VMware虚拟机部署Apollo全攻略：从环境搭建到数据包播放（小白避坑指南）

作为自动驾驶领域的开源标杆，Apollo 的环境部署一直是新手入门的 “第一道坎”。本文基于 VMware 虚拟机 + Ubuntu 系统，详细记录从环境准备到数据包成功播放的完整流程，汇总了部署过程中遇到的 Docker 权限、端口映射、浏览器渲染、数据包下载等核心问题及解决方案，帮助新人少走弯路，快速上手 Apollo 开发。

一、前置环境说明

- 宿主机：Windows 10/11
- 虚拟机：VMware Workstation 16+
- Guest OS：Ubuntu 20.04 LTS（64 位，建议分配 4 核 8G 内存 + 50G 硬盘）
- Apollo 版本：Apollo Neo（最新稳定版）
- 核心目标：完成 Apollo 环境部署、Dreamview 启动、数据包下载与播放

二、核心问题与解决方案全记录

问题 1：Docker 权限不足，无法启动 Apollo 容器

现象

执行 `aem start` 时提示 `permission denied`，Docker 命令需要 `sudo` 才能运行。

解决方案

Code block

```
# 1. 将当前用户添加到docker用户组
sudo usermod -aG docker $USER

# 2. 重启Docker服务
sudo systemctl restart docker

# 3. 注销当前用户重新登录（关键！否则权限不生效）
```

原理

Ubuntu 中 Docker 默认需要 root 权限，将用户加入 docker 组后，可免 sudo 直接操作 Docker，Apollo 的 aem 工具才能正常调用容器。

问题 2：Dreamview 进程已启动，但浏览器访问空白

现象

- 容器内执行 `ps aux | grep dreamview` 能看到 `dreamview_plus` 进程
- 浏览器输入 `http://localhost:8888` 或虚拟机 IP+8888 显示空白页

核心原因

1. 容器 8888 端口未映射到虚拟机
2. 浏览器缓存 / 渲染兼容性问题
3. 虚拟机图形加速未开启

分步解决方案

步骤 1：端口映射（最关键）

退出容器，重新启动 Apollo 并强制映射端口：

Code block

```
# 1. 停止现有环境
aem stop

# 2. 启动并映射8888端口（容器内8888→虚拟机8888）
aem start -p 8888:8888

# 3. 重新进入容器并启动Dreamview
aem enter
aem bootstrap start --plus
```

步骤 2：获取虚拟机真实 IP

虚拟机终端执行（非容器内）：

Code block

```
hostname -I
```

输出示例：`192.168.121.129 172.17.0.1`，取第一个 IP（如 `192.168.121.129`）

步骤 3：浏览器访问正确地址

- 正确地址：`http://192.168.121.129:8888`（不要用localhost）
- 清除浏览器缓存：Ctrl+Shift+Del 勾选缓存 / Cookie，时间范围选 “全部”
- 硬刷新：Ctrl+F5 强制加载

步骤 4：开启虚拟机 3D 加速（根治渲染问题）

1. 关闭 Ubuntu 虚拟机（完全关机，非挂起）
2. VMware 设置：右键虚拟机→设置→显示器→勾选 “加速 3D 图形”，显存调至 128MB+
3. 重启虚拟机，重新启动 Apollo 环境

问题 3：容器内无 netstat/ss 命令，无法检查端口

现象

执行 `netstat -tulpn | grep 8888` 提示 `command not found`

解决方案

1. 虚拟机终端检查端口（容器外，无需进入容器）：

```
ss -tulpn | grep 8888
```

2. 正常输出示例（证明端口已监听）：

```
tcp    LISTEN   0        128              0.0.0.0:8888          0.0.0.0:*
  users:(("dreamview_plus",pid=23617,fd=3))
```

问题 4：数据包下载失败（404 / 域名解析错误）

现象

- 执行官方下载命令提示 `404 Not Found`
- 容器内下载提示 `unable to resolve host address`

核心原因

- 旧版本数据包链接失效
- 容器内 DNS 解析异常

解决方案（终极兜底方案）

步骤 1：虚拟机下载数据包（绕开容器网络限制）

虚拟机终端执行（容器外）：

```
# 1. 创建本地数据包目录
mkdir -p ~/apollo_records

# 2. 下载官方可用数据包（适配Apollo Neo）
wget -O ~/apollo_records/demo_3.5.record https://apollo-system.cdn.bcebos.com/dataset/6.0_edu/demo_3.5.record
```

