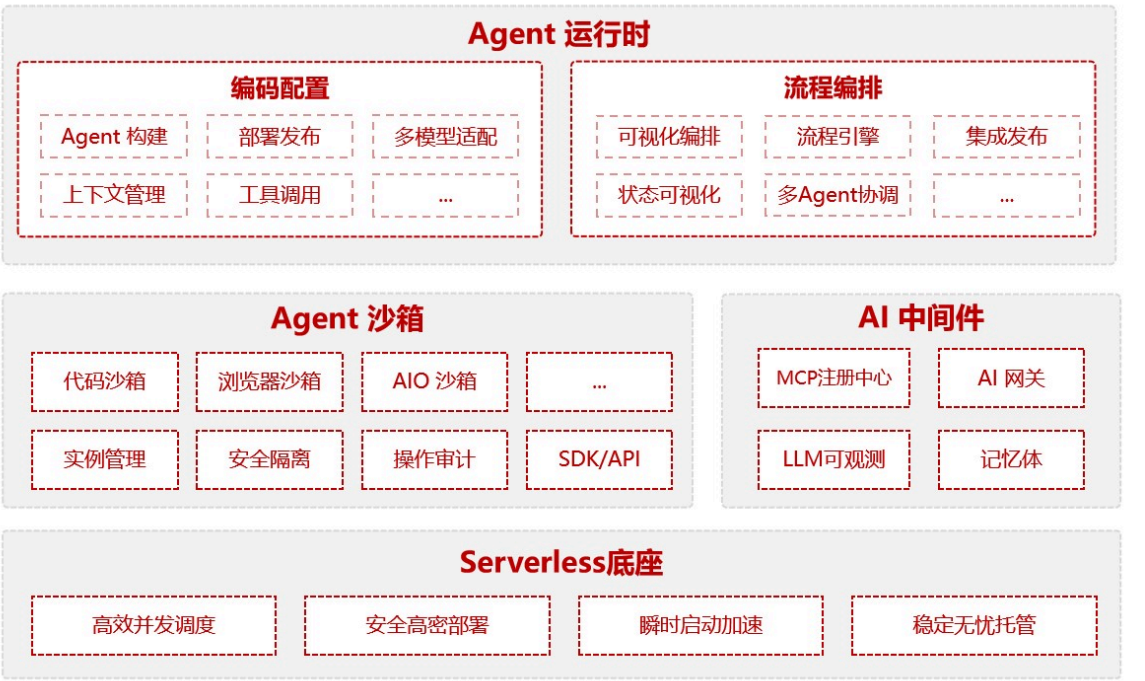
# 智能体引擎（Agent Engine）

# 目录

# 什么是智能体引擎

天翼云智能体引擎（Agent Engine）是一款面向企业级Agent的基础设施服务，为开发者和企业提供安全、弹性、可扩展的云端沙箱运行环境，助力 AI Agent 在多场景下稳定运行与规模化部署。

## 产品架构




## 产品功能

| 功能模块 | 说明 |
| --- | --- |
| Agent运行时 | 负责管理智能体运行、会话和资源调度等核心能力的运行引擎/框架 |
| Agent沙箱 | 提供安全隔离的云端沙箱运行环境，支持多种类型Agent执行和调用 |
| Agent工具库 | 集中管理Agent执行需要的工具、模型及存量系统MCP服务，提升智能体的系统交互能力，支持快速集成与便捷调用 |
| 认证与权限 | 统一管理凭证机权限控制，保障调用链交互过程的安全性与合规性 |

# 产品介绍

## 产品优势

天翼云智能体引擎作为Agent Infra领域产品，提供Agent托管运行、安全隔离、运维监控的全生命周期管理能力，确保Agent从测试体验迈向大规模生产可用。

### 通用兼容

支持多语言/多框架的各类型智能体托管，一键部署、轻松运行

#### 强安全隔离

自研虚机级安全容器沙箱，智能体运行相互隔离权限可控，全面保障系统安全性

#### 高弹性高可用

基于高性能Serverless底座统一调度资源，提供稳定、高效、安全的算力保障

#### 生态拓展

内置工具库和服务，支持集成云服务API，全面拓展业务边界

## 应用场景

### 代码测试与安全分析

为智能体运行提供云端安全隔离的Agent沙箱，主要用于处理不可信或不确定的代码。如果代码在沙箱中崩溃或试图执行危险操作（如删除文件），可以立即阻断，避免污染开发环境。

