盘点镜像、容器层和缓存，再验证存储迁移与清理

完整操作教程 · 参考固定 · 往返恢复验收

Docker占满系统盘时，先区分镜像、停止容器、可写层、构建缓存、卷和日志，再决定能否清理或迁移。不要直接使用带volumes的全量prune。

运行边界：Docker清理与迁移可能影响镜像、容器和数据卷；盘点与备份后由管理员执行。

先找出占用来自Docker还是宿主机其他目录

系统盘空间不足时，先用`df -h`确认文件系统，`docker system df -v`查看镜像、容器、卷和构建缓存的占用，再由管理员核对Docker Root Dir和日志目录。镜像层可能被多个镜像共享，删除一个tag不一定立即释放全部空间；停止容器可写层仍可能占盘；命名卷可能存放数据库或项目结果；JSON日志若无限增长，也可能是主要占用来源。容器内看到的文件系统和宿主机真实分区映射不同，不能只看容器的df结果。

先把对象分成三类：可重建的构建缓存、可重新拉取并已确认版本的镜像、需保留的数据和运行状态。项目输入、数据库、R包库或中间结果可能在卷或bind mount里，不能因为容器停止就认为内容可以删除。给每个重要卷登记负责人、用途、最近备份和保留策略；名称不清楚的卷先查容器挂载关系和应用配置，不要凭`docker volume ls`名字判断。

Docker官方文档说明`docker system prune`会清理未使用的容器、网络、镜像和构建缓存；默认不会删除卷，除非显式加`--volumes`。`docker builder prune`只针对构建缓存，范围相对窄，但构建缓存仍可能是团队加速开发所需资源。清理操作会影响其他用户的工作环境，执行前应确认没有正在运行或待复用的任务，并安排维护窗口。

对象	可能用途	清理前验证
builder cache	加速镜像构建	是否可重新构建及成本
未使用镜像	旧软件环境或回滚版本	是否仍需复现旧任务
容器可写层	临时运行状态	重要输出是否未挂载保存
命名卷	数据库、结果或缓存	负责人、备份与恢复演练

按最小范围清理，并保留可回滚记录

若确认主要占用是可重建的构建缓存，可以先查看缓存明细，按时间或保留空间策略清理，而不是使用会同时处理多个对象类型的广泛命令。清理前保存Docker版本、daemon配置、清理对象列表、可用空间、受影响项目和审批记录。清理后复查`docker system df -v`、宿主机df和容器启动情况，确认没有删除输入、卷或仍需的历史镜像。

如果系统日志膨胀，先检查日志驱动和轮转配置，评估日志是否需保留做故障追踪。不要手动truncate正在使用的文件，除非按运维流程操作并确认服务会正确重开日志。对于构建缓存，按团队使用频率设置保留上限；频繁全清会导致下次构建重新下载大型生信依赖，消耗网络、时间和临时空间。

迁移Docker数据目录前先看daemon和挂载关系

Docker Root Dir通常由daemon配置和启动参数决定。迁移到新盘涉及停止Docker服务、确认所有容器状态、复制或重新拉取镜像、迁移容器状态及卷、调整daemon配置、重新启动并逐项验证。不能仅修改daemon.json路径后直接启动；如果旧数据没复制或挂载点没准备好，容器可能丢失状态或从空目录启动。Docker daemon配置属于系统级运维，应由管理员执行，并保留旧目录直到新路径验证通过。

先决定哪些数据应迁移：镜像可重建，数据库卷可能需一致性备份，bind mount目录可能本来就在独立盘。数据库卷在线复制可能得到不一致快照，需要应用停止、文件系统快照或数据库原生备份。迁移后检查文件属主、SELinux/AppArmor标签、挂载选项、空间告警和容器健康检查。再从一份测试项目启动容器、读写合成文件、验证结果

输出到预期挂载路径。

设置后续容量监控和保留策略

记录Docker对象增长趋势、镜像拉取频率、卷增长速度、日志体积和系统盘可用空间。设定管理员告警阈值时，为镜像构建和临时文件保留缓冲区；系统盘不能长时间接近满载。课题组可约定镜像标签包含软件版本和项目用途，数据使用命名卷或bind mount时写明负责人和备份策略。避免依靠频繁手动prune维持运行。

如果确需扩大Docker工作盘，先估计镜像、缓存、数据库和生信中间文件分别需要多少空间，再确认迁移、备份和回滚方案。额外磁盘不能替代卷生命周期管理，也不应把科研数据默认放入容器可写层。按任务配置服务器或工作盘时，把软件镜像增长和项目数据存储分开规划。

确认对象所有者后再决定清理范围

`docker system df -v`提供Docker管理对象的占用概况，但不能代替对业务数据的归属调查。管理员可以用`docker inspect`查看容器关联卷和bind mounts，用镜像标签和运行记录确定是否还有项目依赖。对于匿名卷，容器已删除并不意味着里面没有需要保留的数据。先把卷ID、挂载容器、创建时间、属主团队、最近使用、备份状态写入清单；找不到责任人时先隔离并联系相关用户，不要直接删除。

缓存清理相对安全，但会影响下次构建速度，并可能需要重新下载基础镜像和R/Python依赖。选择清理范围时看BuildKit缓存标签和保留空间策略，先在一台开发节点验证。镜像清理前确认是否已推送到团队镜像库，是否需要离线复现旧论文环境，以及是否可在网络受限时重新获取。把清理前后可用空间和被清理对象数量记下来，避免之后无法解释容量变化。

