

Cloud Studio (云端 IDE) 操作指南





【版权声明】

©2013-2025 腾讯云版权所有

本文档(含所有文字、数据、图片等内容)完整的著作权归腾讯云计算(北京)有限责任公司单独所有,未经腾讯云 事先明确书面许可,任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成 对腾讯云著作权的侵犯,腾讯云将依法采取措施追究法律责任。

【商标声明】

🔗 腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算(北京)有限责任公司及其关联公司所有。本文档涉及的第三方主体的 商标,依法由权利人所有。未经腾讯云及有关权利人书面许可,任何主体不得以任何方式对前述商标进行使用、复 制、修改、传播、抄录等行为,否则将构成对腾讯云及有关权利人商标权的侵犯,腾讯云将依法采取措施追究法律责 任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况,部分产品、服务的内容可能不时有所调整。 您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定,除非双方另有约定,否则, 腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务,及相应的技术售后服务,任何问题请联系 4009100100或 95716。



文档目录

操作指南

创建使用和管理工作空间 通过 SSH 连接云端 IDE 工作空间 快速导入代码仓库 在线预览调试 常用中间件安装 使用 GPU 开发 模板使用指南 C++ 使用指南 C++ 使用指南 Node.js 使用指南 Vue-TypeScript 模板使用指南 Quana 模板使用指南 混元 DiT 模板使用指南 QwQ 32B 使用指南



操作指南 创建使用和管理工作空间

最近更新时间: 2024-12-25 17:00:13

一个工作空间是一个虚拟计算单元,它包含独立的存储、计算资源以及开发环境。Cloud Studio 是以工作空间来 组织的,本文为您介绍如何创建工作空间。

创建空间

- 1. 登录 腾讯云官网控制台 ,单击**立即使用**进入 Cloud Studio 云端 IDE,可以通过两种方式创建工作空间。
- 2. 创建工作空间。
 - 第一种方式:单击空间模板内的模板卡片直接创建工作空间,可以自动生成工作空间名称,并运行模板的预置环境及样本代码。创建成功后默认会打开新页面。



○ 第二种方式:单击通用工作空间,进入工作空间创建页面。单击新建选择预置环境,填写工作空间名、描述,并选择运行环境和代码来源。配置完成后单击创建,即可完成工作空间的创建。



新建工作空间			
空间名称 *			
空间描述			
代码来源*			0 / 255
● 导入仓库 🌔 仓库地址 🕒 空	间模版 空	2	
选择仓库服务商	管理账号 亿		
	du		
开发环境 *		版本	
All in One		full 1.0.0	
规格配置 *			
免费版 1核2GB内存 / 4GB 存储			~
新建 取消			

配置项	说明
工作空间名	您的工作空间的唯一标识,只能由字母、数字、下划线(_)、中划线 (-)、点(.)组成,不能包含空格或其它字符。
描述	对该工作空间作用的描述。
运行环境	工作空间内代码运行的环境,您可以选择预置环境,包含 Ubuntu、 Python、Java 和 Node.js 四种。
代码来源	工作空间内的代码来源,此处我们选择"空",即不添加任何代码。

您还可以创建代码来自于 Git 仓库的工作空间,代码会被自动克隆到工作空间,请参考使用 Git 进行版本控制。

工作空间的使用

您可以在 Cloud Studio 云端 IDE 的工作空间内存放自己的项目代码,安装所需要的软件环境,运行或编译项目。



界面简介

工作空间是我们主要的工作区域,主要由顶部菜单栏、左侧操作面板、右侧代码编辑区和底部状态栏组成。 您可以根据自己的习惯设置界面外观、偏好,安装自己需要的插件。

需要注意的是,您的偏好设置和插件在每个工作空间中是互相隔离的,也就是说您可以给不同的工作空间设置不同的 偏好,安装不同的插件。

S	文件 编辑 选择 查看 转到 运行 终端	帮助 HelloWorld:vue - RemoteWorking - Cloud Studio	
ß	资源管理器 …	♥ HelloWorld.vue x	ů 🕀 🗉 …
		src > components > 🔻 HelloWorld.vue > {} "HelloWorld.vue" > 🔗 template	
	> .vscode	1 <template></template>	
	> node_modules	2 <div class="hello"></div>	
	> public	3 <h2+{ h1="" msg="" }}<=""></h2+{>	
۲ β	√ src	4 cp2 5 For a quide and recipes on how to configure / customize this project. 	
	> assets	6 Check out the	in the second se
\square	✓ components	7 vue-cli documentation .	
	V HelloWorld.vue	8	
	V App.vue	9 <h3-installed cli="" h3="" plugins<=""></h3-installed>	
Б	JS main is	10 <u> 10 <u<< td=""><td></td></u<<></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u></u>	
	♦ gitignore	12 a href="https://github.com/uejs/ue-cli/tree/dev/packages/%40ue/cli-plugin=eslint" target=" blank" rel="noopener">eslint-/li>	
9	B babel config is	13 //ul>	
	{} package ison	14 <h3>Essential Links</h3>	
	README.md	15 dub 15 Juizs brofallstaas//www.aaratallstaabl.col=laanaanariisfara Barr/az/lis	
Ľø	IS yue config is	10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10 <1/> 10	
	4. varn.lock U	18 Community Chat//i>	
8	- junneen	<pre>19 Twitter</pre>	
		<pre>20 <a href="<u>https://news.vuejs.org</u>" rel="noopener" target="_blank">News</pre>	
		21 <td></td>	
		a	
	◇ 大纲	24 <pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	
	V {} "HelloWorld.vue"	25 <pre>vuex</pre>	
	✓	26 26 26 a href="https://github.com/vuejs/vue-devtools#vue-devtools" target="blank" rel="moopher">vue-devtools 	
	✓ Ø div.hello	27 27 27 <27 < 28 29 21 < 27 28 21 29 21 <	
	10 bi		
	v ⊗ p	问题输出终于调试控制台	2: node \vee $+$ \vee \square \square \wedge \times
	Ø br		
	⊗ a		
	69 h3	App running at:	
	∨ ⊗ ul	- Local: http://localhost:8080/	
	AII	It seems you are running Vue CLI inside a container.	
	✓时间线 HelloWorld.vue -□ ひ…	Access the dev server via http://localhost: <your container's="" external="" mapped="" port="">/</your>	
	♀ Initial commit. 杜茗煜 1 周	Note that the development build is not optimized.	
Q		to create a production build, fun yafn build.	
0			
≫ ha	ppy1183.cloudstudio.net 🖇 master* 🕂 o CL	⑧ 0 ▲ 0 ✿ tabnine ➤ 行 1, 列 1 空格: 2 UTF-8 LF Vue Setup CodeLingo Notebooks ⊘ E	SLint ⊘ Prettier Layout: U.S. Formatting: × 🗘

环境的使用

如上文所述,每个工作空间背后都包含一个完整的云端开发环境,您可以在里面查看或创建文件,也可以安装一些需 要的软件。

我们可以通过终端来进行这些操作,点击菜单栏**终端 > 新终端**,会在底部打开一个面板,点击底部终端切换。



5	文件 编辑 选择 查看 转到 运行	终端帮助	HelloWorld.vue — RemoteWorking — Cloud Studio
ß	资源管理器	新终端	^ <u>0</u> `
	∨ REMOTEWORKING	拆分终端	^夺5 lelloWorld.vue > { } "HelloWorld.vue" > 🔗 template
ر ب	 > vscode > node_modules > public > src > suptract 	运行任务… 运行生成任务… 运行活动文件 运行活动文本	otho
å	 ✓ components ✓ HelloWorld.vue 	▲17月122文本 	<pre> . une /https://cli.vuejs.org" target="_blank" rel="noopener">vue-cli documentation. led (LL Plugins </pre>
₿	♥ App.vue J5 main.js	重后止在运行的任务 终止任务… 	ref="https://github.com/vuejs/vue-cli/tree/dev/packages/%40vue/cli-plugin-babel" target="_blank" rel="noopener"
	 grtignore B babel.config.js package.json 	配置任务… 配置默认生成任务	ei= <u>https://github.com/vuejs/vue-cti/tree/dev/packages/%40vue/cti-ptugin-estint</u> target= _btank
€ ₩	() README.md JS vue.config.js ↓ yarn.lock	U 13 16 17 19 20 21 • 22 23 24	<pre> </pre>
	\lor {} "HelloWorld.vue" \lor \diamondsuit template \lor \diamondsuit div.hello \diamondsuit h1 \lor \diamondsuit p \diamondsuit br \diamondsuit a \diamondsuit h3 \lor \diamondsuit ul	问题 输 → Remote	出 终端 调试控制台 Norking ::(master) X [

我们就可以在终端中使用命令行来进行一些操作了,例如下图中创建了一个 README.md 文件,并全局安装了 vue。





上传与下载文件

如有较小的文件需要上传与下载,可以在本地文件系统选中文件后拖拽到编辑器左侧的文件树区域即可完成上传,在 文件树区域右键可选择下载。

由于编辑器对大型文件的上传与下载的优化有限,建议大型文件通过网络完成上传与下载,如使用 git push 与 git clone。

管理工作空间

在 Cloud Studio 云端 IDE 的工作空间列表页面,您可以运行、停止、删除和恢复工作空间。

运行

单击对应的工作空间卡片,就会在新的页面打开并运行该空间,此时该工作空间卡片上会显示"运行中"状态。

停止

对于处在"运行中"状态的工作空间,单击卡片右边的停止图标 💿 ,就可以停止运行该工作空间。





删除

您可以删除未运行的工作空间,单击工作空间卡片右下角的删除即可删除。

😑 🕜 Cloud Studio		<u></u>
韶 通用工作空间 ①	全部空间 近期删除	新建
🕝 高性能工作空间 📶 🕧		描序: 量近创建 ▼ 视图: 列表 ▼
 ⊡ 空间模板 ○ 应用推荐 	AllinOne-ogoeui 【管无描述】	创建时间 最近使用 标准版 2C 员 × 2024-06-04 几秒前
最近使用 Pytorch2.0.0 bpe【 高性能 ● 运行中 AllinOne-ogoeui 几秒前		点击 弹出删除

恢复

为了防止误删除,已删除的工作空间会展示在下方"已删除的工作空间"列表中,保留24小时。在此之前您可以随 时单击"恢复"图标,还原您的工作空间,超过 24 小时未恢复的工作空间将被永远销毁。





