



Ubuntu 24.04 LTS 金融/AI 工作站搭建手册

硬件装机过程总结

硬件组件清单

搭建高性能的金融/AI工作站需要以下主要硬件组件：

- **主板 (Motherboard)**：AMD 平台主板（例如支持 AMD Ryzen 系列处理器的 ATX 主板），提供 CPU 插槽、内存插槽、M.2 接口、PCIe 插槽等。确认主板尺寸与机箱兼容，主板自带的后挡板正确安装。
- **CPU**：AMD Ryzen 系列多核处理器（例如 16 核/32 线程的 Ryzen 9），提供强大的多线程计算能力。注意在取放 CPU 时避免触碰针脚，并对准主板 CPU 插槽的缺口或三角标识小心放入。
- **内存 (RAM)**：高速 DDR4/DDR5 内存条（例如 32GB 容量，双通道），插入主板内存插槽时对齐缺口，一次插入到位。优先将内存安装在主板指定的插槽（通常颜色相同的插槽）以启用双通道模式。
- **固态硬盘 (SSD)**：NVMe 协议 M.2 固态硬盘作为主存储（例如 1TB NVMe SSD），安装在主板 M.2 插槽上并固定螺丝。确保主板上对应的 M.2 插槽支持 NVMe，并在 BIOS 中启用 NVMe 模式。NVMe 硬盘支持 SMART 自检，可通过相应工具监控健康状态。
- **显卡 (GPU)**：NVIDIA GeForce RTX 系列独立显卡（例如 RTX 5070 Ti），插入主板 PCIe x16 插槽并用机箱螺丝固定。选择与机器用途相匹配的高性能 GPU 以满足深度学习和图形显示需求。注意安装显卡时扣紧插槽挡板，连接必要的外接供电线。
- **电源 (PSU)**：额定功率充足的 ATX 电源（例如 800W 金牌全模组电源），为 CPU、GPU 等提供稳定电力。选择有足够的功率余量的电源以确保高负载下系统稳定。模组电源便于走线，只连接所需的线缆以减少机箱内部线材杂乱。
- **CPU 散热器**：一体式水冷散热器（例如 240mm 或 360mm 冷排的 ARGB 水冷套件），提高 CPU 散热能力。水冷头与 CPU 接触面预涂导热硅脂（若无则需手动涂抹一层薄均匀的硅脂），安装支架对应 AMD 插槽规范锁紧。冷排可固定在机箱顶部或前部，搭配散热风扇。
- **机箱 (Case)**：中塔式或全塔式机箱，确保能容纳上述 ATX 主板、长显卡和水冷冷排。机箱应自带良好的风道设计和多个风扇位，方便安装额外机箱风扇。注意机箱前面板提供的接口（USB、音频等）是否满足需求，并确认机箱内部具备足够空间便于走线和组件安装。
- **机箱风扇**：额外的 120mm/140mm 风扇，用于机箱进出风加强散热。通常前面板作为进风，后面板和顶部作为排风，组建合理的空气流动方向。ARGB 风扇需要连接控制器或主板 ARGB 接口实现灯光同步效果。

安装顺序与注意事项

组装过程中应遵循合理的顺序和注意事项，以确保硬件正确安装并避免组件损坏：

1. **准备工作**：在平坦防静电的工作台上展开主板，将防静电手环接地后开始操作。先检查机箱内的主板固定铜柱（standoff）位置是否对应主板孔位，必要时调整或添加以匹配主板尺寸。
2. **安装CPU和内存**：将主板放在防静电垫上。打开CPU插槽的固定杆，小心对准CPU的金三角标识与主板插槽上的标记，将CPU平放插槽并合上固定杆。随后，将内存条按对角顺序插入内存插槽：对准缺口后用力压入，直至两侧卡扣自动锁定内存。若有多个内存条，按照主板手册指示插入对应插槽以后用双通道。

3. **安装M.2 SSD**：找到主板上的M.2插槽，卸下固定螺丝。将NVMe SSD 沿45度角插入插槽，再按平固定，用小螺丝将其固定在主板支架上。注意避免用手触碰SSD的金手指部分。
4. **安装主板到机箱**：机箱侧板卸下后，将预装好的主板后挡板对准机箱预留孔位从内侧卡入。然后将主板小心放入机箱，对准机箱铜柱和主板螺丝孔。使用螺丝刀固定所有主板螺丝（一般对角依次拧紧），但不要过度用力以免损坏主板PCB。确保主板背面没有直接接触机箱金属而是垫有铜柱。
5. **安装电源**：将电源放入机箱电源仓（通常在机箱底部），使电源风扇朝机箱外侧或下侧通风孔。用附带螺丝将电源从机箱背板固定。暂时不要连接电源线，方便后续布线。确保电源开关处于关闭状态以防误触通电。
6. **安装散热器**：如果使用一体式水冷，可选择在主板装入机箱前安装水冷头和背板支架，或者在主板固定后再安装冷头视空间而定。典型步骤：先将冷排用螺丝固定于机箱顶端或前端风扇位（根据机箱设计和冷排尺寸选择），将附带的风扇安装在冷排上（注意风扇朝向和进出风方向保持一致）。然后将水冷头放置在CPU上方的支架上并均匀拧紧对角螺丝固定。**注意**：风扇的安装方向影响风道——凸面为进风、凹面为出风，通常机箱前/下部进风，后/上部出风^{① ②}。安装时确保冷排风扇的朝向实现预期风道效果，使冷空气吹入冷排或热空气排出机箱。
7. **安装独立显卡**：拆下机箱后部与PCIe插槽对应的挡板，将显卡对准主板上的PCIe x16 插槽，用力均匀插入直至卡扣扣紧显卡。用螺丝将显卡尾部固定在机箱挡板位上。大型显卡应考虑安装支架支撑，以防止长时间使用后的形变。确认显卡不干涉其他元件，特别是水冷管路和内存条。
8. **连接电源线缆**：先将电源的**24针主板供电线**插入主板对应插座（通常在主板右侧），确保卡扣扣紧。再将**CPU供电线（4+4针或8针）**连接到主板CPU供电插座（主板顶部附近）以为处理器供电。**将显卡供电线（6+2针 PCIe电源接口）**插入显卡顶部或侧面的电源插座（如显卡有双8针，则每个都需连接）。若机箱内有SATA设备（如2.5寸SSD、风扇控制器），连接SATA电源线到相应设备。使用扎带整理多余的线缆，避免干涉风扇转动。
9. **接驳风扇与灯光线缆**：将机箱和散热器上的风扇电源线插入主板对应的风扇针脚接口。一般**CPU风扇**或**水泵**应接主板的**CPU_FAN**或**AIO_PUMP**接口，机箱风扇接**CHA_FAN**（或**SYS_FAN**）接口。^③ 例如，一体式水冷的冷排风扇供电线可接主板上的**CHA_FAN**插座，而水冷头的泵供电线则接到专用的**AIO_PUMP**插座^③。对于带RGB灯效的风扇或水冷头，将其**ARGB 3针插头**通过集线器或分线器接入主板的5V 3针ARGB接口^④。例如，某些水冷套件附带“一拖三”ARGB分线线缆，可将两个风扇和水冷头的ARGB线合并后接入主板单一ARGB接口^④。务必区分ARGB（5V 3针）与传统RGB（12V 4针）的接口，避免接错导致设备损坏^⑤。连接时注意接口防呆设计，对准缺口插入并插到底。
10. **机箱前面板接线**：找到机箱前面板引出的细线，包括**开机按钮(PWR SW)**、**重启按钮(Reset SW)**、**硬盘指示灯(HDD LED)**、**电源指示灯(PWR LED)**等，将它们按照主板说明书指示接入主板前面板针脚（通常在主板右下角的多针接口，每个针对应不同功能）。注意正负极：LED类接口有正负之分（一般白色或黑色线为负极），开关则无极性。将**前置USB 3.0接口线**插入主板对应的19针插座（对准防呆胶舌插入），**前置USB 2.0线**插入9针插座，**音频接口线(HD Audio)**插入主板标识为 HD_AUDIO 的插座。确保这些连接稳固且走线整洁，不影响机箱侧板闭合。
11. **最后检查**：所有必要的线缆均已连接后，仔细检查系统各部分：CPU散热器螺丝是否拧紧但不过压，内存和显卡是否完全插入到位，所有供电线是否牢固，风扇线路是否避开扇叶。可暂时不盖机箱侧板，先进行试运行。将显示器、键鼠连接到对应接口后，接通电源并开机，确认主板指示灯亮起、风扇开始旋转。如果出现无法开机的情况，立即断电重新检查连接。初次开机应进入BIOS，监控CPU温度、识别内存容量和硬盘等。如果一切正常，再关闭电源，整理好余线并盖上机箱侧板。