系统盘迁移适合安排停机窗口。迁移前停止新建容器，确认没有数据库正在写入，制作一致性备份，并记录daemon配置与Docker版本。新盘需设置正确挂载点和开机顺序；如果服务启动时新盘未挂载，Docker可能在旧根分区的空目录继续写入，再次占满系统盘。启动后立即核对Docker Root Dir是否指向新盘，比较镜像、卷和容器数量，并做一次合成读写与重启演练。

常见的误区是只移动Docker数据目录却忽视日志、临时文件、包缓存和用户home目录。先使用`du`和文件系统工具按目录级别找出空间来源，再区分Docker、系统日志、数据库和用户数据。日志应设置轮转和保留期；下载缓存可以按项目策略清理；数据库与参考库需制定单独备份。让监控同时关注空间水位、inode、写入速率和挂载状态，避免只在空间快满时排查。

避免把卷、缓存和镜像混为一谈

镜像是创建容器的只读层，容器可写层保存该容器运行期间的修改，卷是Docker管理的持久数据位置，bind mount则直接关联宿主机路径。构建缓存用于加速Dockerfile执行，清理它通常不会删除已创建的数据库卷，但会增加下次构建时间。不同对象生命周期不同，容量报告必须按类型拆分。将数据库写在容器可写层会让迁移和备份更困难；把临时构建缓存放在系统盘则可能引发磁盘告警。

删除镜像前，确认它已推送到可访问的镜像仓库，digest或标签已保存，仓库的保留策略不会很快清除它。离线或受限网络环境可能无法重新拉取大镜像。对正在使用或作为回滚基线的镜像，保留其digest和构建配方。项目专用镜像最好有责任人和最后使用时间，但“最近没跑容器”不代表镜像无人需要，例如论文复现环境可能数月才使用一次。

Docker卷清理风险更高。匿名卷可能与已删除容器断开，但仍保存有价值的数据库或结果；命名卷也可能被应用绑定到特定名称。执行卷清理前，创建备份或快照并验证可恢复；备份时确认应用一致性，尤其是数据库和索引。存储空间紧张时可先迁移长期数据到独立数据盘，确保容器配置引用新的挂载位置，然后小范围演练重启和恢复。

管理员可设置构建缓存保留上限和日志轮转，而不是在事故发生后一次性清空。项目镜像可按版本、架构和CUDA兼容性标记，避免用户拉到错误的环境。容量监控要覆盖`/var/lib/docker`（或配置的Docker根目录）、日志、临时目录、inode和挂载点。磁盘扩容后也要验证开机自动挂载和权限，防止Docker服务提前在根分区写入。

清理动作要分级，扩大范围前先演练

先考虑最窄的范围：检查构建缓存、清理确认可再生成的缓存，再处理已确认不再使用的dangling镜像；删除停止容器或未使用卷之前，分别确认容器状态与卷内容。不要一开始就使用`docker system prune -a --volumes`，这个组合可能删除大范围未使用对象和匿名卷。Docker文档也提醒卷清理会移除当前没有容器使用的卷，而未挂载并不代表没有价值。

每个清理步骤后查看空间变化、容器列表、卷列表和应用状态。若清理没有释放预期空间，可能是共享层仍被其他镜像引用、文件已删除但容器仍打开、日志不在Docker根目录或底层文件系统快照仍保留数据。必要时由管理员检查文件句柄和快照策略。把清理前的对象清单和备份验证记录留存，以便团队发现历史镜像或数据确实需要时进行恢复。

示例：只盘点，不执行删除

在任何清理之前，可执行以下只读盘点命令。不同Docker版本的输出字段略有差别；先记录命令输出和系统盘挂载点，并由对象负责人确认卷与容器关系。

```
docker info --format '{{.DockerRootDir}}'
docker system df -v
docker ps --all --size
docker image ls --digests
docker volume ls
df -h
# Do not add --volumes to a prune command without an approved backup plan.
```

如果仅构建缓存增长，可进一步评估`docker builder prune`的范围和`--keep-storage`选项；执行前仍要确认团队能接受重新构建时间。全盘清理命令不应放入无人值守定时任务，因为它可能在其他用户刚停掉容器后删除其回滚环境。盘点结果应关联采集日期、Docker根目录和节点名称。

区分Docker根目录与用户项目存储

Docker root dir可以放在独立系统工作盘，但项目输入输出最好仍落在有备份与配额的项目存储中。镜像缓存增长与科研数据增长的监控策略不同：镜像可按版本淘汰，项目结果需要负责人批准后归档，数据库卷要做一致性备份。把三者全放进同一Docker目录，会使扩容和生命周期管理变得困难。可以在容量台账中分别记录root dir、项目bind mount、named volume和日志文件的文件系统位置。

若更换盘符或挂载点，检查Docker服务启动依赖，确保数据盘先挂载再启动daemon。新系统盘上线后，设置容量告警和挂载失败告警；启动时若数据盘缺失，应阻止Docker写到根分区的空挂载目录。迁移完成后保存旧配置、迁移时间、文件数量、校验摘要和回滚条件。即使空间恢复，也要复盘何种对象持续增长，并设置责任人与复查时间，避免数周后重复发生。

配套练习包

本PDF对应的ZIP包含本文代码、主题模板或合成数据、README和SHA256校验文件。运行前核对包内清单，并按文章说明在独立目录验证。

参考资料

Docker system prune 文档 · 来源资料

Docker builder prune 文档 · 来源资料

Docker数据卷与bind mount · 来源资料

操作建议：先在合成小样本上验证流程，再依据项目参考、标签合同、存储策略与获配资源验收真实数据。