通过 SSH 连接云端 IDE 工作空间

最近更新时间: 2025-01-15 11:55:22

腾讯云

<u>小 注意:</u>

该能力当前仅支持通用工作空间,高性能工作空间暂不支持通过 SSH 连接。

Cloud Studio 工作空间(预置环境)支持 SSH 和 SCP,您可以通过 SSH 连接到工作空间,包括通过命令行终 端 SSH 连接工作空间,通过 VSCode 的 Remote - SSH 插件连接工作空间等,也可以通过 SCP 命令上传或 下载文件。

1、获取 SSH 命令

"运行中"的工作空间,可以看到 SSH 登录的小图标,单击图标,可以获取 SSH 登录命令。如下图所示:



2、通过 SSH 命令登录工作空间



您可以通过常见 SSH 登录工具登录到工作空间,比如 Mac 上的 iTerm2, Windows 上的 SecureCRT 等。

通过 iTerm2 登录工作空间

将获取到的 SSH 命令粘贴到 iTerm2 直接执行即可,如下图:



通过 SecureCRT 登录工作空间

1. 从前端获取 SSH 登录命令后,可以从命令里获取 Hostname 和 Username,参考下图设置 SecureCRT 连接选项:



2. 在上图选择"PublicKey",然后单击"Properties..."进行创建,单击"Create Identify File..."。如下 图:

腾讯云



Public Key Properties		
<u>U</u> se global public key setting Global <u>s</u> ettings	○ Use session public <u>k</u> ey setting	OK
Use identity or certificate file		Cancer
\bigcirc Use a certificate from your personal CAPI store or a PK	<pre><cs #11="" dll<="" pre="" provider=""></cs></pre>	
CAPI <u>D</u> LL;		
Certificate to use: <pre></pre>		
Get username from certificate: Common Name	~	
Use certificate as raw SSH2 key (server does not suppo	ort X.509)	
Fingerprint:		
Create Identity File Upload Export Public	Key Change <u>P</u> assphrase	

3. 按 SecureCRT 的指引进行操作,创建成功后,可以看到已创建的密钥对的相关信息:



Public Key Properties		
Use global public key setting Ouse session public	<u>k</u> ey setting	ОК
• Us <u>e</u> identity or certificate file		Cancel
C:\Users\!Documents\Identity		
\bigcirc Use a certificate from your personal CAPI store or a PKCS #11 provider DLL		
CAPI V DLL;		
Certificate to use; <pre><try all="" certificates=""></try></pre>		
Get user <u>n</u> ame from certificate: Common Name \sim		
✓ Use certificate as raw SSH2 key (server does not support X.509)		
Fingerprint:	/	
SHA-2: base64: SisePAtg DrmCHZF2alBMA= SHA-2: 4a:2b:1e:3c:0b:6 53:87:45:d7:eb:c0:67:96 SHA-1: 29:64:20:8c:08:(a1:35:28:2a:1d:a8 MD5: a6:a9:10:f6:c0:6c:bc:d7:4b:0b:d0:f0:84:c9:f2:b2	i:a3:ab:	
Create Identity File Upload Export Public Key Change Pas	sphrase	

4. SecureCRT 连接成功后,效果如下图:

Ŀ	Enter host <al< th=""><th>lt+R> 🖸 🗂 樠 🖨 🌣 📾 🍞 🧖 🜃</th></al<>	lt+R> 🖸 🗂 樠 🖨 🌣 📾 🍞 🧖 🜃
Ac	Session Manager 🛛 📮 🗙	✓ ~ ×
ctive Sessions	 ✔ □ + X □ × Filter by folder/session n ✓ Sessions h: q.ssh.clouds 	CloudStudio → ~ ls RemoteWorking CloudStudio → ~

3、通过 VSCode Remote SSH 插件连接工作空间



1. 给 VSCode 安装 Remote SSH 插件:



2. 在 Remote SSH 插件中增加一个连接,粘贴获取到的 SSH 命令,然后回车:



3. 单击远程主机名右边的按钮,即可连接到工作空间:



4、通过 SCP 上传或下载文件

您可以通过 SCP 和工作空间上传或下载文件, SCP 命令格式请参见 scp 使用手册。 工作空间的SSH连接命令格式是: ssh \${TARGET}, 常用的 SCP 命令是:

- 上传: scp file \${TARGET}:/path/to/file
- 下载: scp \${TARGET}:/path/to/file file

使用示例

• 上传文件到 Web IDE 工作空间



假设本地当前目录有个文件: index.html ,希望上传到工作空间的 /workspace/web 目录(将下面命令 中的 \${TARGET} 替换对应工作空间连接地址):

scp index.html \${TARGET}:/workspace/web/index.html

• 下载 Web IDE 工作空间的文件到本地

假设要把工作空间的 /workspace/web/index.js 下载到本地当前目录:

scp \${TARGET}:/workspace/web/index.js ./index.js

快速导入代码仓库

最近更新时间: 2025-05-14 14:54:12

在 Cloud Studio 云端 IDE 中快速导入代码仓库

为方便用户拉取或者发布代码至第三方仓库,Cloud Studio 云端 IDE 现提供快速关联仓库服务商能力。用户可在 新建工作空间导入仓库、个人设置或者编辑器源代码管理中进行账号登录授权关联,即可直接选择所关联服务商的仓 库。

克隆仓库

您可以使用命令面板(ュ๚P)中的 Git:Clone 或源码管理视图中的**克隆仓库**(在没有打开文件夹时)或开始页中 的"克隆 Git 仓库…"从 GitHub 搜索和克隆存储库。下文以 GitHub 为例。



1. 选择 Clone from GitHub 从 GitHub 克隆。





 首次克隆时,需完成 GitHub 绑定。当界面出现相关提示,单击**允许**,系统便会自动跳转至 GitHub 与 CloudStudio 的绑定页面,按指引操作即可完成绑定。完成首次绑定后,后续再使用克隆时会直接显示已绑定 的 GitHub 账号,无需重复此绑定步骤,可直接操作克隆。





3. 当跳转到 GitHub 与 CloudStudio 的账号绑定页面后,您会看到一个名为 Authorize cloudstudiodevops 的按钮。请单击此按钮,以完成两个平台之间的账号绑定操作。



	Authorize Cloud Studio (云端 IDE)	
	We do do Studio (Faill (D2)) by choosestudio Image: A constraint of private states and the	
	Cancel Authorize doublistudio- decops Authorize doublistudio-	8
() © 2005 GBHAB. Inc.	Iteremented or operated by Gifskab O Created B More than 1K Goffskab cores Learn more about Okuth Learn more about Okuth Terms Phage Seate Dos Contact Margar cooles Dos Contact Margar cooles Dos of the phage Seate Dos of the personal information	

4. 请输入与您 GitHub 账号对应的密码,随后单击 Confirm 提交。请确保密码输入准确无误,若密码错误或遗 忘,可通过 GitHub 密码重置流程进行处理 。







6. 当您完成授权登录,页面便会展示您 GitHub 账号下的所有仓库,您可按需选择想要被克隆至 CloudStudio 的 仓库。确定所需仓库后,单击选定,后续即可开启克隆流程 。



7. 选择克隆下来的项目目录。





8. 克隆操作开始后,页面右下角将弹出提示框,实时显示克隆进度状态。该提示框会直观展现克隆进程,助您及时 了解任务执行情况,在此期间请耐心等待直至克隆完成。





- 9. 当系统提示克隆成功后,会弹出窗口询问您打开方式。按需做出选择,便能顺利进入克隆窗口开展后续工作 。
 - 若您倾向于在当前操作环境下直接开启克隆内容,延续连贯操作,那么选择 Open 在当前窗口打开即可。
 - 若您想在独立空间查看克隆成果,避免干扰,可选择 Open in New Window 打开一个新窗口。





推送仓库

1. 您可以使用命令面板(ュ≋P)中的 Git:Push 或源码管理视图中的发布到 GitHub 储存库(当前文件夹为非 Git 项目时)来推送仓库。





2. 然后单击**允许**进行 GitHub 登录。





 若您此前已完成 GitHub 与 CloudStudio 的绑定,在此界面,可直接从账号列表中,便捷选择您需要用于推送 的 GitHub 账号,快速开展后续工作。倘若尚未进行绑定,可以按照下一流程的详细指引逐步操作,便能顺利完 成绑定,实现账号间的顺畅联动。





若您需要绑定其他的 GitHub 账号,只需单击界面中的**登录到其他账户**选项。然后,系统将自动跳转至 GitHub 与 CloudStudio 的绑定页面。在该页面中,您可以按照提示进行授权登录操作,从而完成新账号的绑定流程。 这样,您就能够灵活切换不同的 GitHub 账号,满足多样化的使用需求。





4. 选择您需要推送的 GitHub 仓库空间。





5. 然后,选择您需要推送到 GitHub 仓库的文件,单击确定。





6. 单击确定后,页面右下角将弹出实时进度弹窗,动态展示文件推送至 GitHub 仓库的进度情况,助您直观了解任 务执行状态,可随时掌握推送进展 。





7. 当页面右下角弹出提示"Successfully",意味着文件已成功推送至 GitHub 仓库。此时,你可单击 Open on GitHub,跳转到对应的仓库页面。在仓库页面中,仔细查看文件列表及最新提交记录,确认所推送的文件是 否完整呈现,且内容无误,以此核实推送操作是否圆满达成。





8. 查看仓库页面推送的内容,如下示例:



4 workspace Private	⊙ Unwatch	1 • ¥ Fork 0 • 🛱 Star 0 •	
${\mathfrak P}$ main * ${\mathfrak P}$ 1 Branch ${\mathfrak H}$ 0 Tags ${\mathbb Q}$. Go to file	t Add file * 🗘 Code *	About ®	
cloudstudio_app2 first commit	1983910 now 🕲 1 Commit	No description, website, or topics provided.	
test.py first commit	now	か Activity ☆ O stars	
Ф пеадме		 ⊙ 1 watching ♀ 0 forks 	
		Releases	
		No releases published <u>Create a new release</u>	
Add a READ/WE Add a README with an overview of your project.		Packages	
Add a README		No packages published Publish your first package	
		Languages	
		Python 100.0%	
		Suggested workflows Based on your tech stack	
		Python application Configure Create and test a Python application.	
		Python Package using Configure Anaconda Craste and test a Python package on multiple Python versions sing Anaconda for package management.	
		dj Django Configure Build and Test a Django Project	
		More workflows Dismiss supportions	