- 测试AI生成的代码：在沙箱中运行AI生成的代码，进行单元测试或功能验证；大模型（如Cursor、Copilot）生成的代码可能存在语法错误、逻辑漏洞或安全风险（如SQL注入、命令注入）
- 恶意软件分析与逆向工程：将可疑文件放入沙箱中运行，观察其行为：是否修改注册表、是否连接暗网IP、是否加密文件；沙箱提供了“观察窗”，让分析师在不冒风险的情况下了解恶意软件的行为

#### 智能体执行测试

目前最热门的应用场景，让Agent在智能体引擎的安全沙箱中运行，调用外部工具（如浏览器、计算器、终端）来完成任务；包含如下场景：

- UI自动化测试：通过浏览器沙箱运行自动化测试智能体，避免Agent的危险操作或恶意万战访问，无法危及用户的个人电脑或企业内网
- 终端命令执行：通过沙箱运行智能体执行终端命令，即使Agent的指令被恶意prompt注入（例如，用户诱导AI执行rm -rf /），由于沙箱限制了文件系统权限（可能只允许操作Docker相关目录，或挂载了只读根文件系统），也能避免宿主机系统崩溃
- 多Agent协作：通过沙箱运行多个智能体，模拟协作任务或攻防演练；避免影响真实业务

#### 安全合规与数据保护

在医疗、金融等相关领域，使用Agent处理敏感数据时，Agent沙箱可以作为一种数据不落地或数据隔离的手段。

- 私有数据处理：企业使用Agent处理包含客户个人信息（PII，即个人身份信息）或商业机密的Excel表格，Agent启动一个临时的、内存级的沙箱环境。数据在沙箱中被解密和处理，处理完成后，沙箱被销毁。这样可以确保临时文件不会残留在物理磁盘上，降低数据泄露风险。

- 联邦学习与隐私计算：多个团队需要联合训练AI模型，但不能直接共享私密数据；涉密数据在本地沙箱中进行预处理和训练，只将脱敏后的梯度或参数上传到中央服务器，原始数据始终锁在沙箱的安全边界内。

**自动化测试与持续集成/持续部署**

构建智能化软件开发和运维CICD系统，在相关场景可以使用智能体引擎平台的安全沙箱辅助测试：

- 依赖项安全扫描：CI/CD拉取新的开源库后，可在沙箱执行安全扫描；在一个临时沙箱中安装该依赖，并监控其安装后的脚本行为（例如，是否有恶意软件在安装时执行挖矿程序或尝试读取/etc/passwd）。
- 兼容性测试：在沙箱里快速拉起多个不同版本的浏览器环境，并行运行测试脚本，确保页面渲染正常且无崩溃