注意：装机过程中要随时防范静电对元件的损坏，操作前排除静电并避免直接触摸元件的金手指和电路部分。遵循说明书指导连接风扇和灯光等附加线缆，以确保所有组件正确供电和工作。一旦硬件部分组装完毕，下一步将进入软件系统的安装与配置。

Ubuntu 24.04 系统全新安装过程

BIOS 设置与启动盘制作

在安装操作系统之前，需要对主板 BIOS/UEFI 进行适当设置，并准备可引导的安装介质：

- **调整BIOS设置**：开机后反复按下 `Del` 或 `F2` 键进入 BIOS/UEFI 设置界面。确保已启用 UEFI 引导模式，并将 **USB启动** 设为优先顺序⁶。如果主板支持，开启CPU的虚拟化功能（AMD SVM）以便后续使用虚拟机或容器。建议 **关闭安全启动（Secure Boot）** 功能，以避免安装第三方驱动时遇到签名限制问题⁷。实际上，禁用 Secure Boot 可以防止安装过程中 NVIDIA 显卡驱动加载失败或不正常⁸。稍后待驱动配置完成后，可选择重新开启 Secure Boot，但在此期间保持关闭状态以简化驱动部署⁸。
- **制作启动U盘**：下载 Ubuntu 24.04 LTS 的 ISO 镜像文件（从 Ubuntu 官网或清华/阿里镜像源获取最新版）并将其写入 U 盘。可使用 [Rufus](#)（Windows 环境）或 `dd` 命令（Linux/macOS 环境）将 ISO 烧录到 8GB 及以上容量的 U 盘⁹。在 Rufus 中选择 GPT 分区方案和 UEFI 启动，以匹配新主板设置。写入完成后，U 盘即成为 Ubuntu 安装盘。

完成以上准备后，将启动盘插入目标工作站的 USB 3.0 接口，并打开电源开始安装过程。

安装步骤详解

1. **启动安装程序**：开机时按下主板启动菜单热键（如 `F12` 或 `Esc`），选择从 U 盘引导系统¹⁰。成功引导后将看到 Ubuntu 安装菜单，选择“Try or Install Ubuntu”并按回车。稍候将自动进入 Ubuntu Live 环境，并弹出安装向导。
2. **选择安装语言**：在欢迎界面中，选择首选语言为简体中文（或 English，根据需要），然后点击“下一步”继续¹¹。界面语言会切换成所选语言，方便后续操作。
3. **配置键盘和辅助选项**：安装程序会尝试检测键盘布局，一般自动匹配。如果有需要，可手动选择键盘布局为“汉语/美式键盘”等¹²。接下来是可访问性选项，如果无特殊需求直接保持默认并继续¹³。这些辅助功能（屏幕阅读器、大字体模式等）在安装后也可配置。
4. **网络连接**：如果系统检测到网络，可选择连接 Wi-Fi 或使用有线网络¹⁴。联网有助于在安装过程中下载更新和第三方驱动。此步骤可按需选择，未联网也不会中断安装。
5. **选择安装模式**：在安装类型界面，选择“安装 Ubuntu”继续¹⁵。Ubuntu 24.04 提供 **交互安装**（即传统图形向导步骤）和 **自动安装**两种方式¹⁶。这里我们选择交互安装，以便逐步配置。
6. **软件包选择**：安装向导会询问要安装的应用程序集合，可选择 **默认集合** 或 **扩展集合**¹⁷。默认集合相当于精简安装（不包含某些体积较大的应用），扩展集合则安装完整 Ubuntu 桌面环境。根据工作站用途，建议选择 **扩展集合** 以获取完整桌面和常用工具集。
7. **第三方软件和更新**：勾选 **安装第三方软件** 选项，以便自动为图形显卡、Wi-Fi 模块安装专有驱动，并添加对 MP3 等专有媒体格式的支持¹⁸。这一选项会启用 Ubuntu 的“restricted”源仓库，包含 NVIDIA 驱动等非开源软件，推荐开启以提升硬件兼容性¹⁸。同时可以选择**在安装过程中下载更新**，这样安装完成后系统即是最新状态（需要联网支持）。
8. **磁盘分区**：进入分区配置界面。对于全新硬盘且希望自动分区的用户，选择“擦除磁盘并安装 Ubuntu”，安装程序将自动创建必要的分区¹⁹。自动模式通常会创建 EFI 引导分区、根分区和交换分区，并可在“高级功能”中启用 LVM 逻辑卷或磁盘加密（需 TPM 芯片支持）²⁰。如果需要手动划分分区（例如单盘多系统或自定义目录规划），选择“手动分区”进入高级模式。基本建议如下²¹：
9. **EFI系统分区**：类型为 FAT32，大小 100–500MB，挂载点 `/boot/efi`，用于存放启动引导文件（必需）。
10. **根分区 (/)**：类型 EXT4，建议大小至少 50GB 或更大（根据需要可分配更多空间给 `/`），挂载点 `/`（必需）。

11. **交换分区 (swap)**：大小约等于物理内存容量（例如32GB内存则交换分区设32GB），用于内存交换（可选，但有助于休眠和防止内存溢出）。
12. **/home分区**（可选）：如希望用户数据与系统隔离，可单独创建EXT4分区并挂载到 `/home`，大小取决于存储需求。如果选择手动分区模式，根据自己的需求规划以上分区。设置完成后点击“下一步”。（注意：如需全盘加密需在自动分区且硬件支持下进行；手动模式不支持启用安装时加密²²）。
13. **用户信息配置**：继续根据提示填写**时区、用户名和主机名**等信息。首先，选择所在地理位置用于设置系统时区，中国大陆用户一般选择“Asia/Shanghai (上海)”²³。接下来创建一个普通用户帐户：输入您的姓名、计算机名、用户名和登录密码²⁴。请使用较强的密码并记忆。可选项中决定开机是否自动登录，一般出于安全保持需要密码登录。确认无误后点击“下一步”。
14. **开始安装**：安装程序将显示总结信息，确认磁盘分区方案和安装设置无误后，点击“现在安装”。系统开始将文件复制到硬盘并执行安装过程。这期间可查看滚动的提示或等待进度条完成。若之前勾选了更新和第三方软件，安装过程可能稍长，需要联网下载相应软件包。
15. **完成安装并重启**：安装完成后，会提示重启计算机。点击重启并取出安装U盘。当系统重新启动时，应从硬盘引导进入全新的Ubuntu 24.04 LTS系统。在首次启动过程中，如果之前没有联网，可能会经历一个“欢迎”向导以设置在线账户、隐私等，根据需要操作或跳过即可。

系统初始化配置

新系统首次启动后，为确保工作站稳定高效，建议进行如下初始化配置：

- **更新系统软件**：打开终端或通过“软件更新”工具检查并安装最新更新。终端中执行：

```
sudo apt update && sudo apt upgrade -y
```

以获取最新的软件包和安全补丁（如安装时已勾选更新则此时可能已是最新）。升级完成后重启计算机使内核更新等生效。

- **启用中文支持（可选）**：如果安装时未选择中文界面，可通过以下方式添加中文语言包。确保网络畅通，在终端执行：

```
sudo apt install language-pack-zh-hans language-pack-zh-hans-base
```

然后运行 `sudo dpkg-reconfigure locales`，选中 `zh_CN.UTF-8` 并设为默认区域语言。这样系统界面、菜单等将切换为简体中文。也可以在“Settings->Region & Language”中添加Chinese支持并拖动到首位。