使用 Git 进行版本控制

Cloud Studio 云端 IDE 的工作空间支持从代码仓库创建,不过在此之前您需要将工作空间的 SSH Key 添加至对 应代码托管平台的个人公钥列表。

我们以 GitHub 仓库为例,假如我们想创建一个工作空间,代码来源为一个 GitHub 仓库。

1. 选择代码来源

首先进入工作空间创建页面,打开代码仓库选项。



	deepseek r1 v deepseek r1 v ws python 3.11 v ollama v ws fortran v
3	额外资源
	送得资源
4	预装插件 为工作空间默认安装 VSCode 插件。
5	自定义命令 支持 tab 键跳转下一行,支持在不同阶段执行自定义 Shell 命令
	+ 増加 ◆ 启动命令
6	
•	
0	· 存储设置 额外配置选项
	模板环境配置以文件形式(workspace.yami)存储到开发空间中。
8	Кидер
	开启后,可为模板绑定一个默认仓库,在创建时会自动拉取代码仓库。
	● 导入仓库 ● 仓库地址 ● 空
	请输入 Git 仓库地址,HTTPS 只能克隆公开仓库
	○ 请确保已经将 SSH 公钥 ① 加到个人公钥列表,否则会导致创建失败。查看数程
	新建 号出 workspaceyml 取消

2. 添加公钥至目标仓库

2.1. 单击 SSH Key 复制公钥。


	deepseek r1 v deepseek r1 v ws python 3.11 v ollama v ws fortran v	
3	额外资源	
	选择资源 🗸	
4	预装插件 为工作空间默认发装 VSCode 插件。	
5	自定义命令 支持 tab 銀跳转下一行,支持在不同阶段执行自定义 Shell 命令	
	4> 初始化命令	
	小 启动命令 ***/*********************************	
6	环境变量 支持 lab 键跳转下一行	
7	存储设置 额外配置这项	
8	代码仓库	
	● 导入仓库 ● 仓库地址 ● 空	
		Ø
	◎ 请确保已经将 SSH 公钥 Ĵ 加到个人公钥列表,否则会导致创建失败。查看教程	
	新建 导出 workspace.yml 取消	199 199 192 192 193 193 193 193 193 193 193 193 193 193

2.2. 登录 GitHub 账号后,单击右上角头像,选择 Settings 进入设置页面。





E Dashboard		Q. Type [7] to search	ti Ke ne ∉ ×
Improvisions Improvisions	Home Add Capita Add Capita 「 Find Bach script location ① O teno tota LinkedList/ArrayLis ① Jana List conversion guide Feed 「 Introffing repositories - See more 」 byreforant defeer final Defension is a community-driven finanework for deep research, combining language models with totals like web secht, canaling and Python exection, while contributing back to the open-source community. Python ① State	▶ Jam sp. ■ Jam sp. (+1.00) Jam sp. • Hermostor • Adamsot. • Markenhardhar •	A Your profile G Your repositories Your organizations Your organizations Your organizations Your spancas Your spancas Your spancas G Hature preview S Settings G Hatub Support G Hatub Suppor

2.3. 在左侧菜单栏找到并单击 SSH and GPG keys,进入密钥管理界面。单击 New SSH key,即可开始添加 新的 SSH 公钥,完成密钥配置为后续操作提供安全保障。



≡ ○ Settings				Q Type] to search	8 .	+ • 0	n Ø	٩
	K: ne (K: n Vor personal account A Public profile A Account A Appearance A Appearance A cressibility A Notifications Accessibility A Notifications Image: Compare and account Acress Emails O Resource and addrestication Image: Compare and account M Sessions Compare and account on the Sections Organizations Image: Codegaacs Code, planning, and automation Image: Codegaacs Accession Repositories Code plans Security O Code security Image: Code security	ne) SSH keys 2 To is a list of SSH keys associated with you or account. Remove any keys that you do not recognize. Subtractication keys	e to your personal profile New SSR kgr Dedre Dedre Dedre New SSR kgr					0
	 Applications Scheduled reminders Archives Security log Sponsenship log Developer settings 							2 2 2 2

2.4. 在左侧 CloudStudio 界面中,找到并单击 **SSH公钥**,即可自动复制生成的公钥内容。随后切换至右侧 GitHub 页面,在"Key"输入框中粘贴刚刚复制的公钥,确认内容无误后,单击 Add SSH key,即可快速完成 新 SSH 公钥的创建与绑定,实现两个平台间的安全连接。



deepseek r1 × deepseek r1 × ws python 3.11 ×	ollama 🔻 ws fortran 💌 🗏 💭 Settings	Q Type [] to search S + + + O Th 🛆 🔱
3 额外资源	Ke n	(Ge to your personal profile
 送探資源 ▼ 預装插件 カエ作空间就认安装 VSCode 操件。 请输入插件名称 Q 	A Public profile B Account d Appearance # Accessibility Q Notifications	Add new SSH Key Title Cloud Key type Authentication Key *
(5) 自定义命令 支持 tab 键跳转下一行,支持在不同阶段执行自定义 Shell 命令 小 初始化命令 + 増加 小 启动命令 + 増加 小 关机命令 + 対加	Acces Billing and idensing (New V Smalls O Researcd and authentication M Sessions P SSM and GPD keys Chromotopartications	Key Shing A C V V U U
 6 环境变量 支持 tab 键跳转下一行 + 增加 7 存储设置 额外配置选项 模板环境配置以文件形式 (workspace.yam)) 存植到开发空间中。 8 代码仓库 	Code planning, and automation Code planning, and automation Responsibilities Code planning Code planning Code planning Code planning Plackages Plackages P	Add SSH key 3
 开自后,可为模板绑定一个数认仓库,在创建时会自动拉取代码仓库。 ● 导入仓库 ● 合库地址 ● 空 靖徐入 Git 仓库地址,HTTPS 只能克隆公开仓库 ○ 请徐安已经 (F SSH 公明) (19) 个人公钥列表,否则会导致创建关税。查看教程 1 新建 导出 workspace.yml 取消 	Security Code security Integrations Applications Scheduled reminders Actives Security log Security log Security log Security log Security log Security log Security log Security log	

3. 填入仓库地址并创建

3.1. 填入想要克隆的仓库地址(目前仅支持 SSH 协议的地址)。



3 额外实源	
选择资源 🗸 🗸	
4 预装插件 为工作空间就认安装 VSCode 插件。	
5 自定义命令 支持 teb 理跳转下一行,支持在不同阶段执行自定义 Shell 命令	
+ 境加	
ゆ 关机命令	
6 环境变量 支持 tab 建脱柱下一行	
+ 増加	
7 存储设置 额外配置进项	
● 导入仓库 ● 仓库地址 ● 空	
https://github.com/KeiranHome/generative-ai-for-beginners.git	
○ 请确保已经将 SSH 公明 ⑦ 加到个人公钥列表,否则会导致创建失败。查看教程	6
₽ ° main →	
新建 寻出 workspace yml 取消	0.1 + 1/s 0.1 + 1/s
	۵

3.2. 单击**新建**,即可创建一个克隆了该仓库代码的工作空间。

在线预览调试

腾讯云

最近更新时间: 2025-01-16 18:56:52

Cloud Studio 云端 IDE 的工作空间内置预览插件,方便您生成一个预览链接,用于实时调试预览或项目展示。下 面我们以创建一个 React 应用,并实时开发调试为例,来简单介绍一下如何使用 Cloud Studio 进行在线预览。

创建工作空间

登录 腾讯云官网控制台 ,选择任意模板即可基于该模板创建空间。此处以选择 Node.js 环境为例,单击卡片创建 并进入工作空间。



初始化 React 应用

接下来,我们来初始化一个 React 应用。为了方便,我们使用 create-react-app 脚手架。 选择上方菜单栏终端 > 新建终端,打开终端。执行 yarn create react-app app 来初始化一个 React 项 目,它会在当前目录下创建一个 app 文件夹,这就是我们应用的代码。





启动应用

上述过程同时也会安装好所需的依赖,所以我们现在只需要在终端执行 cd app 进入 app 。执行 yarn start 就可以启动应用了。此时终端显示应用启动成功,并运行在 http://localhost:3000 。





使用预览插件

用户可以通过使用预览插件打开正确的浏览器端口做快速预览,即可以在 IDE 页面开启内置的的浏览器预览窗口, 进行编码预览和调试。系统会识别 preview.yml 文件的配置执行命令。通过端口自动扫描识别,并进行域名转 发,提供便捷的预览调试的编码体验。

配置文件模板及参数含义

```
# .vscode/preview.yml
autoOpen: true # 打开工作空间时是否自动开启所有应用的预览
apps:
    - port: 3000 # 应用的端口
    run: yarn start # 应用的启动命令
    root: ./app # 应用的启动目录
    name: my-first-app # 应用名称
    description: 我的第一个 App。 # 应用描述
    autoOpen: true # 打开工作空间时是否自动开启预览(优先级高于根级 autoOpen)
    autoPreview: true # 是否自动打开预览,若无则默认为true
```

功能点一: 通过模板自动打开预览窗口



如果 .vscode/preview.yml 文件中, autoOpen 和 autoPreview 均为 true, 进入工作空间的编辑页面会自

动打开预览窗口。



功能点二:重启运行预览窗口

单击文件中的下拉选项"打开预览文件 preview.yml",即可重新运行 preview.yml 文件窗口。点击右上角的

り,即可重启预览页面。





功能点三:命令行启动预览窗口

快捷键 Command + Shift + P或Ctrl + Shift + P打开命令面板。打开命令输入框,输入命令行preview.start即可重新打开预览窗口。

帮助	preview.yml — RemoteWorking — Cloud Studio	
! preview.yml ×	>preview.start	
.vscode > ! preview.yml	Preview: 打开预览标签	
1 # .vscode/preview.yml	Preview: Open Preview Tab	
2 autoOpen: true # 打开工作空		
3 apps:		
4 – port: 3000 # 应用的端口		
5 run: yarn && yarn sta	t # 应用的启动命令	
6 root: ./ # 应用的启动目录		
7 name: React Quickstart	: # 应用名称	
8 description: React Qu:	.ckstart Project # 应用描述	
9 autoOpen: true # 打开ユ	作空间时是否自动开启预览(优先级高于根级 auto0pen)	
10		