# 基本概念

## Agent 沙箱

为企业级Agent应用提供强安全、高弹性、可扩展的云端沙箱运行环境，沙箱与宿主机和其他沙箱相互隔离，可限制沙箱实例资源用量，用完即销毁不留痕迹。

### 沙箱模板

预先配置的Agent沙箱，包含镜像、规格和其他配置，基于沙箱模板可以快速启动沙箱实例；智能体运行在沙箱实例中。

**沙箱实例**

基于沙箱模板启动的“运行中的沙箱”实例，每个沙箱实例都是相互隔离、可控、可变的安全容器。

**VNC 调试**

Virtual Network Computing 图形化远程桌面协议，在Agent沙箱场景中用于访问沙箱的图形界面；例如浏览器沙箱

API Key

API Key（应用程序编程接口密钥）是一种身份认证凭证，用于识别和验证调用API的客户端身份。它是一个唯一的字符串，类似于"密码"，但专门用于程序之间的通信。

# 功能特性

## Agent沙箱

| 模块 | 功能 | 描述 |
| --- | --- | --- |
| 沙箱管理 | 沙箱模板增删改查 | 提供沙箱模板的创建、更新、查看功能，基于沙箱模板快速拉起沙箱实例 |
| | 沙箱模板配置 | 管理沙箱模板的配置信息，如规格、环境变量、网络、存储、日志等 |
| 沙箱实例管理 | 沙箱实例的创建与管理 | 管理沙箱实例，包括沙箱实例数据的查看、启动、终止 |
| | 沙箱实例调试 | 提供沙箱实例调试运行能力，包括文件系统、VNC调试、CDP调试 |

# 产品介绍

## 配置管理

| 模块 | 功能 | 描述 |
| --- | --- | --- |
| 凭证管理 | 身份凭证数据管理 | 提供API Key的创建、查询、删除等功能，用于访问智能体应用、访问沙箱实例等资源使用 |
| 域名管理 | 自定义域名管理 | 提供自定义域名的创建、校验，用于访问智能体应用、访问沙箱实例等资源使用 |

# 快速创建Agent沙箱

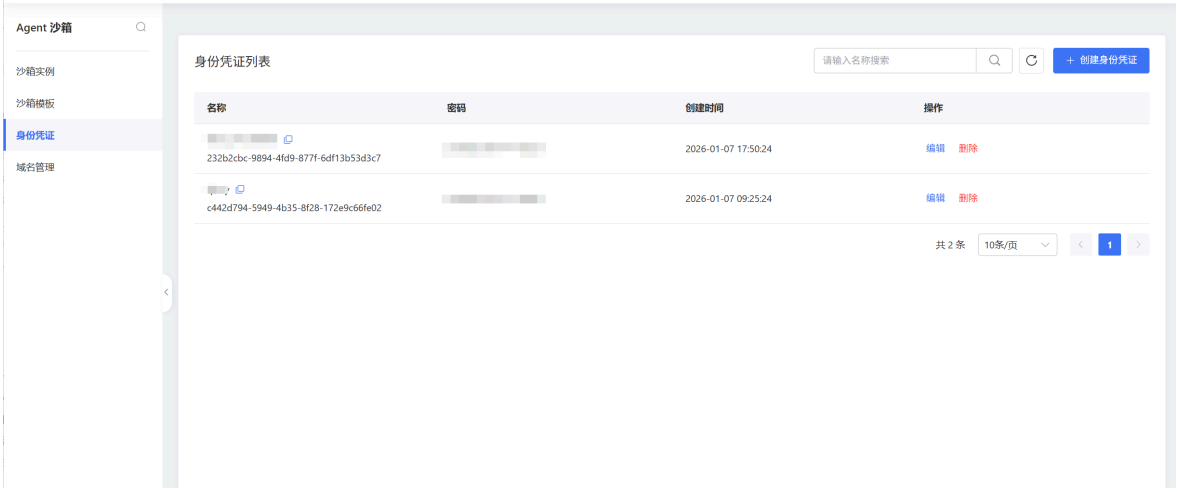
## 概述

本章节将通过sdk快速创建一个沙箱，并在沙箱隔离环境里面运行一段python代码

### 控制台操作

## 创建API Key

登录控制台，在身份凭证管理菜单，创建API Key




## 创建自定义域名

登录控制台，进入“域名管理”菜单，按照提示完成自定义域名创建

# 快速入门

**客户端操作**

## 安装SDK

前置条件：已安装python 3.9+

安装依赖：

```
pip install e2b-code-interpreter==2.3.0
pip install e2b==2.6.0
pip install python-dotenv
```

## 配置环境变量

在您的项目文件夹新建 .env 文件（如果之前不存在的话），并配置 API 密钥和自定义域名，参考如下替换为您的API Key和自定义域名

```
E2B_DOMAIN=your.domain
E2B_API_KEY=your.api.key
```