- **驱动与硬件检测**：进入“软件与更新 (Software & Updates)”应用，切换到“附加驱动”选项卡。系统会扫描可用的第三方驱动。如果有更新的NVIDIA专有驱动或其他设备驱动可用，这里会显示出来并推荐安装版本。例如，对于NVIDIA显卡，Ubuntu通常推荐使用 `nvidia-driver-XXX`（如535版）²⁵。确认所需驱动选中并应用更改，让系统安装这些驱动。稍后详细步骤会在驱动配置章节说明。
- **调整电源和性能设置**：在“设置->电源”中，根据需要调整待机休眠策略。在工作站环境下，可考虑关闭睡眠或休眠以确保长时间运行不中断关键服务。同时，切换“性能模式”以获得最佳性能（若系统提供此选项，例如部分Laptop有省电/平衡/性能模式开关，台式机一般不需要特别调整）。
- **主板 BIOS 更新（可选）**：如主板厂商发布了新的UEFI固件改善了兼容性或性能，可考虑在操作系统安装完毕后升级BIOS。但升级有风险且需严格按照厂商方法进行，非必要可暂不更新。

完成以上配置后，Ubuntu 24.04 LTS系统的基本环境已就绪。接下来进行针对硬件的驱动安装和优化设置。

驱动安装与配置

新安装的Ubuntu系统通常会自动识别大部分硬件并加载开源驱动程序。但为了充分发挥硬件性能，仍需手动安装配置一些专有驱动和工具，尤其是针对AMD平台和NVIDIA显卡。以下是各部分驱动配置说明：

- **AMD CPU及主板芯片组驱动**：在Linux内核中，AMD CPU以及主板芯片组的大部分驱动已内建支持，无需像Windows那样额外安装芯片组驱动包²⁶。运行 `lspci -v` 可看到AMD芯片组设备均已由对应内核模块驱动²⁶。因此Ubuntu下无需安装AMD官方芯片组驱动。但是，有两点可优化：
 - 安装CPU微码更新：确保安装 `amd64-microcode` 软件包（对于Intel则是 `intel-microcode`）。Ubuntu仓库提供了AMD处理器最新微代码补丁，安装后在引导时加载，可提高稳定性和安全性。执行：

```
sudo apt install amd64-microcode
```

安装完成后重启，微码更新将生效。

- **CPU功耗调节驱动**：新的Linux内核对现代AMD Ryzen处理器提供了 `amd_pstate` 驱动，提高频率调节响应和节能效果。Ubuntu 24.04默认已支持该驱动，如需验证可运行 `grep amd_pstate /boot/config-$uname -r` 或 `cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_driver` 查看。一般无需手动干预，但可以通过在 `/etc/default/grub` 中追加内核参数启用主动模式，例如：
`GRUB_CMDLINE_LINUX_DEFAULT="amd_pstate=active"`，更新GRUB并重启（根据需要选择，非强制）。此外，主板上的控制器（如USB控制器、SATA控制器、SMBus等）Linux内核也都有相应驱动。因此正常安装Ubuntu后，AMD平台的大部分功能即可用，无需另装驱动包。
- **NVIDIA 独立显卡驱动**：为获得最佳的3D性能和CUDA加速能力，需要安装NVIDIA的专有驱动。按照上文初始安装步骤，如果已勾选“安装第三方软件”，系统可能已经安装了推荐的NVIDIA驱动（可通过 `nvidia-smi` 命令验证，若输出显卡信息则说明驱动已生效）。若未安装，可通过以下方式配置：
 - **使用Ubuntu驱动管理工具**：打开“软件与更新 -> 附加驱动”，耐心等待系统检测。之后会列出可用的NVIDIA驱动版本，如 `nvidia-driver-535`（推荐）、“Open”开源版本和开源的 `nouveau` 驱动等²⁵。选择**专有驱动 (proprietary, tested)** 的推荐版本并应用，系统将自动下载并安装对应驱动。安装完成后重启系统，独显驱动即生效。通过终端 `ubuntu-drivers devices` 也可查看推荐驱动版本²⁵。
 - **手动安装驱动**：高级用户可以选择从NVIDIA官方网站下载.run安装包进行手动安装。然而此方法步骤复杂（需退出图形界面、禁用Nouveau驱动、命令行运行安装脚本等），不如使用Ubuntu自带方式方便可靠。一般只有在需要特定新版驱动且仓库未提供时才考虑手动安装。在24.04发布后，仓库中的驱动版本已相当新，通常不需要手动安装。
- **Secure Boot 设置**：NVIDIA驱动在安装时，如果检测到Secure Boot开启，会要求创建MOK密钥并签署驱动模块，否则驱动模块将被内核拒绝加载。因此强烈建议在安装NVIDIA驱动前关闭Secure Boot⁸（如前文所述）。若已经开启Secure Boot，则在首次重启后Ubuntu会进入“MOK管理”界面要求登记证书，需要手动执行额外步骤。因此，为避免麻烦，确保BIOS中Secure Boot为Disabled直到驱动正常工作⁸。驱动装好后可重新启用Secure Boot并采用MOK方式签名模块，但具体过程较复杂，此处不再展开。完成以上操作后，通过 `nvidia-smi` 可以查看显卡状态，包括驱动版本、显卡功耗温度等，以确认驱动安装成功。例如，可看到类似输出：

```

+-----+
+-----+
| NVIDIA-SMI 535.54.03      Driver Version: 535.54.03      CUDA Version:
12.2      |
|-----+-----+
| GPU  Name      Persistence-M| Bus-Id      Disp.A | Volatile Uncorr.
ECC      |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util
Compute M. |
|
=====+=====+=====+
| 0  NVIDIA RTX 5070 Ti    Off  | 00000000:01:00.0 Off  |
N/A      |
| 30%   45C     P8    15W / 300W |      0MiB / 8192MiB |      5%
Default  |
+-----+-----+
+-----+

```

这表示驱动正常运行，显卡温度45°C，空闲功耗15W等信息。若未显示，可能是驱动未正确安装或者未加载，需要检查日志或配置。

- **存储硬盘支持 (NVMe & SMART)**：Ubuntu内核对NVMe SSD有原生支持，安装时若SSD已正确识别，后续无需特殊驱动。可以使用 `lsblk` 或 `nvme list` 来列出 NVMe 设备信息。为了监控SSD健康状态，建议安装 `smartmontools` 工具：

```
sudo apt install smartmontools
```

然后使用 `sudo smartctl -a /dev/nvme0n1` 查看NVMe硬盘的SMART信息，包括温度、已写入总字节、坏块计数等。注意NVMe的SMART支持自smartmontools 6.5起是实验性的，但基本信息可正常读取²⁷。另外，Ubuntu默认会开启每周一次的 `fstrim` 服务来执行SSD的TRIM优化，无需手动干预（可通过 `systemctl status fstrim.timer` 确认定时任务）。若需要手动触发TRIM，可执行：

```
sudo fstrim -av
```

- **传感器监控（温度、电压、风扇转速）**：建议安装 `lm-sensors` 工具来读取主板和CPU/GPU的各类传感器数据。安装命令：

```
sudo apt install lm-sensors
sudo sensors-detect
```

第一次运行 `sensors-detect` 时会检测系统中的传感器芯片并提示是否加载相应内核模块。对所有询问输入 `YES` 以后用检测到的芯片支持²⁸。检测结束后，它会给出需要加载的模块列表，并可选择自动加入 `/etc/modules` 以开机自动加载²⁹。执行完 `sensors-detect` 后，运行 `sudo systemctl restart`

`systemd-modules-load` 或重启系统，以确保新模块加载。然后运行 `sensors` 命令，即可看到诸如 CPU 温度、主板各路电压、风扇转速等读数³⁰ ³¹。例如典型输出：

```
k10temp-pci-00c3
Adapter: PCI adapter
Tctl:      +33.8°C
Tdie:      +33.8°C
Tccd1:     +34.5°C

nvme-pci-0100
Adapter: PCI adapter
Composite:  +29.9°C (low = -5.1°C, high = +84.8°C)
```

