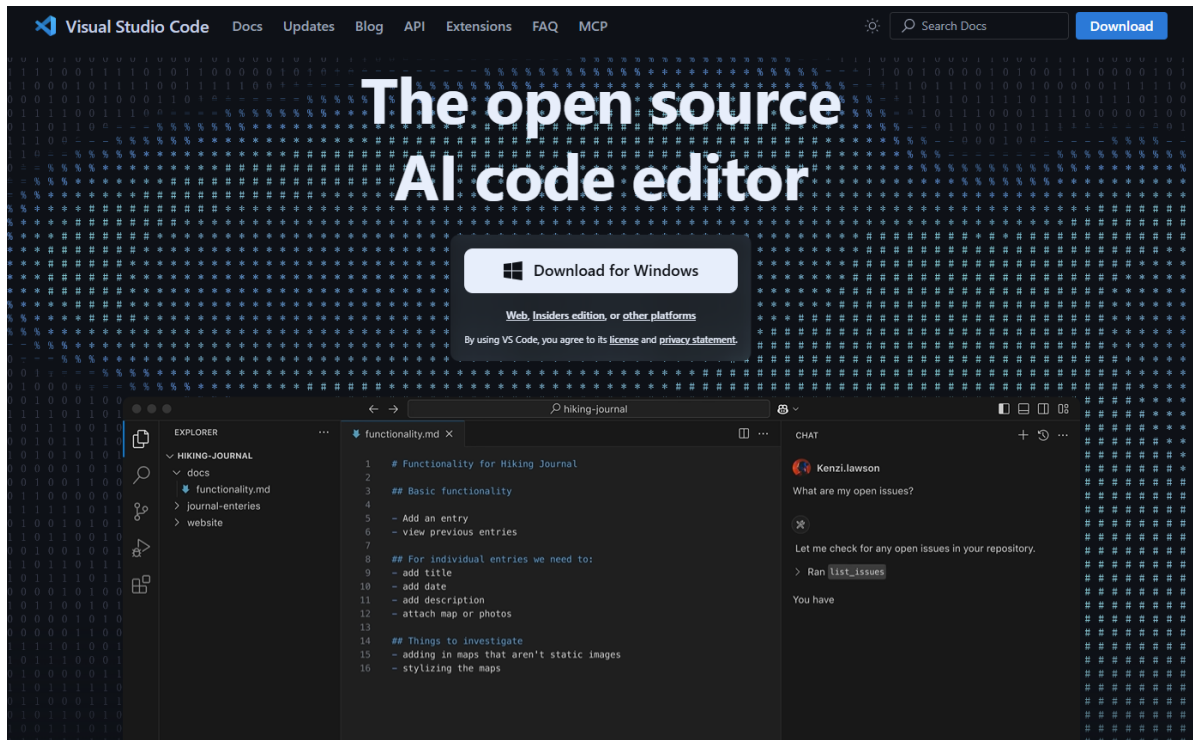


3、ROS2集成开发环境搭建

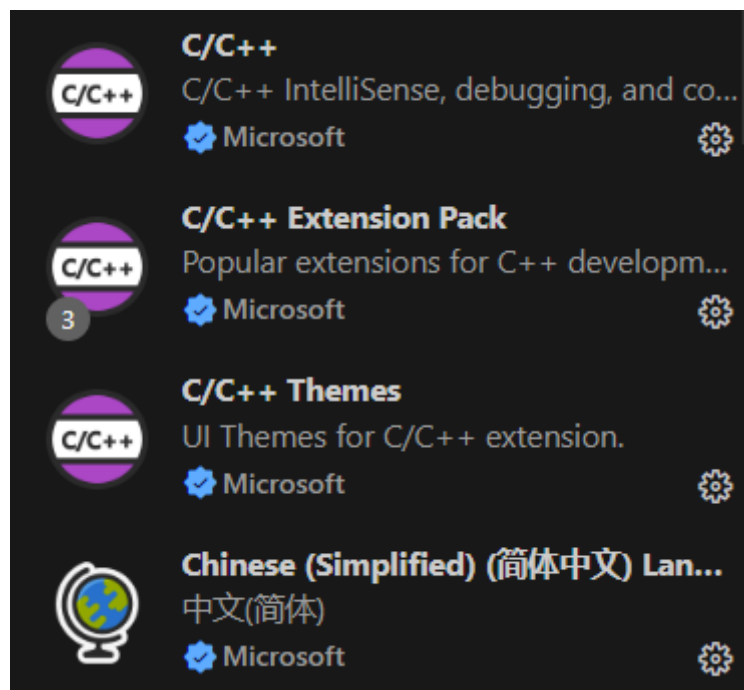
理论上，在ROS2中，只需要记事本就可以编写基本的ROS2程序，但是为了提高开发效率，可以安装集成开发环境vscode。

