-ports
```

bash

不要把 3306、8080 等内部端口对公网开放。**异机部署**时，只在数据库服务器上对应用服务器的 IP 开放 3306：

```
firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="应用服务器IP" port  
rt port="3306" protocol="tcp" accept'
```

1.4 检查 SELinux

```
getenforce
```

bash

输出为 `Disabled` 或 `Permissive` 时跳过本节。输出为 `Enforcing` 时，需要允许 Nginx 监听 9000 端口并转发到本机服务，否则 Nginx 会启动失败或返回 502：

```
dnf install -y policycoreutils-python-utils
semanage port -a -t http_port_t -p tcp 9000
setsebool -P httpd_can_network_connect 1
```

bash

1.5 解压软件包并赋权

```
cd /opt
tar -zxvf IModel-bundle.tar.gz

# 软件包中的主服务脚本名为 imodelabi-server.sh，本手册统一重命名为 imodel-server.sh
mv /opt/IModel-bundle/bin/imodelabi-server.sh /opt/IModel-bundle/bin/imodel-server.sh

chmod +x /opt/IModel-bundle/bin/*.sh \
        /opt/IModel-bundle/executor/*.sh \
        /opt/IModel-bundle/ftpserve/*.sh
```

bash

二、MySQL 8.0 元数据库

说明：本章在数据库服务器上执行；单机部署即为同一台服务器。MySQL 8.0 已于 2026 年 4 月结束官方支持，新项目如需使用其他版本，请先确认与当前软件包的兼容性。

2.1 下载并安装

```
mkdir -p /opt/mysql-rpm && cd /opt/mysql-rpm
wget https://mirrors.tuna.tsinghua.edu.cn/mysql/downloads/MySQL-8.0/mysql-8.0.40-1.el9.x86_64.rpm-
bundle.tar
tar -xvf mysql-8.0.40-1.el9.x86_64.rpm-bundle.tar

# 删除调试符号包，否则 dnf 会因依赖缺失而安装失败
rm -f *debuginfo*.rpm

# 移除与 MySQL 冲突的 MariaDB 库
dnf remove -y mariadb-libs 2>/dev/null || true

dnf install -y mysql-community-common-*.el9.x86_64.rpm \
               mysql-community-client-plugins-*.el9.x86_64.rpm \
               mysql-community-libs-*.el9.x86_64.rpm \
               mysql-community-client-*.el9.x86_64.rpm \
               mysql-community-icu-data-files-*.el9.x86_64.rpm \
               mysql-community-server-*.el9.x86_64.rpm
```

bash

说明：镜像站可能下架旧补丁版本。下载失败时，打开镜像站目录查看当前可用的 8.0.x 版本号，替换命令中的版本号即可。

2.2 配置并启动

编辑 `/etc/my.cnf`，在 `[mysqld]` 段落中加入：

```
[mysqld]
port=3306
character-set-server=utf8mb4
collation-server=utf8mb4_unicode_ci
# 单机部署只监听本机；异机部署改为数据库服务器的内网 IP
bind-address=127.0.0.1
```

ini

```
systemctl enable --now mysqld
systemctl is-active mysqld
```

bash

2.3 修改 root 密码并创建平台账号

密码要求：MySQL 默认密码策略要求至少 8 位，并同时包含大写字母、小写字母、数字和特殊字符。平台账号的密码会写入数据库连接串，**不要使用 @ : / ? # % & 等字符**，否则平台会连接失败；特殊字符建议使用英文句点 `.` 或下划线 `_`。

```
TEMP_PASS=$(grep 'temporary password' /var/log/mysqld.log | awk '{print $NF}')
mysql -uroot -p"${TEMP_PASS}" --connect-expired-password
```

bash

在 MySQL 命令行中执行（把示例密码替换为符合要求的强密码）：

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'Root_Pass.2026';

CREATE DATABASE imodel CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

-- 单机部署：平台通过 127.0.0.1 连接
CREATE USER 'imodel'@'127.0.0.1' IDENTIFIED BY 'Imodel_Pass.2026';
GRANT ALL PRIVILEGES ON imodel.* TO 'imodel'@'127.0.0.1';
-- 异机部署：把 127.0.0.1 换成应用服务器的 IP

FLUSH PRIVILEGES;
EXIT;
```

sql

2.4 导入表结构

```
mysql -uroot -p imodel < /opt/IModel-bundle/db/i-model.sql

# 验证：能看到表清单
mysql -uroot -p -e "USE imodel; SHOW TABLES;" | head
```