## 编写代码创建Agent沙箱

创建一个main.py源文件，代码如下：

```
from dotenv import load_dotenv
from e2b_code_interpreter import Sandbox

# The .env file should be located in the project root directory
# dotenv will automatically look for .env in the current working directory
load_dotenv()

sbx = Sandbox.create(template="code-interpreter")

# 代码执行
execution = sbx.run_code("print('hello world')")
print(execution.logs)

# 不再使用时，关闭沙箱
sbx.kill()
```

## 运行代码

```
python main.py
```

输出内容如下：

```
Logs(stdout: ['hello world\n'], stderr: [])
```

# 沙箱管理

## 沙箱模板管理

### 概述

本章节主要介绍沙箱模板管理相关功能。

#### 沙箱模板管理

### 查看沙箱模板

Agent沙箱提供三种模板，即：基础模板base、代码解释器模板code-interpreter和浏览器模板browser-chromium。

#### 操作步骤

1. 登录Agent沙箱控制台。
2. 左侧菜单选择沙箱模板，查看沙箱模板。

## 沙箱实例管理

### 概述

本章节主要介绍沙箱实例管理相关功能。

#### 沙箱实例管理

Agent沙箱支持创建基础模板base实例、代码解释器模板code-interpreter实例和浏览器模板browser-chromium实例。

### 查看沙箱实例

#### 操作步骤

1. 登录Agent沙箱控制台。
2. 左侧菜单选择沙箱实例，查看沙箱实例。

### 创建沙箱实例

#### 操作步骤

1. 登录Agent沙箱控制台。
2. 左侧菜单选择沙箱实例，点击创建沙箱。
3. 完成创建沙箱实例所需参数选择和填写后，点击确定。

| 参数名称 | 参数说明 |
| --- | --- |
| 模板 | Agent沙箱支持base、code-interpreter和browser-chromium三种类型沙箱模板 |
| 超时时间 | 沙箱存活时长，沙箱创建后超过该时长，会被自动销毁 |

| 参数名称 | 参数说明 |
| --- | --- |
| 能否访问公网 | 默认所有沙箱都能访问公网 |
| 能否开启安全访问沙箱 | 开启后，访问沙箱里面需要鉴权 |
| 环境变量 | 沙箱中会增加配置的环境变量 |

### 删除沙箱实例

#### 操作步骤

1. 登录Agent沙箱控制台。
2. 左侧菜单选择沙箱实例，找到需要删除的沙箱实例，在操作列点击删除，确认无误后，点击删除。 

## 基础沙箱实例

### 概述

本章节主要介绍使用基础模板base类型创建的沙箱。在该类型沙箱里中可以管理文件系统和执行shell命令功能。

#### 前置操作

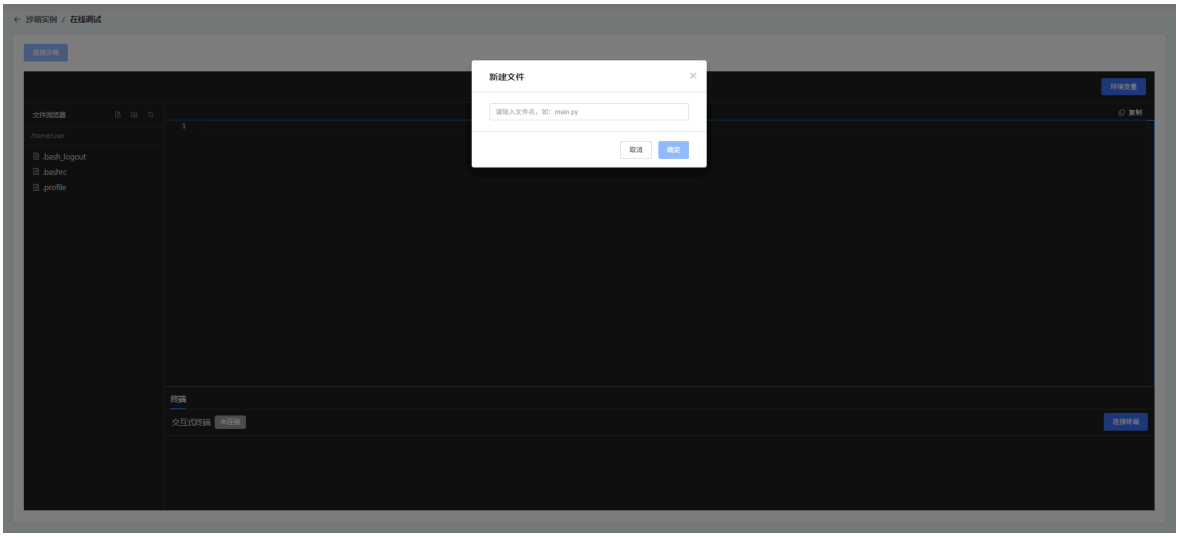
1. 登录Agent沙箱控制台。
2. 左侧菜单选择沙箱实例，创建base类型沙箱实例。
3. 点击在线调试，进入base类型沙箱实例。 