功能点四:端口中进程的快捷操作

对于"端口"中所展示的启动端口进程,将鼠标移入该进程,即可单击右侧显示的 😭 🕀 📭 分别进行"复制地址到剪 贴板"、"打开外部浏览器"、"打开内置预览窗口"的快捷操作。





问题 输出 终端 端口	调试控制台	@ ^ X
V PORTS		
🔵 3000 React Quickstart - 🤅	<u> </u> 推程 PID: 1247	🛱 🖶 🗅
		复制预览链
⊗ 0 ∆ 0		Layout: U.S. 📿



常用中间件安装

最近更新时间: 2025-06-10 15:56:01

前置条件

访问 控制台,打开工作空间的终端页面(以下安装流程皆在此终端中输入)。



MySQL

1. 更新软件包列表。运行此命令后,系统即可获取可用软件包的最新版本信息。

apt update

- 2. 安装 MySQL 数据库服务器和客户端。
 - mysql-server : 是数据库的核心服务组件,负责数据的存储和管理等功能。
 - O mysql-client :则是用于连接和操作 MySQL 数据库的客户端工具。
 - -y : 该选项的作用是自动确认安装过程中的提示,无需人工干预。



apt install mysql-server mysql-client -y

安装成功如下图所示:



3. 等待安装完成后即可启动 MySQL 数据库服务(其在后台运行并监听客户端连接,默认端口:3306。):

service mysql start

启动成功如下图所示:

问题输出调试控制台 性外端 端口 注释			□ ~ □ zsh 🛆 □	🛍 … ^ X
debconf: (No usable dialog-like program is installed, so the dialog based frontend	d cannot be used. at /usr/share/per	rl5/Debconf/FrontEnd/Dialog.p	m line 79.)	
debconf: falling back to frontend: Readline invoke-rc.d: could not determine current runlevel				
invoke-rc.d: policy-rc.d denied execution of stop.				
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.c	cnf) in auto mode			
mysqld will log errors to /var/log/mysql/error.log				
mysqld is running as pid 2004				0
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /usr/	<pre>Lib/systemd/system/mysql.service.</pre>			1
invoke-rc.d: policy-rc.d denied execution of start.				0.2
Setting up libcgi-pm-perl (4.63-1)				+ K/ 0.3 + K/
Setting up libhtml-template-perl (2.97-2)				
Setting up mysql-server (8.0.42-0ubuntu0.24.04.1) Setting up libegi-fast-pepl (1.2 17.1)				
Processing triggers for man-db (2.12.0-4build2)				
Processing triggers for libc-bin (2.39-0ubuntu8.4)				
♦ → /workspace service mysql start		(i) 发现一个新端口: 3306		<u>بن</u> کی کی
su: warning: cannot change directory to /nonexistent: No such file or directory		本语: Cloud Studio Pro	信白 打工力罢预收	打工浏应部
		未尿. Cloud Studio BIO 重有输口	日志	打开冽见器
○ → /workspace				1

① 说明:

Cloud Studio 进程管理使用 init.d 。



4. 启动后即可使用 mysql 命令连接:



Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

() 说明:

- 一般情况下,开发时都会使用密码,因此可以运行 mysql_secure_installation 执行安装初 始化。
- 启动前需要确保 MySQL 服务正在运行,可使用启动命令 systemctl status mysql 来校验。
 - 若 MySQL 正在运行,会显示 active (running)。
 - 若未运行,会显示 inactive (dead) 或类似状态。
- 5. 加固 MySQL 数据库安全性。

5.1 设置密码。

```
▲ 注意:
```

由于当前安装的 MySQL 版本 ≥ 5.7 且未设置 root 用户密码,需先完成登录操作,再执行密码 设置。

- 5.1.1 登录 MySQL: mysql -u root -p
- 5.1.2 设置 root 密码: 登录后,执行以下 SQL 语句设置密码(密码示例: 12345678cs ,建议根据安全 策略自定义):

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '12345678cs';
```



5.1.3 设置成功如下图所示:



5.2 在命令行运行如下脚本:

mysql_secure_installation

运行该脚本后,系统会引导用户进行一系列安全相关操作:

5.2.1 会出现要求输入密码的提示,此时,在光标处输入之前设置的密码(例如: 12345678cs)。如下 图所示:

注意:
 输入过程中密码不会显示在屏幕上,输入完成后按回车键确认,脚本就会继续后续的安全配置流程。

Securing the MySQL server deployment.

Enter password for user root:

- 5.2.2 会评估当前设置密码的强度并给出难度等级(这里是 50):
 - 想更换密码:当脚本提示询问是否更换密码,若觉得当前密码难度不足,输入 y 回车。系统会引导您 设置新密码:





设置后又会告知新密码难度等级,并再次询问是否使用这个新设置的密码。若认可新密码,输入 y 回车确定使用;若不满意,输入 n 回车,就会重复密码输入操作,可再次设置新密码:

Estimated strength of the password: 50 Do you wish to continue with the password provided? (Press y | Y for Yes, any other key for No) : y

不想更换密码:若觉得当前密码难度足够,在提示是否更换密码时,输入 n 回车,脚本便跳过更换密码流程,继续进行后续的数据库安全配置步骤:

```
The 'validate_password' component is installed on the server.
The subsequent steps will run with the existing configuration
of the component.
Using existing password for root.
Estimated strength of the password: 50
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n
```

5.2.3 会提示是否移除匿名用户:

- 选择 n 回车意味着保留 MySQL 中的匿名用户账号。匿名用户可在无需认证的情况下连接数据库,
 存在安全风险。保留匿名用户可能方便一些测试场景,但会降低数据库安全性。
- 选择 y 回车,脚本会自动移除这些匿名用户账号,提升数据库安全防护水平。

这里以选择 n 回车为例:

By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? (Press y Y for Yes, any other key for No) : n

... skipping.

5.2.4 会提示是否禁用 root 用户远程登录:

- 选择 n 并回车,意味着允许 root 用户从本地或远程登录数据库。需注意:
 - root 用户权限极高,若允许远程登录且密码强度不足,可能增加被攻击风险(如暴力破解)。
 - 本地登录(默认)更安全: 若仅需在服务器本地使用 root 用户,建议保持默认的 "不禁 用"(即允许本地登录,禁止远程登录)。
- 选择 y 并回车,脚本会禁用 root 用户的远程登录权限,仅允许通过本地(localhost)登录,这是 更推荐的安全做法。

这里以选择 n 回车为例:



Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network. Disallow root login remotely? (Press y|Y for Yes, any other key for No) : n ... skipping.

5.2.5 会提示是否删除测试数据库:

- \odot 选择 n 并回车,意味着保留 MySQL 自带的 test 数据库。
- 选择 y 并回车,意味着删除测试数据库。
 - () 说明:
 - 测试数据库的作用:默认包含用于测试的示例数据和权限,通常用于开发环境的功能验证。
 - **安全风险:** 若在生产环境中保留 test 数据库,未授权用户可能通过它尝试访问或注入数 据,存在潜在安全隐患。

这里以选择 n 回车为例:

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n ... skipping.

5.2.6 会提示是否"重新加载权限表":

- 输入 y 并回车,意味着让脚本立即刷新 MySQL 的权限配置。
- 输入 n 并回车,意味着让脚本不刷新 MySQL 的权限配置。

() 说明:

- 此操作确保本次安全配置(如密码修改、用户删除等)的权限变更立即生效,无需手动执行
 FLUSH PRIVILEGES 命令。
- 脚本完成所有配置后,通常会最后询问是否重新加载权限表,这是流程的最后一步。

这里以选择 у 回车为例:

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y Success.

All done!



 初始化完成后,可通过以下命令使用密码登录 MySQL(**需将[您的密码] 替换为实际设置的密码,示例为** 123456789cs):

mysql -uroot -p**您的密码**

成功后如下图所示:

腾讯云

```
• → /workspace mysql -uroot -p123456789cs
 mysql: [Warning] Using a password on the command line interface can be insecure.
 Welcome to the MySQL monitor. Commands end with ; or g.
 Your MySQL connection id is 21
 Server version: 8.0.42-Oubuntu0.24.04.1 (Ubuntu)
 Copyright (c) 2000, 2025, Oracle and/or its affiliates.
 Oracle is a registered trademark of Oracle Corporation and/or its
 affiliates. Other names may be trademarks of their respective
 owners.
 Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
 mysql>
 () 说明:
    • 当使用 mysql -u root -p密码 命令登录时, MySQL 会提示警告 Using a password on
      the command line interface can be insecure。这是因为密码会以明文形式显示在命令行历
      史中,存在泄露的风险。

    推荐操作(更安全):