bash

2.5 配置平台的数据库连接

编辑 `/opt/IModel-bundle/app/config/datart.conf`：

```
# ===== datasource config =====
# 必须写 127.0.0.1, 不要写 localhost (localhost 会走本地套接字, 与账号授权不匹配)
datasource.ip=127.0.0.1
datasource.port=3306
datasource.database=imodel
datasource.username=imodel
datasource.password=Imodel_Pass.2026

# ===== datart config =====
# 必填项。填写用户浏览器访问平台的地址 (经 Nginx 为 9000 端口)
datart.address=http://服务器IP:9000
```

ini

用平台账号实际连接一次, 确认账号、密码与授权无误:

```
mysql -uimodel -p -h127.0.0.1 -P3306 imodel -e "SELECT '连接成功' AS 结果;"
```

bash

预期结果: 输出「连接成功」。如果报 Access denied, 检查 2.3 节创建账号时的主机名与密码。

三、主服务首次初始化

首次部署必做。当前版本的发布包缺少 `application-local.yml` 配置文件, 主服务在某些情况下启动时会读取该文件, 找不到就会启动失败, 日志中出现 `application-local.yml cannot be opened`。本章用发布包中已有的 `application-config.yml` 复制补齐, 两者读取的都是 `datart.conf` 中的配置, 不会改变平台行为。补齐一次即可, 之后重启无需重做; 重新安装软件包后需要再做一次。

3.1 定位主程序并备份

```
MAIN_JAR=$(ls /opt/IModel-bundle/app/lib/i-data-viz-with-abi-main-*.jar | head -1)
echo "主程序: $MAIN_JAR"
cp "$MAIN_JAR" "${MAIN_JAR}.bak"
```

bash

3.2 补齐配置文件

```
mkdir -p /tmp/imodel-yml && cd /tmp/imodel-yml
unzip -o "$MAIN_JAR" 'application*.yml'
cp application-config.yml application-local.yml
jar uf "$MAIN_JAR" application-local.yml

# 验证: 必须看到三个 yml 文件
unzip -l "$MAIN_JAR" | grep application
```

bash

预期结果: 列表中同时出现 `application.yml`、`application-config.yml`、`application-local.yml` 三行。

四、Nginx 反向代理

4.1 安装

```
dnf install -y nginx
nginx -v
```

bash

请使用麒麟系统仓库自带的 Nginx，**不要添加 nginx.org 官方软件源**。官方源中的新版本依赖 `libcrypt.so.2`，麒麟 V11 上没有这个库，安装会失败。

4.2 写入配置

```
cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak
```

bash

用以下内容覆盖 `/etc/nginx/nginx.conf`：

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log warn;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;

    map $http_upgrade $connection_upgrade {
        default keep-alive;
        'websocket' upgrade;
    }

    server {
        listen 9000;
        server_name _;
        client_max_body_size 200m;

        location / {
            proxy_pass http://127.0.0.1:8080;
            proxy_set_header Host $http_host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_read_timeout 300s;
        }

        # 模型运行结果流式返回，关闭缓冲并放宽超时
        location /api/v1/models/quick/run {
            proxy_pass http://127.0.0.1:8080;
            proxy_buffering off;
            proxy_cache off;
            proxy_set_header Host $http_host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_read_timeout 3600s;
        }

        # 实时推送 (WebSocket)
        location /socket.io/ {
            proxy_pass http://127.0.0.1:51020/socket.io/;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection $connection_upgrade;
            proxy_set_header Host $http_host;
            proxy_read_timeout 3600s;
        }
    }
}
```

4.3 检查并启动

```
nginx -t && systemctl enable --now nginx
systemctl is-active nginx
```

bash

预期结果：nginx -t 输出 syntax is ok 与 test is successful。启动失败并提示 Permission denied 时，回到 1.4 节检查 SELinux。

五、注册开机自启动

四个服务需要按顺序启动：FTP → 主服务 → Executor → Modern。启动脚本在后台拉起 Java 进程后立即返回，所以每个单元用 ExecStartPost 等待一段时间，确保上一个服务就绪后再启动下一个。

5.1 FTP 服务：/etc/systemd/system/imodel-ftp.service