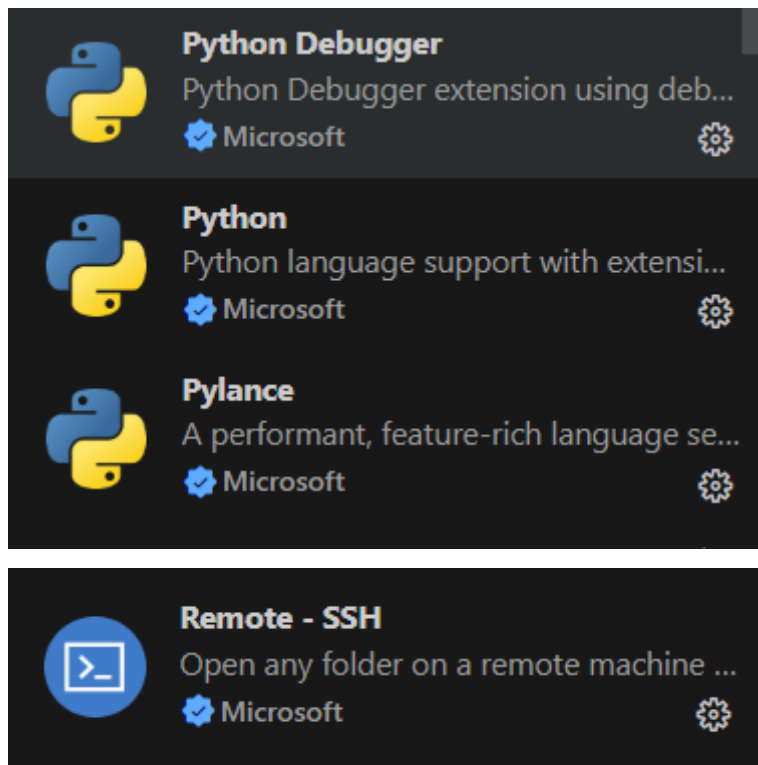
1、使用vscode开发环境

- 浏览器搜索vscode，选择对应系统平台的安装包，下载安装即可



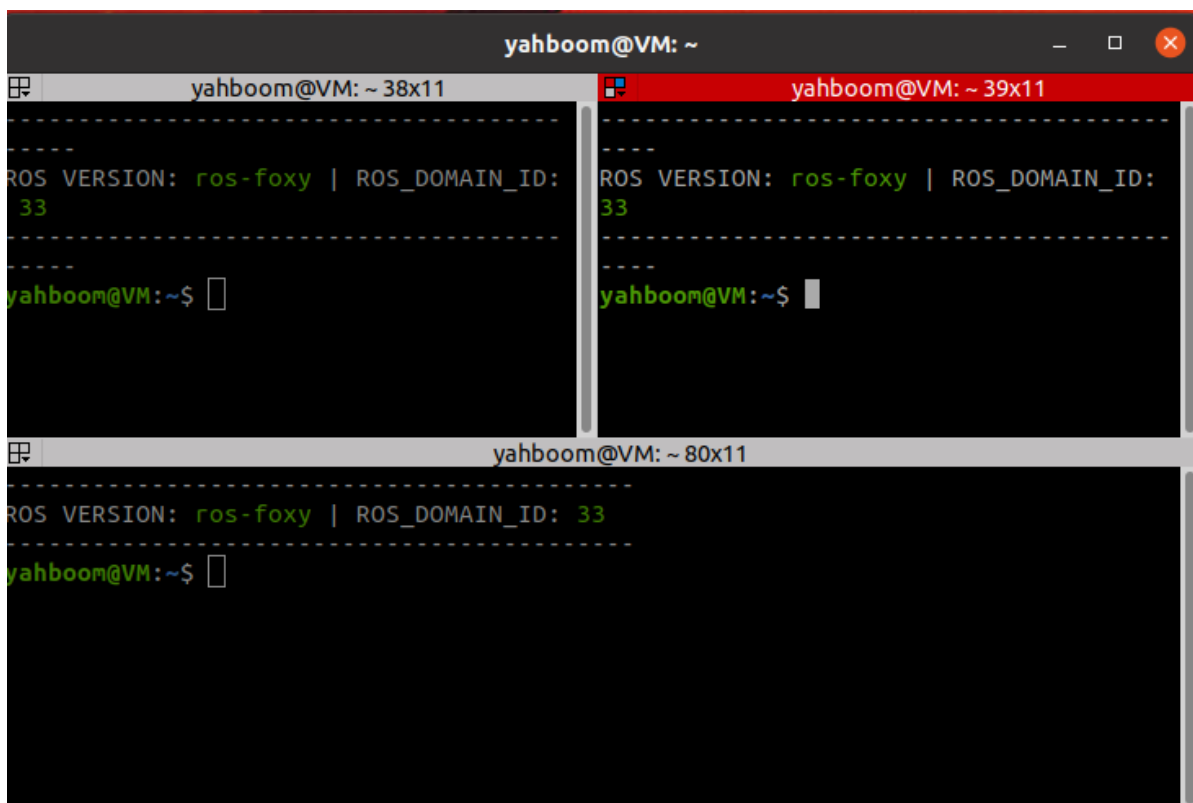
- 安装常用插件，帮助提升工作效率





2、使用终端

在ROS2中，需要频繁的使用到终端，推荐一款较为好用的终端：Terminator，效果如下：



2.1、安装

```
sudo apt install terminator
```

2.2、启动

快捷键`Ctrl+Alt+T`启动

2.3、Terminator 常用快捷键

关于在同一个标签内的操作:

Alt+Up	//移动到上面的终端
Alt+Down	//移动到下面的终端
Alt+Left	//移动到左边的终端
Alt+Right	//移动到右边的终端
Ctrl+Shift+O	//水平分割终端
Ctrl+Shift+E	//垂直分割终端
Ctrl+Shift+Right	//在垂直分割的终端中将分割条向右移动
Ctrl+Shift+Left	//在垂直分割的终端中将分割条向左移动
Ctrl+Shift+Up	//在水平分割的终端中将分割条向上移动
Ctrl+Shift+Down	//在水平分割的终端中将分割条向下移动
Ctrl+Shift+S	//隐藏/显示滚动条
Ctrl+Shift+F	//搜索
Ctrl+Shift+C	//复制选中的内容到剪贴板
Ctrl+Shift+V	//粘贴剪贴板的内容到此处
Ctrl+Shift+W	//关闭当前终端
Ctrl+Shift+Q	//退出当前窗口, 当前窗口的所有终端都将被关闭
Ctrl+Shift+X	//最大化显示当前终端
Ctrl+Shift+Z	//最大化显示当前终端并使字体放大
Ctrl+Shift+N or Ctrl+Tab	//移动到下一个终端
Ctrl+Shift+P or Ctrl+Shift+Tab	//Ctrl+Shift+Tab 移动到之前的一个终端

关于各个标签之间的操作:

F11	//全屏开关
Ctrl+Shift+T	//打开一个新的标签
Ctrl+PageDown	//移动到下一个标签
Ctrl+PageUp	//移动到上一个标签
Ctrl+Shift+PageDown	//将当前标签与其后一个标签交换位置
Ctrl+Shift+PageUp	//将当前标签与其前一个标签交换位置
Ctrl+Plus (+)	//增大字体
Ctrl+Minus (-)	//减小字体
Ctrl+Zero (0)	//恢复字体到原始大小
Ctrl+Shift+R	//重置终端状态
Ctrl+Shift+G	//重置终端状态并clear屏幕
Super+g	//绑定所有的终端, 以便向一个输入能够输入到所有的终端
Super+Shift+G	//解除绑定
Super+t	//绑定当前标签的所有终端, 向一个终端输入的内容会自动输入到
其他终端	
Super+Shift+T	//解除绑定
Ctrl+Shift+I	//打开一个窗口, 新窗口与原来的窗口使用同一个进程
Super+i	//打开一个新窗口, 新窗口与原来的窗口使用不同的进程

3、使用git

3.1、安装

日常工作中，因为都是团队协作，且涉及版本管理，所以git是绕不开的技能。git 是一个免费和开源的分布式版本控制系统，在Ubuntu下安装git：

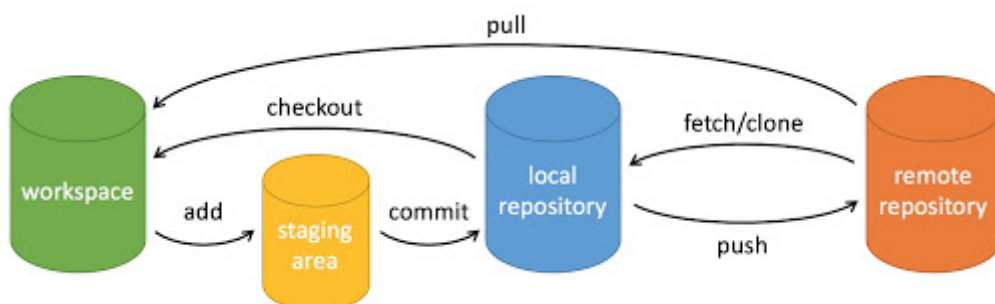
```
sudo apt install git
```

3.2、Git 基本操作

Git 的工作就是创建和保存你项目的快照及与之后的快照进行对比。

本章将对有关创建与提交你的项目快照的命令作介绍。

Git 常用的是以下 6 个命令：**git clone**、**git push**、**git add**、**git commit**、**git checkout**、**git pull**，后面我们会详细介绍。



说明：

- workspace：工作区
- staging area：暂存区/缓存区
- local repository：版本库或本地仓库
- remote repository：远程仓库

一个简单的操作步骤：

```
$ git init
$ git add .
$ git commit
```

- git init - 初始化仓库。
- git add . - 添加文件到暂存区。
- git commit - 将暂存区内容添加到仓库中。

3.2.1、创建仓库命令

下表列出了 git 创建仓库的命令：

命令	说明
<code>git init</code>	初始化仓库
<code>git clone</code>	拷贝一份远程仓库，也就是下载一个项目。

3.2.2、提交与修改

Git 的工作就是创建和保存你的项目的快照及与之后的快照进行对比。

下表列出了有关创建与提交你的项目的快照的命令：

命令	说明
<code>git add</code>	添加文件到暂存区
<code>git status</code>	查看仓库当前的状态，显示有变更的文件。
<code>git diff</code>	比较文件的不同，即暂存区和工作区的差异。
<code>git commit</code>	提交暂存区到本地仓库。
<code>git reset</code>	回退版本。
<code>git rm</code>	将文件从暂存区和工作区中删除。
<code>git mv</code>	移动或重命名工作区文件。
<code>git checkout</code>	分支切换。
<code>git switch</code> （Git 2.23 版本引入）	更清晰地切换分支。
<code>git restore</code> （Git 2.23 版本引入）	恢复或撤销文件的更改。

3.2.3、提交日志

命令	说明
<code>git log</code>	查看历史提交记录
<code>git blame <file></code>	以列表形式查看指定文件的历史修改记录

3.2.4、远程操作

命令	说明
<code>git remote</code>	远程仓库操作
<code>git fetch</code>	从远程获取代码库
<code>git pull</code>	下载远程代码并合并
<code>git push</code>	上传远程代码并合并

关于更多git工具的使用可以在终端下输入：`git --help` 查看帮助文档