      1. 输入此命令后按回车键: mysql -u root -p
      2.系统会提示 Enter password: ,此时再输入密码,屏幕不会显示输入内容。
```

Redis

1. 更新软件包列表。运行此命令后,系统即可获取可用软件包的最新版本信息。

apt update

2. 安装 MySQL 数据库服务器和客户端。

- O apt install: 这是 Debian 系统用于安装软件包的命令。
- redis-server : 它指的是 Redis 键值存储数据库的服务器端程序。
- -y : 该选项的作用是自动确认安装过程中的提示,无需人工干预。





apt install redis-server -y

安装成功如下图所示:

Unpacking redis-tools (5:7.0.15-1ubuntu0.24.04.1) Selecting previously unselected package redis-server. Preparing to unpack/redis-server_5%3a7.0.15-1ubuntu0.24.04.1_amd64.deb Unpacking redis-server (5:7.0.15-1ubuntu0.24.04.1) Setting up libjemalloc2:amd64 (5.3.0-2build1) Setting up redis-tools (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1)
Selecting previously unselected package redis-server. Preparing to unpack/redis-server_5%3a7.0.15-1ubuntu0.24.04.1_amd64.deb Unpacking redis-server (5:7.0.15-1ubuntu0.24.04.1) Setting up libjemalloc2:amd64 (5.3.0-2build1) Setting up liblzf1:amd64 (3.6-4) Setting up redis-tools (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) (Constant determine current runlevel invoke-rc.d: could not determine current runlevel invoke-rc.d: policy-rc.d denied execution of start. (Constant dsymlink / etc/system/dsystem/redis-server service)
Preparing to unpack/redis-server_5%3a7.0.15-1ubuntu0.24.04.1_amd64.deb Unpacking redis-server (5:7.0.15-1ubuntu0.24.04.1) Setting up lib]zf1:amd64 (3.6-4) Setting up redis-tools (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) invoke-rc.d: could not determine current runlevel invoke-rc.d: policy-rc.d denied execution of start. Created symlink /etc/system/desterver/pedis-server service
Unpacking redis-server (5:7.0.15-1ubuntu0.24.04.1) Setting up libjemalloc2:amd64 (5.3.0-2build1) Setting up libjzf1:amd64 (3.6-4) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) invoke-rc.d: could not determine current runlevel invoke-rc.d: policy-rc.d denied execution of start. Created symlink /etc/system/destserver = /usr/lib/system/redis-server service
Setting up libjemalloc2:amd64 (5.3.0-2build1) Setting up liblzf1:amd64 (3.6-4) Setting up redis-stools (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) invoke-rc.d: could not determine current runlevel invoke-rc.d: policy-rc.d denied execution of start. Created symlink /etc/system/decisservice / usr/lib/system/redis-server service
Setting up liblzf1:amd64 (3.6-4) Setting up redis-tools (5:7.0.15-lubuntu0.24.04.1) Setting up redis-server (5:7.0.15-lubuntu0.24.04.1) invoke-rc.d: could not determine current runlevel invoke-rc.d: policy-rc.d denied execution of start. Created swmlink /etc/system/edis service / usr/lib/system/redis-server service
Setting up redis-tools (5:7.0.15-1ubuntu0.24.04.1) Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) invoke-rc.d: could not determine current runlevel invoke-rc.d: policy-rc.d denied execution of start. Created symlink /etc/system/edis service > /usr/lib/system/redis-server service
Setting up redis-server (5:7.0.15-1ubuntu0.24.04.1) invoke-rc.d: could not determine current runlevel invoke-rc.d: policy-rc.d denied execution of start. Created symlink /etc/system/endis service + /usr/lib/system/redis-server service
invoke-rc.d: could not determine current runlevel invoke-rc.d: policy-rc.d denied execution of start. Created symlink /atc/system/cyctem/cyclis service > /usr/lib/system/cyctem/redis-server service
invoke-rc.d: policy-rc.d denied execution of start. Created symlink /atc/system/Acystem/Padis service -> /usr/lib/system//adis-server service
Created sumlink /atr/system/redis service -> /usr/lih/system/redis-server service
Created symlink /etc/systemd/system/multi-user.target.wants/redis-server.service → /usr/lib/systemd/system/redis-server.service.
Processing triggers for man-db (2.12.0-4build2)
Processing triggers for libc-bin (2.39-Oubuntu8.4)
°→ /workspace

3. 安装成功后即可启动 Redis 服务:

service redis-server start

启动成功如下图所示:

```
• → /workspace service redis-server start
Starting redis-server: redis-server.
```

4. 启动 Redis 服务后,可直接通过 redis-cli 命令连接数据库。当前状态为默认未启用密码验证,连接时无需
 输入密码。操作如下:

redis-cli





5. 在 Redis 客户端 (redis-cli) 中, 输入 exit 或 quit 命令并回车,即可退出当前客户端连接 (这不会停 止 Redis 服务进程)。



- 5.1 重新输入 redis-cli 进入 Redis 服务后,使用 auth 密码 即可认证 Redis 密码。
- 5.2 下图中可以看到进入 redis-cli 后的验证认证状态:
 - 未认证时: 执行 keys * 等命令会提示 (error) NOAUTH Authentication required.
 - **认证后:** 成功返回 "OK"。





使用 GPU 开发

最近更新时间: 2025-03-21 15:41:22

Cloud Studio 现已支持 GPU 连接,背后采用高性能应用服务 HAI 作为算力支持。关于高性能应用服务 HAI 的 介绍,可访问 此地址 了解更多。

开始: 创建 GPU 工作空间



选择模板后,根据弹窗提示选择合适的规格。默认提供两种 GPU 空间规格:免费基础型与 HAI 进阶型。收费情况 可前往 购买指南 了解详情。



	选择空间规格		×
л	运用于推理场京,每个月50000万种党员时长 显存: 16GB + 算力: 8 + TFlops SP CPU: 8 核	 同任能,适用于推理、训练场景,购头后使用 显存: 32GB + 算力: 15 + TFlops SP CPU: 8~10 核 	
	内存: 32GB 查看计费规则 🖸	内存: 32GB	
	新建取消		

△ 注意:

- 免费基础型设置了平台共享的一定额度的预热量,创建后即可获得已开机的 HAI GPU 工作空间。若额 度用完,则需要等待 HAI GPU 启动后方可使用,通常需要2 – 5分钟。
- 由于 HAI GPU 算力数量有限,可能会出现无法创建或开机的情况。可以选择 HAI 进阶型创建,切换
 不同地域查看算力库存情况。

编写代码

大部分模板在空间创建后有预设实例代码,可跟随 Readme 文档使用。空间内的文件结构基本遵循如下规则:

```
root/

├ workspace/ # Cloud Studio 内置示例文件,仅该文件夹变更支持持久化

| └ Readme.md

| └ examples/ # Cloud Studio 示例代码

| └ ...

| └ ...

| └ ...

| coot/ # HAI 其他进程,如ComfyUI、WebUI等

| └ ...

| root/ # 一般模型文件存放在该文件夹

| └ llama3 # 仅示例,不同模版内容不同

| └ ...
```

以 Pytorch 模板为例:





查看 web 预览效果

方法一:通过公网 IP 拼接端口号访问

可以通过指定 IP 及端口号创建服务后,通过公网 IP 拼接端口号查看 web 预览效果。

▲ 注意:

由于目前仅 HAI 进阶型及其他非免费版本可自定义开放端口,以下方法仅限 HAI 进阶型及其他非免费版本可用。



配置端口

前往 高性能工作空间 列表,单击 🔿 查看机器来源,前往 HAI 控制台:

Ξ	Cloud Studio)				9
部 ② ● 令	通用工作空间 高性能工作空间 A 空间模板 应用推荐	23 9	免费使用GPU体验Pytorch,	包含示例代码即开即用		
最近復			高性能工作空间 提供GPU算力,能够即开即用使用AI框架、AI模型、AI应用 Pytorch2.0.0 xyythbn			新建
J۱ 2	u pyter-ussvni 1小时前		び [暂无描述]		创建时间 2024-09-23	免费 GPU基础型 💿
P: 22 Ju	ython-krbfym 2小时前 upyter-fdcspp		Pytorch2.0.0 jxbpph [暂无描述]		创建时间 2024-09-05	免费 GPU基础型 ···
22			Hunyuan 测试 [暂无描述] �濯元-DiT WebUl		创建时间 2024-09-03	HAI GPU进阶型 〇 ···
			混元 Dit ssbjuy ❷ [暂无描述] ◆混元-DIT WebUI		创建时间 2024-08-28	HAI GPU基础型 🔶 ···
			Ilama3		创建时间 2024-07-22	HAI GPU基础型 🗘 ···

在 HAI 控制台的端口配置入口查看是否有公网 IP 的开放:



高性能应用服务	÷	hai-			
 X 算力管理 ○ 应用管理 ○ 资源社区 ℃ BETA 		Ċ	hai- 开机 关机 重启 应用存档 元磺血扩音 制绘		
			<u>気</u> 剤 ID 血売名称 算力処型 配置取構 創催対剤	公网 IP 内网 IP 地域	
		监控	端口配置		φ
		来源	第四协议 推議 备注		

在 腾讯云私有网络控制台,进入安全组页面,放通对应的 IP 与端口号。

入站规则 出站规则					
添加规则 导入规则	优先级排序 全部编辑	多个关键字用竖线 " " 分隔,多	多个关键字用竖线 "I" 分隔,多个过滤标签用回车键分隔 Q ₹		
来源 ① ▼	协议端口 🛈	策略	备注	修改时间	操作
(ICMP	允许		2024-08-23 15:37:14 (UTC+08:00)	编辑 插入 ▼ 删除
	ICMPv6	允许		2024-08-23 15:37:14 (UTC+08:00)	编辑 插入 ▼ 删除
1	ALL	允许		2024-08-23 15:37:14 (UTC+08:00)	编辑 插入 ▼ 删除
1	ALL	允许		2024-08-23 15:37:14 (UTC+08:00)	编辑 插入 ▼ 删除
	ALL	允许		2024-08-23 15:37:14 (UTC+08:00)	编辑 插入 ▼ 删除
(TCP:22	允许		2024-08-23 15:37:14 (UTC+08:00)	编辑 插入 ▼ 删除
:	TCP:22	允许		2024-08-23 15:37:14 (UTC+08:00)	编辑 插入 ▼ 删除
(TCP:80	允许		2024-08-23 15:37:14 (UTC+08:00)	编辑 插入 ▼ 删除



启动服务

以创建 TensorBoard 服务为例,在启动服务时指定 host 与 port:

tensorboard --logdir=runs --host=0.0.0.0 --port=8080

回到 HAI 控制台或使用 curl ifconfig.me 或 curl https://api.ipify.org/ 查询当前机器的公网 IP。 并前往浏览器输入公网 IP与端口号,访问服务。



方法二: 通过 Cloud Studio 提供的端口转发服务访问

如果仅需要短时间预览,可使用下方规格拼接:

```
https://${X_IDE_SPACE_KEY}--${PORT}.${REGION}.cloudstudio.work/ # 举例:
hfrsgm--8434.ap-guangzhou.cloudstudio.work/
```

其中 \${X_IDE_SPACE_KEY} 和 \${REGION} 可以通过浏览器地址查询到, \${PORT} 为启动服务的端口号。



以 DeepSeek 模板为例,端口号如下:



其他常见问题

可前往查看 高性能工作空间常见问题 文档。

🔗 腾讯云

模板使用指南 C++ 使用指南

最近更新时间: 2025-03-12 17:03:53

概述

C++ 是一种静态类型、编译式、通用、大小写敏感、不规则的编程语言,支持过程化编程、面向对象编程和泛型编程。它是 C 语言的一个超集,实际上,任何合法的 C 程序都是合法的 C++ 程序。 C/C++ 语言模板包含了一个简单的 C++ 程序,内置了 g++ 编译器 、 gdb 调试工具以及 .clang-tidy 代码 检查工具。

快速开始

1. 访问 控制台,在空间模板 > 语言模板下单击选择 C/C++ 来创建包含 C++ 运行环境的工作空间。



2. 单击终端 > 新建终端。





3. 在终端中使用如下编译命令,快速运行一个简单的 C++ 程序:

```
g++ main.cpp -o main && ./main
```

O g++ main.cpp −o main

- g++ 是 GNU 编译器集合(GCC)中的 C++ 编译器,用于将 C++ 源代码文件编译成可执行文件。
- O main.cpp 待编译的 C++ 源文件的文件名。
- -o main 用来指定输出文件的名称,不指定名称默认生成 a.out。
- O ./main
 - O ./main 执行编译后的可执行文件。

() 关于构建工具:

本环境默认安装了 GNU 编译器集合(GCC),可以满足基础的 C++ 代码编译需求。但如果您想要使 用其他的构建工具,例如 CMake,由于它并未预装,您需要额外自行安装。

文件结构

clang-quickstart/	
	// gdb 配置文件
	// Cloud Studio 配置文件(运行、预览等)
	// vscode 配置文件(外观,代码编辑等)
🖵 tasks.json	// 在vscode 中用来定义和配置任务的



├── main.cpp	// 主源文件	
├── .clang-tidy	// Clang-Tidy 代码检查工具的配置文件	
	// 项目说明文档	
├── img	// 图片目录	
└──		

C++ 官方文档与资源

- GCC: https://gcc.gnu.org/
- Clang-Tidy: clang-tidy

腾讯云

Node.js 使用指南

最近更新时间: 2025-03-12 17:03:53

Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。Node.js 使用了一个事件驱动、非阻塞式 I/O 的模型,使其轻量又高效。Node.js 的包管理器 npm,是全球最大的开源库生态系统。

快速开始

1. 单击空间模板,选择 NodeJS 模板新建一个 Node.js 预置环境工作空间。



2. 打开后如需查看预置代码的运行效果,可选择 **preview.yml**,将图示的 autoOpen 修改为 true,并单击右上 角的刷新。





3. 再单击端口,并选择"在浏览器打开"或者"在当前标签页打开预览"查看预览。



前端框架应用开发



- 以一个 Vue 的 Demo 为例,操作如下:
- 1. 单击终端 > 新建终端。



2. 在终端,通过 Vue CLI 初始化项目 vue create my-project。

问题 输出 终端 端口 调试控制台 → RemoteWorking vue -V @vue/cli 4.5.13 → RemoteWorking vue create my-project	
Vue CLI v4.5.13 New version available 4.5.13 → 5.0.4 Run yarn global add @vue/cli to update!	
 <pre>? Please pick a preset: (Use arrow keys) > Default ([Vue 2] babel, eslint) Default (Vue 3) ([Vue 3] babel, eslint) Manually select features []</pre>	

3. 使用 yarn run serve 启动项目。



教演管理器 ···· ♥ HelloWorldvue A () package.json A × v REMOTEWORKING my-project > () package.json A × v my-project 1 0 > node_modules 2 > public 3 > vsr c 3	
Yetho TEWORKING my-project > {} package.json > my-project mode_modules public src src mane*: "my-project", "version: "0.1.0", "private": true,	
> my-project 1 億 > node_modules 2 > public 3 > public 4 > vsrc > with	
> node_modules 2 "name": "my-project", > public 3 "version: "0.1.0", > src ▷ 減減	
◇public 3 ^version: *0.10*, ◇src 4 *private*: true, ◇src > 測試	
🚺 🗸 src 🔹 ト 潮流	
vassets 5 "scripts": {	
k logo,png A 6 "serve": "vue-ti-service serve", ™uttut "uttut": two-ti-service serve",	
v components / fulnt: "vue-cli-service build",	
□□ ▼HelloWorldsue A 9 },	
└── ▼ App.vue A 10 "dependencies": {	
Js main.js A 11 "core-js": "3,6.5",	
0.0 ≣.browserslistrc A 12 Vuet ::::0.0"	
esintrojs A 14 "devDependencies": {	
Image: System of the system A 15 "@vue/cli-plugin-babel": "~4.5.13",	
β babel.config.js A 16 "Quuge/Lip-Digin-estimit": "-4.5,13", "Quuge/Lip-Digin-estimit": "-4.5,13",	
() packagejson A / P eruby(L1-Service	
() README.md A 19 "babet-eslint": "^10.1.0",	
& yarn.lock A 20 "estint": "^6.7.2",	
21 "estint-plugin-vue": "^7.0.0"	
问題 輸出 终端 端口 谓试控制台 2:/bin/zsh ◇ + ◇ □ 億	
→ my-project git: (master) × yarn run serve	
yam run v1.22.11 S. Wied-Ti servize serve	
INFO Starting development server	
98% after emitting CopyPlugin	
DONE Compiled successfully in 1598ms 4:26	
App rufining at: - Local: http://localhost:8888/	
It seems you are running Vue CLI inside a container.	
Access the dev server via http://localhost:syour container's external mapped ports/	
Note that the development build is not optimized. To create a production build, run yarn build.	
⑧ 发现一个新编口: 8080	₿×
スカーンエル# 未返 Browser Preview	浏览器
	J.S. (t

4. 单击右下角弹窗打开内置预览/打开浏览器,在端口的可以实时预览前端。





常见问题

如何解决 Vue 项目中的 "Invalid Host header"?

在根目录下创建文件 vue.config.js,然后填入如下内容,重新启动项目即可。









最近更新时间: 2025-03-12 17:03:53

概述

腾讯云

Vue.js 是一个渐进式 JavaScript 框架,专注于构建用户界面和单页应用(SPA)。它由 Evan You 于2014年 发布,核心设计理念是自底向上逐层应用,核心库仅关注视图层,易于与现有项目或第三方库整合。TypeScript 是微软开发的 JavaScript 超集,通过添加静态类型和类等特性,增强代码可维护性和开发效率,尤其适合大型项 目。

使用 TypeScript 开发 Vue 应用能提升代码健壮性,尤其在复杂项目中,类型检查可减少潜在错误。例如,Vue 单文件组件(.vue)可通过 TypeScript 定义 Props 类型和组件逻辑,确保数据流清晰可控。两者结合既能发挥 Vue 的灵活性与高效渲染,又能利用 TypeScript 的类型安全优势,是现代前端开发的常见选择。

预装环境

Node v18.13.0 、 @vue/cli 、 typescript 、 Extentions: Vue-Official 、 pnpm 、 yarn 、 node-gyp 、 git 等。

快速开始

 访问 控制台,在空间模板 > 前端模板下单击选择 Vue-TypeScript 来创建一个预置工作环境空间。该模板项 目代码基于 Vue+TypeScript编写,启动成功后将展示欢迎页。



2. 单击终端 > 新建终端。




3. 在终端中输入如下命令,快速运行一个 Vue+TypeScript 程序:



输出结果:









├── public
│
│ └── index.html // 项目入口 HTML
├── README.md // 项目说明文档
├── src
└── logo.png
— components // 自定义组件
└── HelloWorld.vue
│
└── shims-vue.d.ts // 为 .vue 文件提供类型声明
├── tsconfig.json // TypeScript 配置文件
├── img
∣ └─
└── vue.config.js // Vue CLI 配置文件

Vue.js 官方文档与资源

- Vue.js 官网
- 搭配 TypeScript 使用 Vue
- Node.js 在任何地方运行 JavaScript
- npm官网
- yarn官网

Ollama 模板使用指南

最近更新时间: 2025-03-12 17:03:53

概述

Ollama 是一款支持在本地运行多种开源大型语言模型,具有本地运行、开源模型支持、简化流程、高度定制且跨平 台兼容等特性的软件平台。在 Cloud Studio 中,已内置 Ollama 环境,方便用户快速开展相关操作并探索模型功 能。

环境配置

Ubuntu20.04、Python 3.10、Llama3 8B、CUDA 11.7、cuDNN 8、Ollama 和 JupyterLab。

快速开始

1. 访问 控制台,在空间模板 > AI 模板下单击选择 Ollama 来创建包含 Ollama 运行环境的工作空间。

全部模	反 热门模板 AI模板 前端模板 语言模板			新建模版 💌
热门模	δ.			
>	QwQ 328	DeepSeek-R11.58 ···· 试会社会社会社、或品牌交流会社、或品牌可能均能用的公量上的应用、 ···· ···· Deepseek-R1-Core (DeepSeek R1 1.58) ···· // DeepSeek-R1 ····	DeepSeek-R1 78, 88 ◆ … 然前中局到出资法: 加达法费用: 目标: 对法主意印印目前: … (小 Deeponek-R1 Cast 7, 58 (DeepSeek-R1 78, 68) … … DAMESTAT: … … …	DeepSeek-R1 148
0	DeepSeek-R1 328 🝦	Multi Models 정수는 문문 전, 비교육 전, 부소로, 문제 가루지 문제 가운 보이지, () multi-models (Multi Model) (ALRE::::::::::::::::::::::::::::::::::::	All in One 作意思述 今 programming kanguage demo (政府相由) 法理 Go Python	+TML-CSS-J5
ę	Python	Jupyter Notebook Avarier Notebook 是是完全的意味和研究性能、进行Jupyter Notebook、Pytion、Da. 中 Wulleschungk 保密性 AI 影响的		
AI模板				
0	DeepSeek R1 环境已聚發 DeepSeek R1 水境已聚發 DeepSeek R1		Pytorch2.0.0 (電電) ··· ··	
6	ChatGLM3 68 (高性能) ChatGLM3 68 是—凯由智谱 AI 研发并开强的 ILM 機型,支持工具调用 (Function C ↓ ChatDLM3 68	Liama3 88 Instruct 単電影 Liama3 是一歌曲Meta A开開的LLM模型。Liama-3-80-Instruct是端性能原生开激版… 今 Liama3 88 Instruct	 満元 Dit 通信差」 満元 Dit 温等加品研約端性認知能量中文理解多分領導計数Transformer機型, 提供双 小 満元 Dit 	Liama3.2 - RAG SF135,37 <u>後</u> 性態 通行Lismo-3.2 - IB-Instruct, 内置学习部引,学习如何通过监督学习对大概型进行微 今 Itama32-raq-sft
	CUDA Pytorch Jupyterlab	资言赞型 Ulama3	A应用 语言使型 显元	AI应用 语言模型 Liama3

2. 单击终端 > 新建终端,在终端中输入以下命令直接运行。



S	文件	编辑	选择	查看	转到	运行	终端	帮助		\leftarrow	\rightarrow	
Ð	资源	原管理器				7	;; ;	新建终端 _乐 分终端	^ው` ፞ [፞]			
Q	✓ CL, > . Ξ .	ANG-QL vscode clang-t	лскэт idy	RT			ì	运行任务… 运行生成任务…	ው # B			
2º	G+ I	main.cp	р				ì	运行活动文件				
å							ì	运行所选文本				
₽								显示止在运行的任约 重启正在运行的任约 冬止任务…	务			
හි							궡	记置任务…				
0							E	记置默认生成任务.				

- # 查看Ollama语言模型列表
- ollama list
- # 选择语言模型,启动ollama
- ollama run llama3:latest
- 3. 查看运行结果。
 - Ollama 的运行结果示例图如下:



○ 输入 Ollama 查看更多命令:



(base) root	t@VM-1-177-ubuntu:/workspace# ollama							
Usage:	Usage:							
ollama [†	ollama [flags]							
ollama [o	ollama [command]							
Available (Commands:							
serve	Start ollama							
create	Create a model from a Modelfile							
show	Show information for a model							
run	Run a model							
stop	Stop a running model							
pull	Pull a model from a registry							
push	Push a model to a registry							
list	List models							
ps	List running models							
ср	Copy a model							
rm	Remove a model							
help	Help about any command							
Flags:								
-h,hel	lp help for ollama							
-v,ver	rsion Show version information							
Use "ollama	a [command]help" for more information about							

文件结构

当前目录树默认为 workspace 目录,完整文件结构如下:







Ollama 官方文档与资源

- Ollama 官方 使用文档。
- 查看 Ollama 语言模型。



混元 DiT 模板使用指南

最近更新时间: 2025-03-17 10:14:12

概述

HunyuanDiT 是**腾讯开源的一款文生图大模型**,它支持中英文理解能力,能够根据文本提示生成多分辨率的高质量 图像。

核心特点

- 首个中英双语 DiT 架构: 一个基于 Diffusion transformer 的文本到图像生成模型,此模型具有中英文细粒度 理解能力。
- 中文元素理解: 混元 DiT 提供双语生成能力,对中国元素理解具有优势。
- 长文本理解能力: 混元 DiT 能分析和理解长篇文本中的信息并生成相应艺术作品。
- 细粒度语义理解: 混元 DiT 能捕捉文本中的细微之处,从而生成符合用户需要的图片。
- 多轮对话文生图: 混元 DiT 可以在多轮对话中通过与用户的持续协作,精炼并完善创意构想。

快速开始

1. 访问 控制台,在空间模板 > AI 模板下单击选择混元 Dit 来创建包含混元 DiT 模型及其运行环境的工作空间。

🕛 说明:

该模型建议使用V100或A100,建议您选择 HAI GPU 进阶型,产生的相关费用情况参见 扣费说明。



2. Cloud Studio 已内置 HunyuanDiT 环境,您可以输入命令直接运行。







3. 查看运行结果。

root@VM-0-14-ubuntu: -/FX root@VM-0-14-ubuntu: ~ X +