```
[Unit]
Description=iModel FTP Server
After=network.target

[Service]
Type=oneshot
RemainAfterExit=yes
WorkingDirectory=/opt/IModel-bundle/ftpserve
Environment=JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
ExecStart=/opt/IModel-bundle/ftpserve/ftpd-server.sh start
ExecStartPost=/bin/sleep 10
ExecStop=/opt/IModel-bundle/ftpserve/ftpd-server.sh stop
TimeoutStartSec=120

[Install]
WantedBy=multi-user.target
```

ini

5.2 主服务：/etc/systemd/system/imodel.service

ini

```
[Unit]
Description=iModel Main Server
After=network.target mysqld.service imodel-ftp.service
Wants=mysqld.service imodel-ftp.service

[Service]
Type=oneshot
RemainAfterExit=yes
WorkingDirectory=/opt/IModel-bundle/bin
Environment=JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
ExecStart=/opt/IModel-bundle/bin/imodel-server.sh start
ExecStartPost=/bin/sleep 40
ExecStop=/opt/IModel-bundle/bin/imodel-server.sh stop
TimeoutStartSec=240

[Install]
WantedBy=multi-user.target
```

说明：异机部署时，应用服务器上没有 `mysqld.service`，请从 `After` 与 `Wants` 中删掉它。

5.3 Executor 服务：/etc/systemd/system/imodel-executor.service

ini

```
[Unit]
Description=iModel Executor Server
After=network.target imodel.service
Wants=imodel.service

[Service]
Type=oneshot
RemainAfterExit=yes
WorkingDirectory=/opt/IModel-bundle/executor
Environment=JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
ExecStart=/opt/IModel-bundle/executor/executor-server.sh start
ExecStartPost=/bin/sleep 30
ExecStop=/opt/IModel-bundle/executor/executor-server.sh stop
TimeoutStartSec=240

[Install]
WantedBy=multi-user.target
```

5.4 Modern 服务：/etc/systemd/system/imodel-modern.service

```
[Unit]
Description=iModel Modern Server
After=network.target imodel-executor.service
Wants=imodel-executor.service

[Service]
Type=oneshot
RemainAfterExit=yes
WorkingDirectory=/opt/IModel-bundle/executor
Environment=JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
ExecStart=/opt/IModel-bundle/executor/modern-server.sh start
ExecStartPost=/bin/sleep 20
ExecStop=/opt/IModel-bundle/executor/modern-server.sh stop
TimeoutStartSec=240

[Install]
WantedBy=multi-user.target
```

ini

5.5 加载并启动

```
systemctl daemon-reload
systemctl enable imodel-ftp imodel imodel-executor imodel-modern

# 启动 Modern 会按依赖关系依次拉起前面三个服务，约需 2 分钟
systemctl start imodel-modern
```

bash

说明：这类服务的状态显示为 `active (exited)` 属于正常：表示启动脚本已执行完毕，Java 进程在后台运行。服务是否真正在运行，以第七章的检查结果为准。

六、进程守护

systemd 只负责开机时启动。Java 进程运行中意外退出时，需要由定时任务重新拉起。

6.1 先确认启动脚本不会重复启动

在服务已运行的情况下，再执行一次启动命令，确认不会多出一个进程：

```
ps -ef | grep DatartServerApplication | grep -v grep | wc -l
/opt/IModel-bundle/bin/imodel-server.sh start
sleep 5
ps -ef | grep DatartServerApplication | grep -v grep | wc -l
```

bash

预期结果：两次统计的进程数相同，并且第二次启动时提示服务已在运行。如果进程数增加了，说明脚本不会检查已有进程，请不要配置下面的定时任务，并先结束多出的进程。

6.2 配置定时任务

```
crontab -e
```

bash

加入以下内容（每分钟检查一次，进程不在时自动拉起）：

```
* * * * * cd /opt/IModel-bundle/ftpservice && ./ftpd-server.sh start >/dev/null 2>&1
* * * * * cd /opt/IModel-bundle/bin && ./imodel-server.sh start >/dev/null 2>&1
* * * * * cd /opt/IModel-bundle/executor && ./executor-server.sh start >/dev/null 2>&1
* * * * * cd /opt/IModel-bundle/executor && ./modern-server.sh start >/dev/null 2>&1
```

cron

说明：定时任务中没有加载 `/etc/profile`。如果检查发现进程没有被拉起，在每行命令前加上 `export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk`；。

七、部署验证