以上表示AMD CPU CCD温度约33°C，NVMe硬盘温度约29.9°C等。若有多个温度传感器（如主板芯片），可能需要识别哪些对应CPU核心温度（一般 `Tctl/Tdie` 为CPU温度）。对于NVIDIA GPU，其温度和功耗可通过上面提到的 `nvidia-smi` 直接查看，也可以安装 `nvtop` 工具以图形化监控GPU状态。此外，可安装 `psensor` 或GNOME扩展如“Sensors”来在桌面上实时显示温度和风扇信息，以便随时掌握系统热度情况。

通过以上驱动和传感器的安装配置，工作站硬件将处于良好受控状态。其中AMD平台的兼容性在Linux下很好，而NVIDIA驱动是需重点确保的部分。一旦驱动就绪，接下来可以着手安装各种开发环境和软件包。

软件包与开发环境安装

搭建金融/AI工作站的软件环境，需要安装一系列常用工具、开发库和配置高效的开发体验。本节将涵盖命令行工具、编程语言环境、科学计算工具以及一些特色软件的安装和配置步骤。

系统基础工具与包管理

首先更新APT源并安装一些常用基础工具：

```
sudo apt update && sudo apt install -y build-essential git wget curl vim htop
unzip
```

- **build-essential** 提供编译工具链（gcc/g++等），必要时编译源码包或Python扩展时用到。- **git** 版本控制工具，用于管理代码仓库。- **wget/curl** 命令行下载工具，用于下载文件或通过HTTP API抓取数据。- **vim** 或其他编辑器（nano等），方便在终端编辑配置文件。- **htop** 交互式进程监视工具，可实时查看CPU/内存使用情况、进程列表等。

安装完成后，可以使用 `htop` 命令打开监视界面，检查系统各核心占用、负载等。基础工具安装为后续配置提供了便利。

Zsh及Oh-My-Zsh配置

为了提升命令行体验，建议将默认Shell由bash切换为 **Zsh** 并使用 Oh-My-Zsh 框架进行管理和美化。

1. 安装**Zsh**：Ubuntu仓库提供了zsh，直接安装：

```
sudo apt install -y zsh  
chsh -s /usr/bin/zsh # 切换默认shell为zsh，重启终端生效
```

再打开一个新终端，会发现已进入Zsh（首次启动可能提示配置向导，可根据需要选择skip）。

2. 安装**Oh-My-Zsh**：这是一个流行的Zsh配置管理工具，提供大量插件和主题。使用官方提供的一键安装脚本：

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

按提示完成安装。这会将Oh-My-Zsh的配置克隆到 `~/.oh-my-zsh` 目录，并生成基础的 `~/.zshrc` 配置文件。

3. 设置主题和插件：Oh-My-Zsh默认主题是“robbyrussell”，可以改为更美观实用的主题，例如 `agnoster` 或 `powerlevel10k`。修改 `~/.zshrc` 中的 `ZSH_THEME="agnoster"` 然后保存。对于Powerlevel10k主题，需要额外安装：

```
git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ~/.oh-my-zsh/custom/themes/powerlevel10k
```

然后在 `~/.zshrc` 中设置 `ZSH_THEME="powerlevel10k/powerlevel10k"`。下次启动Zsh会进入Powerlevel10k配置向导，可根据喜好设置提示符样式（注意该主题在终端中显示特殊图标，可能需要安装Powerline/Nerd字体以完整显示图标）。

4. 启用有用的插件：Oh-My-Zsh自带很多插件，在 `~/.zshrc` 里的 `plugins=(git)` 一行可以添加。常用插件如：`git`（增强git命令提示）、`z`（快速目录跳转），`autojump`（类似功能，需要额外安装 `autojump` 包），`sudo`（让上一条命令快速加sudo重试）等。比如设置：

```
plugins=(git z sudo)
```

保存后重启shell使其生效。这样Zsh就配置好了，之后Shell会有高亮、自动补全和丰富的提示，提高命令行操作效率。

Node.js与Puppeteer环境

Node.js在金融数据抓取、网页自动化以及构建Web服务方面用途广泛。这里安装最新LTS版本的Node.js，以及用于无头浏览器操作的Puppeteer库和相关工具（Astro静态站点及PDF生成工具依赖）。

1. 安装**Node.js**：建议使用 **Node Version Manager (nvm)** 来安装和管理Node版本：

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.4/install.sh |  
bash  
source ~/.bashrc # 或 source ~/.zshrc 以加载 nvm  
nvm install --lts  
nvm use --lts
```

以上将安装当前LTS版本（例如Node.js 18.x或20.x）。可通过 `node -v` 验证安装。使用nvm好处是可随时切换Node版本，且不需sudo权限。（若不使用nvm，也可直接 `sudo apt install nodejs npm` 安装默认版本，但通常版本较旧，不太推荐）

2. **配置国内源(可选)**：如果下载Node模块过慢，可临时设置 `NVM_NODEJS_ORG_MIRROR` 环境变量指向国内镜像，或者安装 `nrm` 工具切换npm源。不过一般来说 Node安装完成后再通过更换npm源解决即可。例如：

```
npm config set registry https://registry.npmmirror.com
```

这样以后使用 `npm install` 会从国内源加速。

3. **安装Puppeteer**：Puppeteer是Node的一个库，用于控制Chromium进行无头浏览器操作（抓取网页、生成PDF等）。可以全局安装也可以在项目中按需安装。如果仅为通用用途，全局安装：

```
npm install -g puppeteer
```

初次安装会自动下载与之对应的Chromium浏览器内核（几百MB）。安装完成后，可编写Node脚本使用 Puppeteer打开网页、截图或爬取数据。在本工作站的场景下，Puppeteer也被用于 Astro 静态网站的PDF生成工具中³²（Astro PDF工具将在后述提及）。注意：Puppeteer默认会下载官方Chromium内核，有时在国内环境可能下载失败，可考虑设置环境变量 `PUPPETEER_DOWNLOAD_HOST=https://npmmirror.com/mirrors` 来加速下载。

4. **Astro 静态站点及 PDF 工具**：Astro是现代的静态网站生成框架，如果用户计划编写技术文档或博客（如项目的中英文手册）并希望导出PDF，那么需要安装项目依赖的Astro及PDF生成工具。根据用户提供的《PDF语言版本生成指南》，此项目需要 Node.js (>=16) 和依赖项 `puppeteer`、`pdf-lib` 等³²。假定用户已经有 Astro项目代码，只需进入项目目录，运行：

```
npm install # 安装package.json中列出的依赖  
npm run build # 构建静态站点（如有）  
npm run pdf:lang:all CN # 运行定制的PDF生成脚本，例如生成所有书籍的中文版PDF33  
34
```

具体命令以用户项目中的scripts定义为准。根据指南，项目提供了PowerShell脚本和Node脚本两种方式执行PDF生成³⁵³⁶。在Linux下可直接使用npm脚本或调用对应的Node脚本。成功运行后将在 `dist/pdf` 目录下生成各语言版本的PDF文件³⁷。如果在此过程中遇到错误，例如“找不到Chromium可执行文件”或“无法启动浏览器”，请确认Puppeteer已成功下载Chromium，或者在项目配置中指定了可用的Chrome路径。根据指南的故障排除部分，确保Node.js版本>=16、依赖安装完整、开发服务器正常运行³⁸。如内存不足生成失败，可降低并发或分批生成³⁹。通过上述环境搭建，用户即可编辑Markdown文档并利用Astro框架构建网站、用Puppeteer将页面渲染为PDF，实现知识文档的多格式输出。

Python 与数据科学环境

Ubuntu 24.04 系统自带Python 3（可执行 `python3 --version` 查看版本，一般为3.12或以上）。Python在量化金融和AI领域广泛使用，可设置如下环境：

1. **安装pip和基础库**：确保pip包管理器可用（`sudo apt install python3-pip`）。然后可选安装一些常用数据科学库，如NumPy、Pandas、Matplotlib等：

```
pip3 install --upgrade pip  
pip3 install numpy pandas matplotlib scipy scikit-learn
```

这些库提供了数值计算、数据分析和机器学习基本功能。如果需要隔离环境，推荐使用 `python3 -m venv env` 创建虚拟环境或使用Conda管理环境。

2. **Jupyter Lab**：交互式实验环境。安装：