<pre>(HunyuanDiT) root@M-0-14-ubuntu:-/HunyuanDiT# python app/hydit_app.py 2024-05-16 06:50:47.094 INF0 hydit.inference:iII:172 - Loading CLIP Text Encoder r 2024-05-16 06:50:47.094 INF0 hydit.inference:iII_:175 - Loading CLIP Text Encoder r 2024-05-16 06:50:43.095 INF0 hydit.inference:iII_:175 - Loading CLIP Text Encoder rinished 2024-05-16 06:50:43.096 INF0 hydit.inference:iII_:175 - Loading CLIP Tokenizer r 2024-05-16 06:50:43.096 INF0 hydit.inference:iII_:181 - Loading CLIP Tokenizer rinished 2024-05-16 06:50:43.096 INF0 hydit.inference:iIII_:181 - Loading CLIP Tokenizer rinished 2024-05-16 06:50:43.094 INF0 hydit.inference:iIII_:181 - Loading T5 Text Encoder and T5 Tokenizer'>. This is expected, and simply means that the `legacy` (previous) behavior will be used so nothing changes for you. If you want to use the new behaviour, set `legacy=False'. This should only be set if you understand what it means, and thoroughly read the reason why this was added as explaine d in https://github.com/huggingface/transformers/pull/24565 /root/mincond3/ensyHnuyunaDiT/Livpython3.0f/site-packages/transformers/convert_slow_tokenizer.py:515: UserWarning: The sentencepiece tokenizer that you are converting to a fast tokenizer uses the by te fallback option which is not implemented in the fast tokenizers. In practice this means that the fast version of the tokenizer can produce unknown tokens whereas the sentencepiece version would have converted these unknown tokens into a sequence of byte tokens matching the original piece of text. warnings.warn(You are using a model of type mt5 to instantiate a model of type t5. This is not supported for all configurations of models and can yield errors. 2024-05-16 06:51:06:53: INF0 hydit.inference:init_:184 - Loading V&E finished 2024-05-16 06:51:06:53: INF0 hydit.inference:init_:194 - Loading V&E finished 2024-05-16 06:51:06:073 INF0 hydit.inference:init_:198 - Building HunTuan-DiT model 2024-05-16 06:51:0</pre>
2024-05-16 06:52:05.252 INFO hydit.inference:init_:229 - Loading inference pipeline 2024-05-16 06:52:05.271 INFO hydit.inference:init_:231 - Loading pipeline finished 2024-05-16 06:52:05.272 INFO hydit.inference:init_:235 - Model is ready. 2024-05-16 06:52:05.272 INFO hydit.inference:init_:236 - Model is ready. 2024-05-16 06:52:05.272 INFO hydit.inference:init_:237
Running on public URL: https://9090a0a685e65f7321.gradio.live This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to Spaces (https://huggingface.co/spaces) The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `attention_mask` to obtain reliable results. Setting `pad_token_id' to `eos_token_id': 2 for open-end generation. 2024-05-16 06:54:18.976 INFO hydit.inference:predict:301 - Align to 16: (height, width) = (1024, 1024) 2024-05-16 06:54:18.976 DEBUG hydit.inference:predict:301 - Align to 16: (height, width) = (1024, 1024) 2024-05-16 06:54:18.976 DEBUG hydit.inference:predict:301 - Align to 16: (height, width) = (1024, 1024) 2024-05-16 06:54:18.976 DEBUG hydit.inference:predict:301 - Align to 16: (height, width) = (1024, 1024) 2024-05-16 06:54:18.976 DEBUG hydit.inference:predict:301 - Align to 16: (height, width) = (1024, 1024) action = 1 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0 = 0
11% 11/100 [00:10<01:28, 1.01it/s]



混元-DiT

< 1			
● 方形(1024x1024) (横屏	译(768x1280)	3)	
本描述			
一只可爱的猫			
样步数 100	种子 1	✔ 文本増强	
级设置			
	提交生成		
王多示例			
		七个个时间 计标识 计标识 计	

文件结构

当前目录树默认为 workspace 目录,完整文件结构如下:

root/
├── workspace/ # Cloud Studio 内置示例文件,仅该文件夹变更支持持久化
Readme.md
├── qcloud_hai/ # HAI 其他进程,如 ComfyUI、WebUI 等
│ └─
├── root/ # 一般模型文件存放在该文件夹
│ └── HunYuanDiT
IndexKits
LICENSE.txt
Notice
README.md
pycache
app



asset
│
controlnet
│
diffusers
environment.yml
example_prompts.txt
│
│
launch_hydit_webui.py
lite
lora
requirements.txt
sample_controlnet.py
sample_t2i.py
utils

HunyuanDiT 官方文档与资源

- HunyuanDiT 官方 使用文档。
- 混元-DiT
- HunyuanDiT 在线体验:腾讯混元助手。

QwQ 32B 使用指南

最近更新时间: 2025-03-12 17:03:53

概述

腾讯云

使用 QwQ 32B 模板,我们已为您预装并启动了以下服务,工作空间创建后等待加载十几秒即可查看效果。

- Ollama 服务: 支持通过 API 调用多个模型。
- AnythingLLM 前端服务:提供交互式聊天界面,支持知识库上传并基于知识库进行问答。
- Open WebUI 前端服务:提供交互式聊天界面,支持上传文件作为上下文进行问答。
- 预装模型: QwQ 32B 及其他 Ollama 官网上的热门模型,包括多个通用模型及嵌入模型,并且我们会持续更新。

● → multi-models ollama list	:		
NAME	ID	SIZE	MODIFIED
qwq:32b	38ee5094e51e	19 GB	6 hours ago
qwq:latest	38ee5094e51e	19 GB	6 hours ago
bge-m3:latest	790764642607	1.2 GB	6 days ago
gemma:latest	a72c7f4d0a15	5.0 GB	6 days ago
mistral:latest	f974a74358d6	4.1 GB	6 days ago
<pre>mxbai_embed_large:latest</pre>	468836162de7	669 MB	6 days ago
nomic-embed-text:latest	0a109f422b47	274 MB	6 days ago
deepseek-r1:32b	38056bbcbb2d	19 GB	6 days ago
falcon3:latest	472ea1c89f64	4.6 GB	6 days ago
llama3.2:3b	a80c4f17acd5	2.0 GB	6 days ago
llama3.2:1b	baf6a787fdff	1.3 GB	6 days ago
deepseek-r1:7b	0a8c26691023	4.7 GB	7 days ago
qwen2.5:1.5b	65ec06548149	986 MB	7 days ago
○ → multi-models			

使用教程

访问 控制台,在空间模板 > 语言模板/热门模板下单击选择 QwQ 32B 来创建包含 QwQ 32B 模型的工作空间。

全部	板 热门楔板 AI 楔板 前端楔板 语言模板			新建地址
热门相	ŧ۶			
4)	QwQ 328	ـــــــــــــــــــــــــــــــــ	DeepSeek-R178, 88 €	DeepSeek-R114目。
	CeepSeek-R1 328 (م)	Multi Models	All in One	
¢	Python	Jupyter Notebook Duroter Notebook SEREADENARTER: BEJageter Notebook. Pytton, Da. of WilcochingA (ERE: A EXAMP)		

您可以通过 AnythingLLM 及 Open WebUI 管理知识库并对话,通过命令行调用模型、使用 Ollama 原生命令 对话等方式来使用 QwQ 32B。



一. 通过 AnythingLLM 管理知识库并对话

我们提供了一个开箱即用的聊天界面,您可以直接通过浏览器与模型交互。等待页面加载完成即可查看到下图中的预 览界面。单击红框区域,即可在新页面打开聊天页面。或单击 New Workspace 创建工作区进行对话。



如页面加载完成后没有看到预览界面,可依次单击端口 > 4001 > 🖶 🛛 来查看预览界面。



操作步骤

△ 注意:

该浏览地址仅在当前工作空间内有效,如果关闭工作空间,则无法访问。所以使用中请保持打开当前编辑器 标签页,不要关闭。

1. 设置模型



单击页面左下方的 🖌 打开设置。

\leftrightarrow \rightarrow C \sim brgevm-vjwnog-4001.app.cloudstudio.work	☆ 💦 🏠 🗌 🍓 有新版 Chrome 可用 🗄
CJ Anything LLM 口 +新工作区	 ● 欢迎使用 Cloud Studio 云端IDE × AnythingLLM ■ 云端编码: 全功能在线开发环境 ● 智能对话: 随时与AI助手交互 ● 如识管理: 构建专属知识库系统 + 创建您的第一个工作区
۹ ون	哇,这听起来很棒,让我马上试试! 注意:如果使用的是通用工作空间,请不要关闭编辑器标签页,否则当前页面会因服务断连而失效。
3 II 65 F	

2. 选择模型

2.1 在人工智能提供商 > LLM 首选项 > Ollama Model,选择 Ollama 模型。

\equiv anythingllm-frontend(4001) $ imes$					8 🗉 -
\square \leftarrow \rightarrow \square 4001 \checkmark https:/	/hiretn-grjgji-4001.app.cloudstudio.work				0 🖬 🗋
CO Anything LLM 设置 ③ 人工智能提供商 ~ LLM首选项 向量数源年	Your loaded models bge-m3:latest gemma:latest mistral:latest mxbai-embed-large:latest nomic-embed-text:latest	溶和设置。重要约是,这些密钥是最新的和正确的,	否则AnythingLLM将无法正常运行。	Save changed	
Embedder 首逸功 文本分割 语音和讲话 转录模型 ④ 管理员 > (④ 管理员 > (④ 代理技能 ⊕ Community Hub >	deepsek-r1:32b falcon3:latest Ilama3:2:3b deepseek-r1:7b qwen2:5:1.5b Hide advanced settings	machine. Max Tokens 4096 Maximum number of tokens for context and response.	×		
⊘ 外观	Ollama Base URL	Ollama Keep Alive	Performance Mode ①		
≙ IA >	http://127.0.0.1:8434	5 minutes V	Base (Default)		
 ① 用户与安全 联系支持 総払与政選 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	Enter the URL where Ollama is running.	Choose how long Ollama should keep your model in memory before unloading. Learn more 4	Choose the performance mode for the Ollama model.		

2.2 在人工智能提供商 > Embedder 首选项 > Ollama Embedding Model,选择 Ollama 嵌入模型。



\equiv anythingllm-frontend(4001) $ imes$		0 [
\square \leftarrow \rightarrow \square 4001 \checkmark https://h	retn-grjgji-4001.app.cloudstudio.work	Q 0
DJ Anything LLM 设置 ③ 人工智能提供商 ~	嵌入首选项 当使用本身不支持能入目통的LLM时,低可应需要整外指注用于形入文本的凭思。 嵌入是将文本转换为矢量的过程,需要这些笑慌才能将忽的文件和提示转换为 ArythingLLM 可以用来处理的格式。	
LLM 首选项 向量数据库	嵌入引擎提供商	保存更改
Embedder首选项	Ollama Rin embedding models locally on your own machine	
文本分割 语音和讲述 特決模型 ④ 管理员 > ④ 代理技能 ④ CommunityHub > ② 外現 合 工具 > ③ 用户与安全 平正素 考 降私与数据	Your loaded models Max Embedding Chunk Length gemmä:latest 8192 gemmä:latest Maximum length of text chunks for embedding. mxbai-embed-large:latest momic-embed-text:latest deepseek-rt:32b falcon3:latest llama3.2:3b llama3.2:1b deepseek-rt:72b qwen2.5:1.5b	

3. 开始对话

输入问题或指令,发送后即可获取模型回复。

Webser	
weicome to your new workspace.	
to get started either <u>upload a document</u> or <i>send a chat.</i>	
Send a message	*
	Welcome to your new workspace. To get started either <u>upload a document</u> tor <i>send a chat</i> .

4. 上传知识库

单击上传,选择本地文件,即可上传知识库。



CA Anything LLM	
Documents Data Con	nectors X
My Documents Q Search for document + New Folder	Cloud Studio
Name	Name
No Documents ↔	No Documents
\sim	
Click to upload or drag and drop	
supports text files, csv's, spreadsheets, audio files, and more!	
https://example.com	
These files will be uploaded to the document processor running on this AnythingLLM instance. These files are not sent or shared with a third party.	
# # # #	

二. 其他本地组件配置知识库进行对话(Cherry Studio)

具体操作,请参见 使用 Cherry Studio 搭建个人知识库。

常见问题

1. 如何修改系统语言?

单击页面左下方的 🥙 打开设置,选择**外观 > Display Language**,即可修改系统语言。



J Anything LLM		
+ New Workspace		
👪 Cloud Studio 🗘 🛞		
default		
+ New Thread		
	Welcome to your new workspace.	
	To get started either upload a document or <i>send a chat.</i>	
•	\$end a message	>
	Ú Z @ T	

C/J Anything LLM		外观
设置		自定义平台的外观设置。
🐼 人工智能提供商	>	Theme
@ 管理员	>	Select your preferred theme for the application.
同 代理技能		Light 🗸
Community Hub	>	Diepley Lenguege
		Select the preferred language to render AnythingLLM's UI in, when applicable.
	>	English 🗸
⑥ 用户与安全		English Chinese crollbar
联系支持		Chinese (Taiwan)
隐私与数据		Spanish
		French
		Korean 不天机器人成为您的。
		Russian
		Italian
		Portuguese
		Hebrew B00×200
		Dutch
		Vietnamese
		Persian

2. 如何在其他设备打开?

腾讯云

复制链接地址粘贴到其他设备打开;或将鼠标悬停在界面中如图的红框处并扫码,也可在移动端打开。



三. 通过命令行调用和切换其他模型

您可以使用 curl 直接与 Ollama 服务交互,适用于自动化任务或脚本调用。