#### 文件系统管理

### 创建文件

#### 操作步骤

1. 页面左侧文件浏览器，点击创建文件。

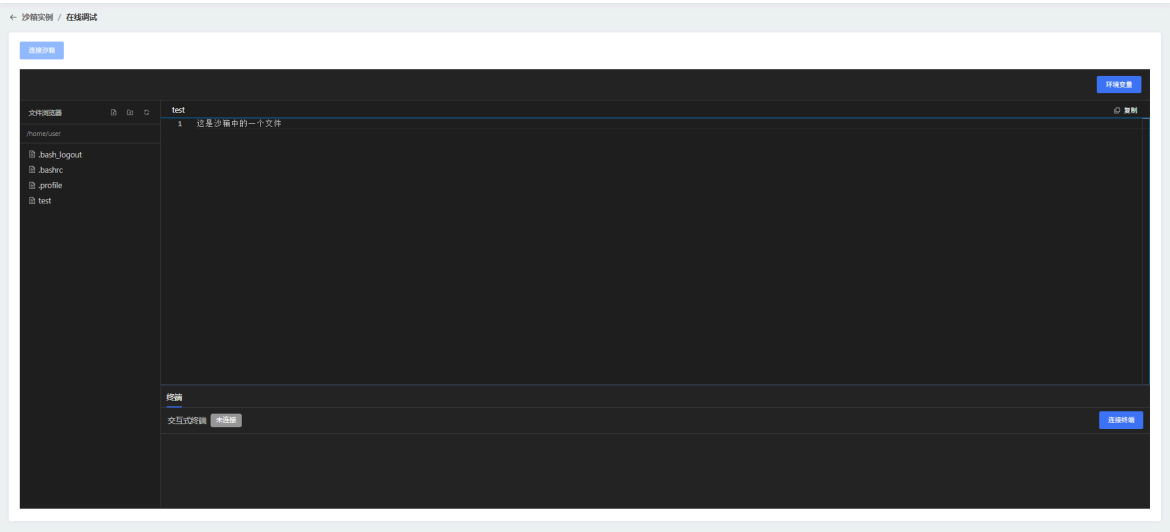



2. 输入文件名，点击创建。

## 编辑文件

### 操作步骤

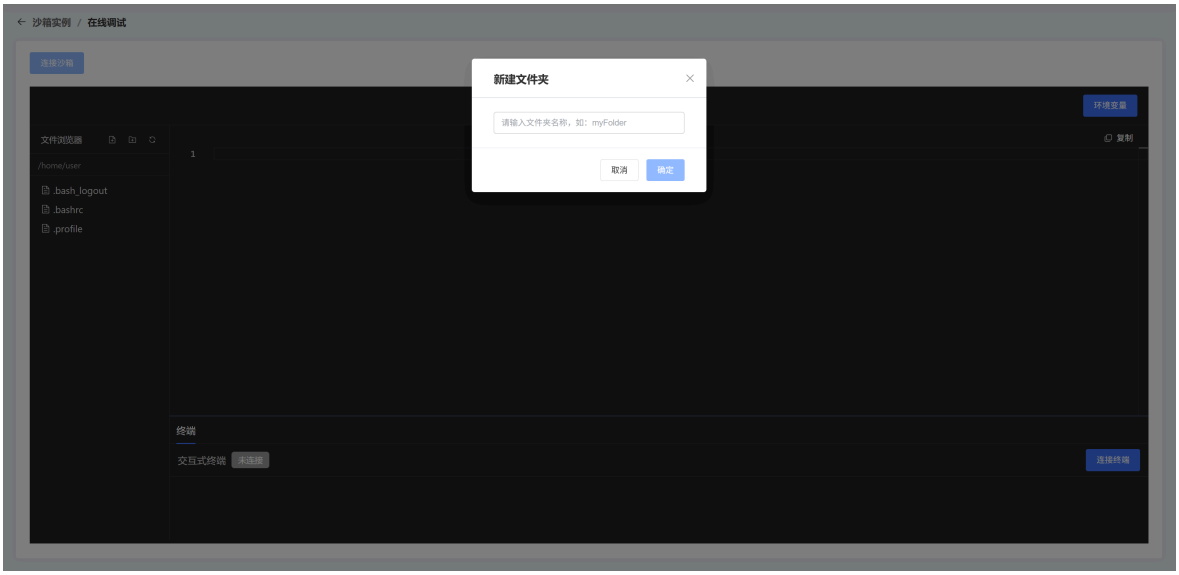
1. 页面左侧文件浏览器，选择文件。
2. 在文件内容区域编辑内容。




## 创建文件夹

### 操作步骤

1. 页面左侧文件浏览器，点击创建文件夹。
2. 输入文件夹名，点击创建。



### 终端执行shell命令

### 操作步骤

1. 点击连接终端。
2. 输入shell命令，查看执行结果，使用完毕后，点击断开终端 。 

## 代码解释器沙箱实例

### 概述

本章节主要介绍使用代码解释器code-interpreter模板创建的沙箱。代码解释器沙箱实例提供了一个安全且强大的代码代码执行沙箱环境，在该类型沙箱时里中可以执行以Python、JavaScript等5种语言代码、管理文件系统、执行shell命令等功能。 代码解释器类型沙箱在基础沙箱的基础上，增加了代码运行的能力，这里为了避免赘述，只展示代码运行的功能。