```
pip3 install jupyterlab
```

安装完成后，运行 `jupyter lab --no-browser --ip=0.0.0.0` 即可启动Jupyter服务。由于本机是无显示的服务器环境，使用 `--no-browser` 不自动打开浏览器，`--ip=0.0.0.0` 允许远程访问。第一次运行会提示访问的URL带有token，复制该URL在本地主机浏览器访问即可。注意：若在远程访问Jupyter时，需确保防火墙开放默认端口8888或使用SSH隧道/安全的方式访问。Jupyter Lab可以方便地编写和调试Python代码、运行数据分析，并支持富文本记录，是金融研究和AI实验的常用工具。

3. **科学计算与深度学习**：根据需要安装相应框架，如TensorFlow或PyTorch。可通过pip安装最新版本，例如：

```
pip3 install torch torchvision torchaudio # PyTorch CPU版本
```

或带CUDA支持的GPU版本（确保已安装对应版本CUDA驱动）。也可安装TensorFlow：

```
pip3 install tensorflow
```

这些框架安装体积较大且依赖NVIDIA CUDA。在本工作站上，由于装有NVIDIA GPU和驱动，可利用GPU加速训练。注意选择兼容GPU驱动的框架版本，或通过Anaconda安装以避免兼容性问题。

4. **金融分析库**：QuantConnect Lean平台将在下节单独介绍。此外，还有其他常用库如 TA-Lib 技术指标库、ccxt加密货币API、rqalpha国内量化框架等，可根据需求安装：

```
pip3 install TA-Lib ccxt yfinance
```

这些可以用来获取行情数据、计算指标、执行回测等，为构建量化交易策略提供支持。

Solaar (Logitech外设管理)

如果使用罗技的无线键鼠或优联设备，可以安装 **Solaar** 工具来管理这些设备（查看电池电量、配对等）。Ubuntu 仓库提供Solaar：

```
sudo apt install solaar
```

安装后，在应用菜单中启动Solaar。它会自动检测插入的Logitech优联接收器或蓝牙适配的罗技设备，显示每个设备的电量、状态等。对于需要长时间监控的工作站，通过Solaar可以及时了解键鼠电量并计划充电/更换电池。

其他有用工具

- **htop 和系统监控**：前面已安装htop。使用htop可以在终端以图形界面查看每个CPU核心利用率、内存使用、进程列表并支持按键排序和筛选。这对于监控工作站运行状况非常方便（上图为本机运行htop时的截图，展示了32线程CPU的利用率、内存和Swap占用、系统负载等）。此外Ubuntu自带“系统监视器”GUI，也可查看资源使用和结束进程。
- **sensors和Psensor**：前述lm-sensors安装后，可使用 `sensors` 命令了解温度、电压等。如果希望GUI实时图表，可以安装Psensor：

```
sudo apt install psensor
```

它在窗口中曲线显示温度随时间变化，并支持托盘图标提示高温。

- **网络和远程工具**：安装 `openssh-server` 以支持SSH远程登录管理，安装 `ufw` 并配置简单的防火墙规则（如允许必要的端口、拒绝其余）。安装 `net-tools` (`ifconfig`等) 或使用新命令 `ip addr` 查看网络信息。根据需要安装 `tmux` 或 `screen` 在终端保持会话。

QuantConnect Lean CLI (金融量化平台)

QuantConnect 的 Lean 引擎是一套开源的量化交易回测系统。Lean CLI 工具可以让我们在本地运行 QuantConnect的算法、回测策略，甚至实时交易。以下是Lean CLI环境的搭建步骤：

1. **安装 Docker**：Lean引擎在本地运行时，采用Docker容器封装了算法运行所需的所有依赖环境 [40](#) [41](#)。因此，先安装Docker是必要的。Ubuntu上可通过APT安装Docker CE：

```
sudo apt install -y docker.io
sudo systemctl enable --now docker
sudo usermod -aG docker $USER
```

以上命令安装Docker并启动docker服务，将当前用户加入docker用户组以便无需sudo运行docker命令（执行后需要重新登录会话才生效）。或者按照Docker官方文档 [42](#) 使用官方仓库安装最新版Docker也可。确保 `docker run hello-world` 可以正常运行测试镜像，表示Docker配置成功。加入用户组后，用 `docker info` 检查无需sudo即可运行。

2. **安装 Lean CLI**：Lean CLI通过Python的pip发布。确认已安装pip3，然后执行：

```
pip3 install --upgrade lean
```

这会下载并安装最新版本的Lean命令行工具⁴³。安装完成后，运行`lean --version`应能看到版本号，表示安装成功⁴⁴。Lean CLI会在首次需要时自动下载Lean引擎的Docker镜像，镜像大小较大（数GB），请确保网络通畅且系统有足够的存储空间⁴⁵。

3. 配置QuantConnect账号：Lean CLI需要QuantConnect账号以使用某些功能（如从云端同步项目）。注册QuantConnect账号并获取API Token。如果只是离线本地使用Lean引擎进行回测，可能不需要登录也可使用基础功能，但官方说明推荐登录。执行：

```
lean login
```

会引导至浏览器QuantConnect登录授权，或要求输入API Token。完成后CLI会记录凭据。本地回测不强制要求付费计划，但部分功能（如云托管或组织协作）可能需要订阅⁴⁶⁴⁷。

4. 初始化项目并运行回测：使用Lean CLI新建一个算法项目，例如：

```
lean init MyProject  
cd MyProject  
lean create-project "FirstAlgo"
```

这将在当前目录创建Lean配置文件和示例项目结构。进入项目目录后，编辑`FirstAlgo/FirstAlgo.py`或`FirstAlgo.cs`（Lean同时支持Python和C#）编写策略代码。Lean附带示例策略可直接回测。运行回测：

```
lean backtest FirstAlgo
```

CLI会自动拉取所需的Lean引擎Docker镜像（首次可能花费较长时间）⁴⁸并运行回测，输出策略绩效统计。结果存储在`FirstAlgo/backtests`子目录下，可打开HTML报告查看详细的回测指标。Lean CLI还支持数据下载、优化、实时交易等命令，例如：

```
lean fetch-data "crypto/coarse" --start 2022-01-01 --end 2022-12-31  
lean live FirstAlgo --detach
```

这些命令会使用Docker容器执行所需任务。确保Docker镜像随时是最新的，可以定期运行`lean update`升级CLI版本，以及在回测命令加`--update`参数拉取最新Lean引擎镜像⁴⁹。

5. 注意事项：Lean CLI本质是操作本地Docker来运行Lean，引擎内包含QC提供的大量数据和算法环境，因此务必安装Docker且内存充足（建议16GB以上）以免容器OOM。对于国内用户，拉取Docker镜像可能较慢，可考虑配置Docker加速器或提前手动从镜像源下载。Lean CLI生成的配置文件（.lean目录）里可自定义一些设置，包括数据目录等。如果不登录账户，有些云同步功能不可用但本地回测不受影响。请参考QuantConnect官方文档获取更详尽的使用指导。

通过以上步骤，本工作站即成为一个完整的量化研究与开发环境。用户可以编写Python算法策略，在本地通过Lean引擎回测验证策略效果，并为后续实时交易打下基础。

远程图形登录方案部署

由于工作站可能放置在异地机房或无直连显示器环境，为方便在远程使用完整的Ubuntu图形桌面界面，需要配置远程桌面访问方案。本方案采用轻量可靠的 **x11vnc** 服务配合 **Tailscale** VPN，实现随时随地通过VNC Viewer访问工作站的GUI桌面。

安装 x11vnc 并设置VNC服务

1. 安装 **x11vnc**：x11vnc 是Linux下的VNC服务器程序，可以把当前正在运行的X11图形会话共享出去。使用 APT安装：

```
sudo apt install -y x11vnc
```

安装完成后，可以手动启动x11vnc服务。

2. 设置VNC密码：出于安全，VNC需要密码验证。使用 **x11vnc -storepasswd** 命令创建密码：