```
基础请求示例
```

调用 11ama3.2:1.5B



流式输出(实时逐句返回)

将 stream 参数设为 true:



curl -X	<pre>POST http://localhost:8434/api/generate \</pre>
-H	"Content-Type: application/json" \
-d	
	"model": "ollama3.2:1b",
	"prompt": " 详细说明如何训练一个神经网络 ",
	"stream": true

四. Ollama 原生命令进阶使用

除了 HTTP API, 您还可以直接通过 ollama 命令行工具与模型交互。

1. 查看已安装模型

输入命令:

ollama list

输出示例:

NAME	ID	SIZE	MODIFIED
qwq:32b	38ee5094e51e	19 GB	6 hours ago
qwq:latest	38ee5094e51e	19 GB	6 hours ago
bge-m3:latest		1.2 GB	6 days ago
gemma:latest	a72c7f4d0a15	5.0 GB	6 days ago
mistral:latest	£974a74358d6	4.1 GB	6 days ago
mxbai-embed-large:latest	468836162de7	669 MB	6 days ago
nomic-embed-text:latest	0a109f422b47	274 MB	6 days ago
deepseek-r1:32b	38056bbcbb2d	19 GB	6 days ago
falcon3:latest	472ea1c89f64	4.6 GB	6 days ago
llama3.2:3b	a80c4f17acd5	2.0 GB	6 days ago
llama3.2:1b	baf6a787fdff	1.3 GB	6 days ago
deepseek-r1:7b	0a8c26691023	4.7 GB	7 days ago
qwen2.5:1.5b	65ec06548149	986 MB	7 days ago

2. 启动交互式对话

输入命令:

ollama run qwq:32b

输入 /bye 或者点击 Ctrl + D 退出对话。



示例对话流程:

>>> 推荐一个适合初学者的机器学习项目 QwQ: 一个手写数字识别系统是一个不错的入门项目,可以使用MNIST数据集和Python的 scikit-learn库快速实现...
>> 如何评估模型性能?
QwQ: 常用方法包括划分训练集/测试集、交叉验证、计算准确率/精确率/召回率等指标...
>>> /bye

3. 模型管理

命令	说明
ollama pull qwq:32b	重新拉取模型(更新时使用)。
ollama rm qwq:32b	删除模型。

4. 切换模型

如需切换到其他模型,可在退出当前进程后再次输入以下代码即可:

ollama run qwq:32b

▲ 注意:

当前机器为 CPU,7B模型推理速度较慢,建议使用1.5B模型。更大参数模型建议使用 HAI 进阶型。

常见问题

如何查看更多预览使用?

如需了解更多使用内置 Web 预览的介绍,可参见文档 在线预览调试 。

端口占用,如何释放?

如果需要释放端口,关闭某个进程,可以通过下述命令查看端口占用情况:

lsof -i -P -n

再使用下述命令关闭进程:



kill -9 **<进程**ID>

提示空间不足怎么处理?

如果右下角提示空间不足,可单击右下角的资源规格(如"2核4G"),选择"旗舰版",即可升级空间。



免费版工作空间时长用完如何继续使用?

免费版主要用于帮助用户体验产品功能,暂不支持续费。可以等待下个月再次发放时长后继续使用。或加入 CloudStudio自媒体特权计划领取免费GPU时长。

相关文档与资源

- QwQ-32B 官方介绍文档: QwQ-32B: 领略强化学习之力
- Cloud Studio 更多操作指南: Cloud Studio (云端 IDE) 操作指南