- 若仍 404，手动浏览器打开链接下载：https://apollo-system.cdn.bcebos.com/dataset/6.0_edu/demo_3.5.record
- 保存路径：`~/apollo_records/`

步骤 2：复制数据包到容器内

```
# 1. 查看Apollo容器ID/名称
sudo docker ps | grep apollo

# 2. 复制本地数据包到容器指定目录（替换<容器ID>为实际值）
sudo docker cp ~/apollo_records/demo_3.5.record <容器ID>:/home/liumaomao/.apollo/resources/records/
```

问题 5：执行 cyber_recorder play 提示文件不存在

现象

```
E0208 21:39:29.580139 34004 record_file_reader.cc:31] [cyber_recorder]File not exist
```

核心原因

- 命令中 “数据包名称” 未替换为真实文件名
- 数据包路径错误

解决方案

```
# 进入容器执行（替换为真实文件名和路径）
cyber_recorder play -f ~/.apollo/resources/records/demo_3.5.record -l
```

- `-l` 参数：循环播放数据包
- 成功标志：终端输出 `[RUNNING] Record Time: xxx Progress: xxx`

问题 6：播放数据包时出现 channel_manager 警告

现象

Code block

```
E0208 21:49:34.612497 34448 channel_manager.cc:326] [cyber_recorder]newly added writer...message type does not match
```

解决方案

忽略警告（不影响核心功能），或添加 `--ignore_type_check` 参数消除警告：

Code block

```
cyber_recorder play -f ~/.apollo/resources/records/demo_3.5.record -l --ignore_type_check
```

原理

数据包中部分消息类型与当前 Apollo Neo 版本略有差异，忽略类型校验即可正常播放。

三、完整流程梳理（新手直接复制执行）

1. 环境部署与 Dreamview 启动

Code block

```
# 1. 安装依赖（若未安装）
sudo apt update && sudo apt install -y docker.io docker-compose

# 2. 配置Docker权限
sudo usermod -aG docker $USER
sudo systemctl restart docker
# 注销重新登录!

# 3. 启动Apollo环境并映射端口
aem stop
aem start -p 8888:8888
```

```
aem enter
aem bootstrap start --plus

# 4. 获取虚拟机IP
exit  # 退出容器
hostname -I  # 记录IP（如192.168.121.129）
```

2. 数据包下载与播放

Code block

```
# 1. 虚拟机下载数据包
mkdir -p ~/apollo_records
wget -O ~/apollo_records/demo_3.5.record https://apollo-system.cdn.bcebos.com/dataset/6.0_edu/demo_3.5.record

# 2. 复制到容器内
CONTAINER_ID=$(sudo docker ps | grep apollo | awk '{print $1}')
sudo docker cp ~/apollo_records/demo_3.5.record $CONTAINER_ID:/home/liumaomao/.apollo/resources/records/

# 3. 容器内播放数据包
aem enter
cyber_recorder play -f ~/.apollo/resources/records/demo_3.5.record -l --ignore_type_check
```

3. 验证效果

- 浏览器访问：`http://192.168.121.129:8888`
- Dreamview 操作：选择 Default Mode→勾选协议→Operations 选 Record→HDMap 选 Sunnyvale Big Loop→点击播放
- 成功标志：车辆在地图上按数据包路径移动，终端显示播放进度

四、新手避坑总结

1. **Docker 权限是基础**：必须将用户加入 docker 组并重新登录，否则后续所有操作都会报错。
2. **端口映射不能少**：Apollo 容器启动时一定要加 `-p 8888:8888` ，否则浏览器无法访问。
3. **访问地址用虚拟机 IP**：localhost仅指向容器内部，必须用 `虚拟机IP:8888` 访问。
4. **数据包下载优先虚拟机**：容器内 DNS 可能异常，先在虚拟机下载再复制更稳妥。
5. **3D 加速是渲染关键**：VMware 未开启 3D 加速时，Dreamview 界面可能空白或卡顿。

五、后续学习方向

1. 熟悉 Dreamview 界面功能：模块控制、仿真场景配置、日志查看
2. 数据包分析：使用 `cyber_topic echo` 查看定位、感知、规划等话题数据
3. 模块开发：基于 Apollo 框架开发自定义算法（如路径规划优化）

通过本文的步骤，你已经成功跨过 Apollo 入门的 “环境部署关”。后续遇到问题时，可重点关注终端报错信息，优先排查路径、权限、网络这三大核心维度。祝你在自动驾驶学习之路上一帆风顺！