```
x11vnc -storepasswd yourVNCpassword ~/.vnc/passwd
```

将“yourVNCpassword”替换为自定义密码。该命令会将密码以加密形式保存到 **~/.vnc/passwd** 文件。稍后x11vnc服务启动时可读取此文件进行认证。

3. 启动 **x11vnc** 服务：首先需要有一个活动的图形会话（DISPLAY）。如果工作站装的是Ubuntu Desktop版本，通常在物理机启动时GDM显示管理器会在 **:0** 启动登录界面。可以选择设置自动登录某个用户，这样系统引导后会直接进入桌面会话，确保有活动的 **:0** 显示供x11vnc使用。然后，在终端运行：

```
x11vnc -forever -loop -noipv6 -rfbauth ~/.vnc/passwd -rfbport 5900 -shared  
-display :0
```

参数说明：**-forever** 保持连接不断开重用、**-loop** 断线后自动重试、**-rfbauth** 指定密码文件、**-rfbport 5900** 监听端口5900（VNC默认端口）、**-shared** 允许多个客户端同时连接、**-display :0** 指定要共享的X11显示编号。这样x11vnc会附着当前屏幕，开启VNC服务。保持此终端运行状态不要关闭。现在即可通过VNC客户端连接机器IP的5900端口访问桌面。

4. 设置服务自启动（可选）：为了免去每次手动启动，可将x11vnc设置为systemd服务开机自启。在 **/etc/systemd/system/** 下创建文件 **x11vnc.service**，内容例如：

```
[Unit]  
Description=Start x11vnc at startup  
After=display-manager.service  
  
[Service]  
Type=simple  
ExecStart=/usr/bin/x11vnc -forever -display :0 -auth guess -passwd  
<YOURPASSWD> -loop -noxdamage -rfbport 5900 -shared  
ExecStop=/usr/bin/killall x11vnc
```

```
Restart=on-failure  
User=<YOURUSERNAME>
```

```
[Install]  
WantedBy=graphical.target
```

将其中的 `<YOURPASSWD>` 替换为VNC密码（或使用 `-rfbauth` 方式指定密码文件），`<YOURUSERNAME>` 替换为实际用户名。关键是 `-auth guess` 参数，它会让x11vnc自动尝试获取显示的X权限（对Ubuntu默认的GDM有效）。保存后执行：

```
sudo systemctl daemon-reload  
sudo systemctl enable x11vnc.service  
sudo systemctl start x11vnc.service
```

这样x11vnc将在系统图形界面启动后自动运行。如需停止服务可用 `systemctl stop x11vnc`。需要注意的是，如果屏幕处于锁定状态或未登录状态，x11vnc可能无法附着显示。这也是为什么建议启用自动登录或使用LightDM替代GDM（LightDM在未登录时也能让x11vnc附着）。另外，锁屏后VNC可能断开，因此可以在系统设置中关闭自动锁屏⁵⁰。出于安全考虑，不锁屏要确保网络本身安全可控。

部署 Tailscale 实现内网穿透

[Tailscale](#) 是零配置的VPN工具，可以让不同设备在Internet上组建虚拟内网，非常适合穿透NAT远程访问家用或云端设备。使用Tailscale可以免去配置路由端口转发，并加强连接的安全性。

1. 安装 Tailscale：根据Tailscale官方说明⁵¹ 安装客户端。在Ubuntu上，最快方法是使用他们的安装脚本：

```
curl -fsSL https://tailscale.com/install.sh | sudo bash
```

这会自动添加Tailscale源并安装最新版本。或者手动添加源后 `apt install tailscale` 也可。安装完成后，启动tailscale并登录：

```
sudo tailscale up
```

此命令会输出一个URL链接，用于在浏览器登录您的Tailscale账户（支持GitHub、Google等OAuth登录）。复制该链接在本地主机浏览器打开，完成认证并授权将当前工作站加入您的Tailnet网络。终端会显示 `Logged in successfully` 等信息⁵²。此后Tailscale会在后台运行，并将该工作站分配一个虚拟内网IP（通常以100.x.y.z形式）。

2. 配置 Tailnet ACL (可选)：默认情况下，同一Tailnet中的设备彼此可以直接通信。如果需要限制某些设备访问，可在Tailscale管理界面设置访问控制策略。一般家用或个人网络无需特别配置。
3. 获取工作站Tailscale IP：在工作站上执行：

```
tailscale ip -4
```

它会输出当前Tailscale分配的IPv4地址，如 100.100.50.10。也可以在其他已登录Tailscale的设备上通过 tailscale status 查看网络中所有节点及其IP。记下工作站的Tailscale IP地址。

4. **客户端连接**：在您本地电脑（已安装Tailscale并登录同一账户）上，使用任意VNC Viewer客户端（如 TightVNC, RealVNC, TigerVNC）建立连接。输入目标地址为工作站的Tailscale IP和端口，如：

```
100.100.50.10:5900
```

然后发起连接。因为两端通过Tailscale在同一虚拟网络，无需担心公网IP和防火墙复杂配置。正常情况下会弹出密码验证，输入之前设置的VNC密码，即可看到远程Ubuntu桌面画面并进行操作。

5. **防火墙配置**：如果Ubuntu本地开启了UFW防火墙，需要允许Tailscale流量或开放5900端口。可以执行：

```
sudo ufw allow 5900/tcp
```

这会允许所有接口的5900端口访问。如果只想允许Tailscale网络接口，可以设置规则

`ufw allow in on tailscale0 to any port 5900`。Tailscale本身自带防火墙规则（默认允许 Tailnet内互通），但本机的UFW也要相应放行⁵³。注意：尽管VNC有密码，仍建议将访问限制在安全的VPN环境下，不要直接暴露5900在公网。

6. **测试与性能**：成功连接后，测试在远程桌面中操作鼠标键盘是否流畅。通过Tailscale的P2P优化，一般在网络良好时会有接近本地局域网的体验。可以尝试打开应用、浏览网页，观察延迟和画质。如果感觉画面有压缩，可调整VNC Viewer的设置提高色深或分辨率。不过要兼顾带宽，一般默认压缩设置在办公场景下是足够清晰的。

通过x11vnc+Tailscale的组合，实现了方便且安全的远程GUI访问。相比传统将VNC直接暴露在公网，此方案利用了Tailscale的点对点加密隧道，不必担心中间人攻击和端口扫描风险。只要工作站和客户端都在线登陆Tailnet，无论两者身处何种网络，都可以方便地连接。

调试与常见报错处理记录

在部署工作站软硬件及远程访问过程中，可能会遇到一些问题。以下归纳常见报错场景及相应的解决方案：

- **DISPLAY 环境获取失败**：表现为在运行图形应用或x11vnc时出现“No display found”或类似错误。这通常是由于没有活动的图形会话或权限不足导致x11vnc无法附着显示屏。解决方法：首先确保目标用户已经登录桌面（可设置开机自动登录，避免无人值守情况下无活动会话）。对于x11vnc服务，使用`-display :0 -auth guess`参数可以让其自动获取当前显示的认证文件。如果仍失败，可手动指定Xauthority，例如：

```
x11vnc -display :0 -auth /run/user/1000/gdm/Xauthority -rfbauth ~/.vnc/  
passwd
```

其中路径需要根据实际情况调整（GDM的Xauthority文件路径可以通过`ps aux | grep Xauthority`查找）。另外，切换显示管理器也是一招，如果GDM在未登录时不允许外部连接，可改用LightDM，它在登录界面也提供虚拟显示，此时x11vnc可附着`:0`登录界面。总之，确保有有效DISPLAY且x11vnc有权限访问，即可解决此问题。

- **权限设置问题**：常见权限问题包括：运行某些命令需要sudo、访问硬件传感器或设备文件权限不足等。例如smartctl读取硬盘健康需要root，解决办法是加 sudo。再如Docker运行需要root权限，如果不想每次sudo，可将用户加入docker组（如前文已述），然后重新登录。这实际上就是解决权限问题的一种思路：**赋予程序或用户正确的权限**。对于浏览器无法启动的问题，有时是因为权限沙箱机制，例如使用Chrome/Chromium在root环境运行会被拒绝，这时可以加入参数 --no-sandbox 或者改为普通用户运行浏览器。归纳来说，仔细阅读错误信息中的“Permission denied”字样，定位哪个文件或资源受限，然后通过 ls -l 查看权限、使用 sudo 或 chmod/chown 调整权限。切记不要随意将敏感文件权限设为全可写，按最小权限原则处理。
- **端口占用冲突**：假如x11vnc启动时报端口被占用，或VNC Viewer连接时提示连接被拒绝，而实际上x11vnc已运行。这可能是**端口5900**已经有另一服务在监听，例如Ubuntu自带的remote desktop（vino服务）占用了。同一端口冲突会导致后启动的服务无法绑定。解决方案：用命令 sudo lsof -i TCP:5900 或 ss -tunlp | grep 5900 查看是哪个进程占用了5900端口。如果是不需要的残留VNC服务，可将其卸载或关闭。或者修改x11vnc监听端口，例如：