```
# 1. 各服务状态
bash /opt/IModel-bundle/ftpservice/ftpd-server.sh status
bash /opt/IModel-bundle/bin/imodel-server.sh status
systemctl is-active mysqld nginx

# 2. 端口监听
ss -tlnp | grep -E ':(9000|8080|51020|4646|50000|59100|3306)\b'

# 3. 主服务启动日志
grep "Started DatartServerApplication" /opt/IModel-bundle/app/logs/$(date +%Y-%m-%d).log

# 4. 经 Nginx 访问
curl -I http://127.0.0.1:9000/
```

bash

预期结果：状态命令提示服务正在运行；表中各端口处于 LISTEN；日志中出现 Started DatartServerApplication；curl 返回 200 或 302。最后在客户端浏览器打开 `http://服务器IP:9000`，能看到登录页即部署完成。

八、日常运维

8.1 常用命令

操作	命令
查看主服务状态	<code>bash /opt/IModel-bundle/bin/imodel-server.sh status</code>
查看当天主服务日志	<code>tail -f /opt/IModel-bundle/app/logs/\$(date +%Y-%m-%d).log</code>
查看 Nginx 错误日志	<code>tail -f /var/log/nginx/error.log</code>

操作	命令
查看各 Java 进程监听端口	<code>ss -tlnp grep java</code>

8.2 全部服务重启脚本

将以下内容保存为 `/usr/local/sbin/imodel-restart.sh`，并执行 `chmod +x /usr/local/sbin/imodel-restart.sh`。脚本放在软件包目录之外，避免清理进程时误结束脚本自身。

```
#!/bin/bash
# iModel 全部服务重启
BUNDLE="/opt/IModel-bundle"

echo "[1/4] 停止服务"
systemctl stop nginx
systemctl stop imodel-modern imodel-executor imodel imodel-ftp
sleep 5

echo "[2/4] 清理残留进程与锁文件"
# 只匹配 Java 进程，不会误结束本脚本
for pid in $(pgrep -f "java.*(${BUNDLE}|KNIME_BATCH_APPLICATION|DatartServerApplication)"); do
    kill -9 "$pid" 2>/dev/null
done
rm -f "${BUNDLE}/executor/workspace/.metadata/.lock"
rm -f "${BUNDLE}/executor/modern-workspace/.metadata/.lock"

echo "[3/4] 按顺序启动 (约 2 分钟)"
systemctl start imodel-modern
systemctl start nginx

echo "[4/4] 检查"
bash "${BUNDLE}/bin/imodel-server.sh" status
curl -s -o /dev/null -w "Nginx 访问返回: %{http_code}\n" http://127.0.0.1:9000/
```

九、常见问题

现象	原因	处理
<code>dnf install</code> 提示缺少 <code>mysql-community-debuginfo</code>	安装包中含调试符号包	执行 <code>rm -f *debuginfo*.rpm</code> 后重新安装
安装 Nginx 提示需要 <code>libcrypt.so.2</code>	添加了 nginx.org 官方源	删除官方源配置，改用麒麟系统仓库安装
日志出现 <code>application-local.yml cannot be opened</code>	未完成首次初始化	按第三章补齐配置文件
日志出现 <code>running in local mode</code>	<code>datart.conf</code> 数据库配置缺失或错误	逐项检查 2.5 节配置，特别是 <code>datart.address</code> 与 <code>datasource.ip</code>

现象	原因	处理
日志出现 <code>Access denied for user 'imodel'</code>	账号主机名或密码与配置不一致	检查 2.3 节授权主机与 2.5 节密码
日志出现 <code>Communications link failure</code>	MySQL 未运行或不可达	<code>systemctl start mysqld</code> ，并确认 3306 端口监听
密码正确仍无法连接数据库	密码含 <code>@</code> 等字符，破坏了连接串	按 2.3 节要求更换密码，同步修改 <code>datart.conf</code>
日志出现 <code>BindException</code> 与端口号	端口已被占用	<code>ss -tlnp grep 端口号</code> 找到占用进程后处理
Nginx 启动失败，提示 <code>Permission denied</code>	SELinux 未放行 9000 端口	按 1.4 节执行 <code>semanage</code> 命令
浏览器访问返回 502	主服务未就绪，或 SELinux 禁止 Nginx 转发	检查 8080 端口；SELinux 为 Enforcing 时执行 1.4 节 <code>setsebool</code>
服务器重启后平台无法访问	未注册开机自启动	按第五章注册并 <code>enable</code> 各服务