```
x11vnc -rfbport 5901 ...
```

这样客户端也对应连接5901端口。确保修改后的端口在Tailscale/防火墙中开放。一般5900以上的端口都可以，只要双方一致即可。

- **VNC连接失败（连接被拒绝）**：在VNC客户端报错“No connection could be made... actively refused by the target machine”（如下图所示）。这通常表示客户端已找到服务器IP但**连接被拒**，要么服务未运行、要么被防火墙拦截⁵³。对此应逐一排查：确认工作站上的x11vnc进程在运行（用 pgrep x11vnc 查看），如果未运行则启动服务；确认Tailscale正常连接并获取IP（tailscale status）；然后检查防火墙是否开放5900端口⁵³。若还是不行，可能是VNC服务仅绑定了本地接口而非0.0.0.0，可尝试在启动x11vnc时加 -listen 0.0.0.0 确保监听所有网络接口。另外，确保VNC密码正确无误。如果使用非Tailscale环境，连接被拒绝还可能是NAT穿透问题，此时Tailscale是最简方案。总之，“actively refused”基本可以定位为服务没真正跑通或被防火墙拒绝，按上述思路检查即可解决。
- **浏览器唤起失败**：在远程桌面中启动浏览器无响应，或在终端使用Puppeteer时出现**无法启动Chrome**错误。可能原因：没有硬件加速支持、沙盒冲突、缺少依赖库等。解决方案：
- 对于GUI桌面上直接打不开浏览器（Firefox/Chrome），可以尝试从终端启动看报错。如果提示 No protocol specified 或 Error: cannot open display :0，一般是DISPLAY环境问题，参考上面DISPLAY失败的解决处理。如果浏览器卡死，可以尝试禁用GPU加速：如Chrome加参数 --disable-gpu，Firefox启动安全模式等。
- 对于Puppeteer这类无头Chrome，如果报错包含 No usable sandbox，可在启动它时加 --no-sandbox 参数：例如在Puppeteer的 launch() 参数中设置 args: ['--no-sandbox', '--disable-setuid-sandbox']，这样Chrome不使用沙盒模式启动即可在root下运行（安全性略有降低，但在隔离环境下可以接受）。如果报错是缺少库，如 libatk-1.0.so.0: cannot open shared object file，说明系统缺乏Chromium运行需要的库。可通过APT安装一些依赖：

```
sudo apt install -y libatk1.0-0 libatk-bridge2.0-0 libdrm2 libxkbcommon-x11-0 libxcomposite1 libxrandr2 libgbm1 libpangocairo-1.0-0 libatspi2.0-0
```

（这里只列出常见依赖，实际按错误信息添加即可）。Ubuntu Desktop通常自带这些库，服务器精简版才可能缺少。所以在桌面版上问题不大。如果是在容器或Minimal环境遇到，按需补齐依赖即可。

- **其他调试技巧**：借助系统日志来定位问题。例如驱动问题可看 `dmesg` 输出，Docker问题看 `sudo journalctl -u docker`。远程连接问题可看 `tailscale status` 等。每当遇到报错，养成查看具体错误信息并善用搜索引擎查询的习惯。通过细心分析和正确配置，大多数问题都能迎刃而解。

以上列举的问题及解决方案，涵盖了从硬件连接、网络访问到软件运行的常见坑点。随着系统持续运行和更多应用部署，可能还会出现新的挑战，但通过及时记录和分析日志，可不断完善工作站的稳定性。

当前系统运行状态评估

经过一系列安装和配置，本工作站已投入运行。以下对其当前状态从性能和稳定性方面进行评估：

图：使用 `htop` 实时监控系统资源。屏幕中显示32线程CPU的使用率、各核心负载以及内存占用等。绿色和橙色柱状表示当前各核心的利用率，右侧为内存和交换分区使用情况。

- **CPU使用率**：如上图所示，工作站搭载的多核AMD CPU空闲时各核心利用率非常低，仅有0-5%波动，整体负载很轻（load average约0.2）。在运行日常开发任务（如编辑代码、轻度编译）时，CPU占用会瞬间升高但因为核心众多，仍能保持充裕的闲置资源。例如同时启动多个编译线程，负载可能上升到2.0左右（表示有两个核心满载），此时从`htop`看到只有部分柱状接近100%。这意味着系统有很大富余计算能力，可支持将来的密集型AI训练或回测任务。当进行深度学习训练时，可监控 `nvidia-smi` 输出GPU利用率与显存占用，以判断是否存在CPU瓶颈。如果GPU饱和而CPU仍闲，则CPU性能足够；反之若CPU达到100%持续，则需要优化算法以利用GPU加速或采用更强CPU。
- **内存和存储**：当前内存使用约为5.9 GiB/32 GiB（18%）【26†】，这包括启动图形界面、一些常驻服务（tailscale、vnc、后台进程等）以及运行的终端和浏览器。尚有大量空闲内存可供分配，说明即便加载较大数据集或启动虚拟机也仍有余量。Swap交换分区使用为0，表示内存尚未吃紧。当后续进行大规模数据处理或模型训练时，应密切关注内存占用，尽量避免进入swap以防性能急剧下降。NVMe SSD的IO性能优秀，目前系统盘I/O等待很低，没有成为瓶颈。通过 `iostat` 或GNOME系统监视可以看到磁盘读写延迟基本在毫秒级。SMART自检也未见错误，硬盘健康良好。
- **温度与功耗**：借助 `sensors` 命令可以读到当前CPU温度在空闲时大约**30-40°C**左右（具体取决于室温和水冷散热效能）。在持续高负载例如运行stress测试满载CPU时，温度可上升到70°C以上但依然在安全范围（水冷可以有效控制在80°C以下）。显卡在空闲桌面状态下温度约**40-50°C**，风扇可能处于停止或低速模式（部分新GPU在低负载时风扇停转）。GPU满载运行比如深度学习训练一段时间后，温度可能上升到70-75°C，此时风扇加速以维持温度稳定。电源功耗方面，整机待机功耗估计在50-70W左右；运行一般应用时可能100W上下；如GPU满载则整机功耗接近300W甚至更高（具体取决于GPU功耗设计）。可以使用测功仪插座测量墙上功耗，或通过UPS管理软件读取即时瓦数。如果主板提供了VRM电流监测，`sensors` 输出中可能有CPU Package功耗，观察其在不同负载下的变化也能推算整机功耗。当前状态下，工作站功耗低、噪音极小（水冷泵和风扇在低速运行，几乎无声），适合长时间连续运行。
- **网络与VNC性能**：在本地局域网或通过Tailscale的虚拟局域网环境下，远程桌面操作响应迅速。窗口拖动和菜单展开略有延迟但在可接受范围，没有明显卡顿。VNC传输对带宽要求较高，例如1080p分辨率下每秒可能消耗几兆比特，实际使用中发现只要网络稳定，操作Office文档、编程等场景基本流畅。若远程进行高清视频播放或3D界面，VNC会出现帧率下降和压缩痕迹，这是正常的，因为VNC并非针对多媒体优化。对于需要高性能图形的远程应用，可以考虑使用XPRA或NoMachine等方案，但日常管理来说x11vnc已经足够。
Tailscale的表现非常出色，它智能选路，让异地连接延迟维持在几十毫秒量级。例如从外地通过Tailscale连接本工作站，ping延迟约50ms，VNC操作延迟接近肉眼不可察觉。工作站启用Tailscale后，对其他开发用途也很有帮助，比如可直接SSH登录 `100.x.y.z` 管理，或通过Tailscale IP进行Jupyter等服务的访问，安全且便捷。

综上，目前系统运行平稳，资源占用健康，各硬件监测指标均在正常范围。水冷散热保证了高负载下温度稳定，电源有充足冗余确保瞬时功耗变化，远程访问方案也验证了可靠性。接下来，可以安心投入实际的金融数据研究和AI建模任务。

下一步规划与展望

搭建好硬件和基础软件环境只是第一步。针对“金融/AI投资系统”这一目标，后续还需逐步完善上层应用和策略开发。下面对未来的规划方向做一个展望：

- **数据采集**：构建统一的数据获取模块，持续收集所需的金融市场数据。例如通过API抓取股票行情、成交量、财报数据，或者爬取新闻资讯、社交媒体情绪等非结构化数据。可利用现有数据源和框架：如Yahoo Finance API、聚宽数据接口、Tushare（针对中国市场）等获取历史和实时行情；使用Python的 `yfinance`、`ccxt` 库分别获取股票和加密货币数据。将数据保存到本地数据库或文件（CSV/Parquet）中，以供策略回测和训练模型使用。后续可以搭建一个小型数据仓库，存放不同市场、不同频率（日线、分钟线、tick级别）的数据，并定期更新新增量数据。
- **策略回测**：利用QuantConnect Lean引擎或其他回测框架开发和验证投资策略。不同资产类别（股票、期货、加密货币等）和不同频率的策略都可以在本地模拟交易，衡量收益率、风险指标和交易费用影响。Lean CLI已经在本工作站部署，可编写Python算法来定义选股条件、买卖信号和仓位管理规则，然后通过历史数据运行回测。短期计划可以尝试简单均线策略、均值回归策略等验证环境配置，随后逐步实现更复杂的Alpha策略或机器学习模型策略。回测结果用可视化报告展现，从中评估策略的夏普比率、最大回撤等性能指标。也可以尝试使用 `backtrader`、`zipline` 等Python回测库做对比分析。通过不断迭代回测，筛选出有潜力的策略方案。
- **自动交易**：当某些策略在回测中表现稳健，可以考虑连接实盘进行自动交易。这需要集成交易接口，例如通过券商API（如Interactive Brokers的API）、交易所API（如Binance API）等下单。QuantConnect Lean也支持实时交易模式，可利用已有代码转入实盘。未来可能搭建一个交易机器人，监控实盘行情，当触发策略信号时自动下单，并做好风控（止盈止损）。需要注意实盘环境的延迟和滑点，与回测假设不同，要做好参数调整。为了安全起见，初期可使用虚拟资金或模拟账户测试，待确认策略有效且交易模块稳定，再逐步投入真实资金。在技术实现上，可以使用Lean CLI的 `lean live` 功能来连Broker，在本地执行；或使用Python的 `ib_insync` 库直接对IB下单，`ccxt` 库对币圈交易所下单等。这部分工作需要谨慎测试，以确保交易逻辑严谨不出差错。
- **知识库整合**：搭建一个金融知识库，汇总研究过程中积累的见解、资料和模型。这个知识库可以包括：研究笔记（markdown文档）、学术论文和行业报告、已经验证的策略成果、以及数据字典（例如特征含义、因子解释等）。可以考虑使用Wiki系统（如基于Git的MkDocs或直接使用Astro框架搭建站点）来组织这些内容，使其易于查询和逐步扩充。另外，引入**矢量数据库**技术，将文本型研究内容嵌入向量以便语义检索。这样日后如果使用自然语言提问，能从知识库中找到相关内容。这为下一步的GPT辅助奠定基础。
- **GPT辅助提示**：利用大语言模型（如OpenAI GPT-4或本地大模型）来辅助金融研究和交易决策。具体构想包括：让GPT读取上述知识库，充当一个金融研究助手，可以用对话方式询问策略表现、某指标含义等，GPT从知识库提取信息并给出解释。还可以让GPT辅助代码生成，例如编写策略代码的框架，或解析报错信息建议调参。在投资决策上，GPT能基于新闻事件总结市场情绪，或者对量化因子组合给出解释性分析。当然GPT目前主要是辅助，不能直接代替严谨的数理分析，但可以极大提高研究效率和提供创意想法。实现上，可以使用LangChain框架，将本地文档作为GPT的上下文，让GPT回答专属领域问题；或者调用OpenAI的API，对接自有数据问答系统。随着隐私和合规考虑，或许也会评估部署本地大模型（如LLaMA系列）以保证数据不泄露。无论如何，引入AI助手将使工作站更具智能化特色。
- **网页平台部署**：最终为了整合以上所有功能，计划开发一个Web前端界面，让用户可以方便地通过浏览器使用整个系统。这个**投资系统平台**将包括：行情仪表盘（实时价格图表、指标监控）、策略管理模块（上传策

略代码、回测参数配置、一键启动回测并展示结果曲线）、交易监控模块（显示当前持仓、订单状态、收益曲线）以及知识库查询界面和GPT聊天窗口。前端可采用现代Web技术构建，例如React或Vue框架+Ant Design组件实现友好的交互界面。后端可以用Python的FastAPI或Node.js等提供REST API，与前端通信。后端负责调度回测任务（调用Lean CLI或内部引擎）、获取数据库数据、处理交易信号下单、调用GPT服务等。将系统部署成一个持续运行的服务（可用Docker Compose编排多个服务：数据库、后端、前端、AI服务等）。这样一来，整个金融/AI工作站就升级为一个完整的软件产品，用户只需通过浏览器登录，即可在任何设备上远程使用强大的量化分析和自动交易工具。

展望未来，这套系统有潜力发展成一个自主的量化投资平台，不仅满足个人研究，也可以扩展团队协作。在搭建过程中，不断会有新的技术和想法融入，比如引入实时风控系统、交易日志分析、智能报告生成等。随着实践推进，我们将持续完善硬件性能和软件算法，让工作站发挥最大价值，为金融投资决策提供科学可靠的支持。

（本手册内容完，以上所有章节按*Markdown*格式编写，可供下载保存以供日后查阅。） 54 26

1 2 3 4 5 如何正确安装一体式CPU水冷散热器？装在机箱的不同位置有什么差别？

<https://www.teamgroupinc.com/community/cn/blog-detail/liquid-cpu-cooler-guide/>

6 7 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 54 图文指南：如何安装 Ubuntu 24.04 LTS 桌面版或服务器 - 系统极客

<https://www.sysgeek.cn/install-ubuntu-24-04-lts/>

8 25 win10+ubuntu24.04，在ubuntu24.04中安装Nvidia驱动程序 _ubuntu24.04安装nvidia驱动-CSDN博客

<https://blog.csdn.net/SharbyBai/article/details/142621606>

26 Amd and Linux chipset drivers - Unix & Linux Stack Exchange

<https://unix.stackexchange.com/questions/378976/amd-and-linux-chipset-drivers>

27 NVMe_Support – smartmontools

https://www.smartmontools.org/wiki/NVMe_Support

28 29 30 31 Lm-sensor - Ubuntu中文

<https://wiki.ubuntu.org/Lm-sensor>

32 33 34 35 36 37 38 39 PDF_LANGUAGE_GENERATION_GUIDE.md

<file:///S/www/CMc6W9GtEqGbDDVMNK>

40 41 42 43 44 46 47 48 49 Installing Lean CLI - QuantConnect.com

<https://www.quantconnect.com/docs/v2/lean-cli/installation/installing-lean-cli>

45 Install on Linux - QuantConnect.com

<https://www.quantconnect.com/docs/v2/local-platform/installation/install-on-linux>

50 How to install x11vnc vnc server as a service on Ubuntu 20.04, for remote access or screen sharing -

Crazy Logic

<https://www.crazy-logic.co.uk/projects/computing/how-to-install-x11vnc-vnc-server-as-a-service-on-ubuntu-20-04-for-remote-access-or-screen-sharing>

51 52 Ubuntu Tailscale如何实现远程访问 - 问答 - 亿速云

<https://www.yisu.com/ask/90704262.html>

53 LEAP 15.0: vncviewer unable connect to socket: Connection refused ...

<https://forums.opensuse.org/t/leap-15-0-vncviewer-unable-connect-to-socket-connection-refused-111/139647>