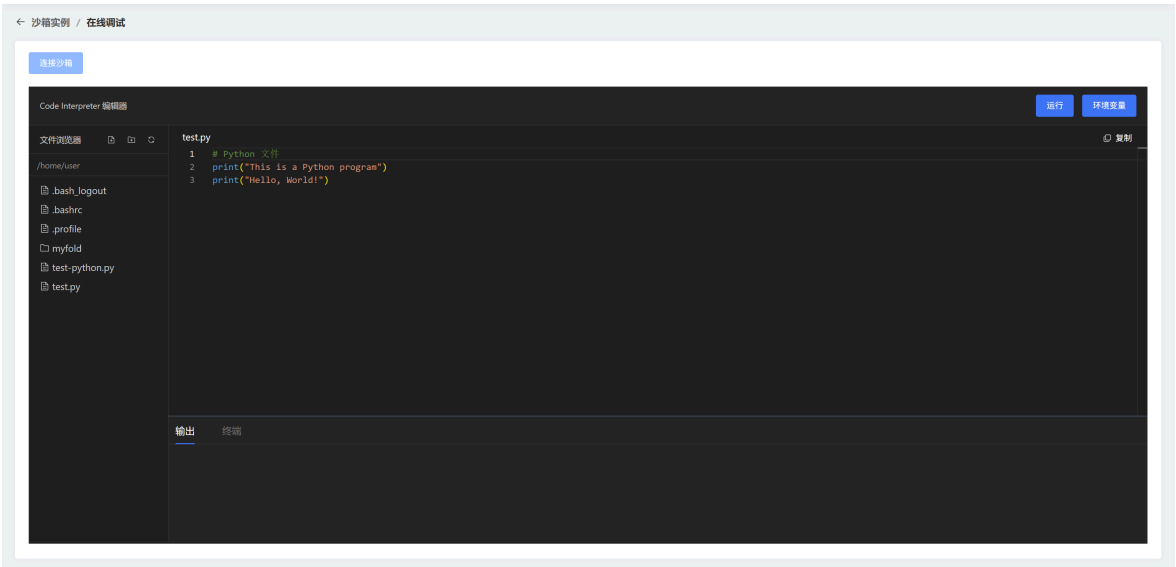
### 前置操作

1. 登录Agent沙箱控制台。
2. 左侧菜单选择沙箱实例，创建code-interpreter类型沙箱实例。
3. 点击在线调试，进入code-interpreter类型沙箱实例。 

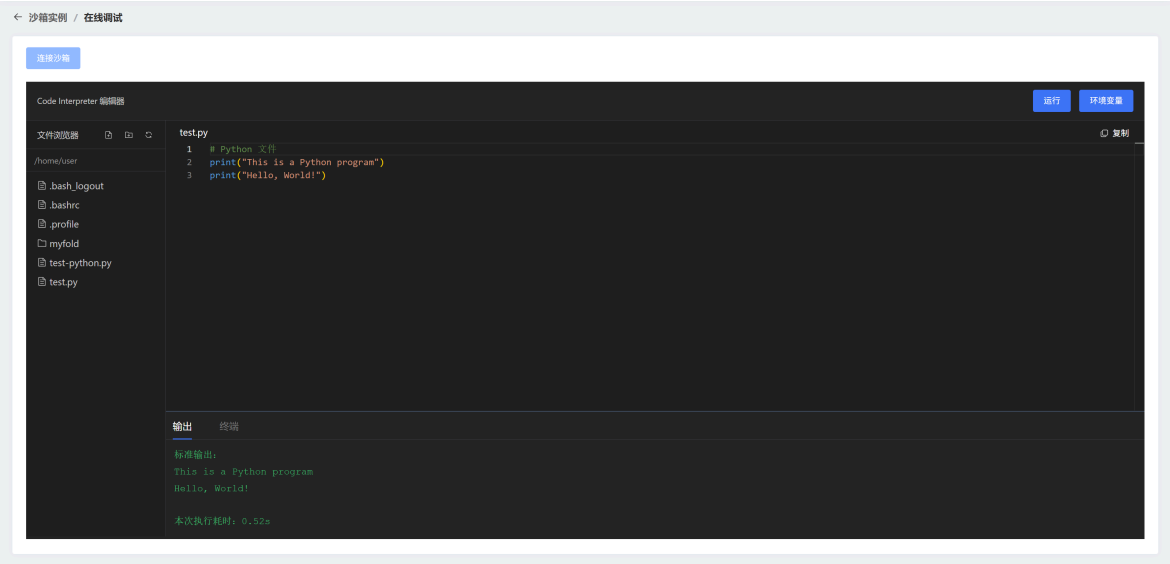
### 执行代码

### 操作步骤

1. 选择需要执行的代码文件，点击运行。



2. 查看代码运行结果。




## 浏览器沙箱实例

### 功能说明

浏览器实例用于在 Agent 沙箱中提供可远程访问的浏览器运行环境。 您可以通过可视化方式查看浏览器运行画面，也可以通过自动化接口对页面进行程序化操作，适用于调试、测试、流程自动化等场景。

#### 适用场景

- 网页自动化测试（打开页面、点击、输入、断言等）。
- 业务流程自动化（登录、检索、采集、表单提交等）。
- 在线问题排查（通过可视化调试观察实例状态并复现问题）。

#### 前提条件

- 已开通并进入 Agent 沙箱服务。
- 已具备创建沙箱实例的权限。

使用限制

- 浏览器实例运行时长受沙箱生命周期与配额限制。
- 自动化调试地址包含访问令牌，请妥善保管，避免泄露。
- 涉及登录态、验证码或人机校验页面时，需结合业务逻辑处理。
- 请勿在实例内存放敏感长期数据，建议任务完成后及时释放实例。

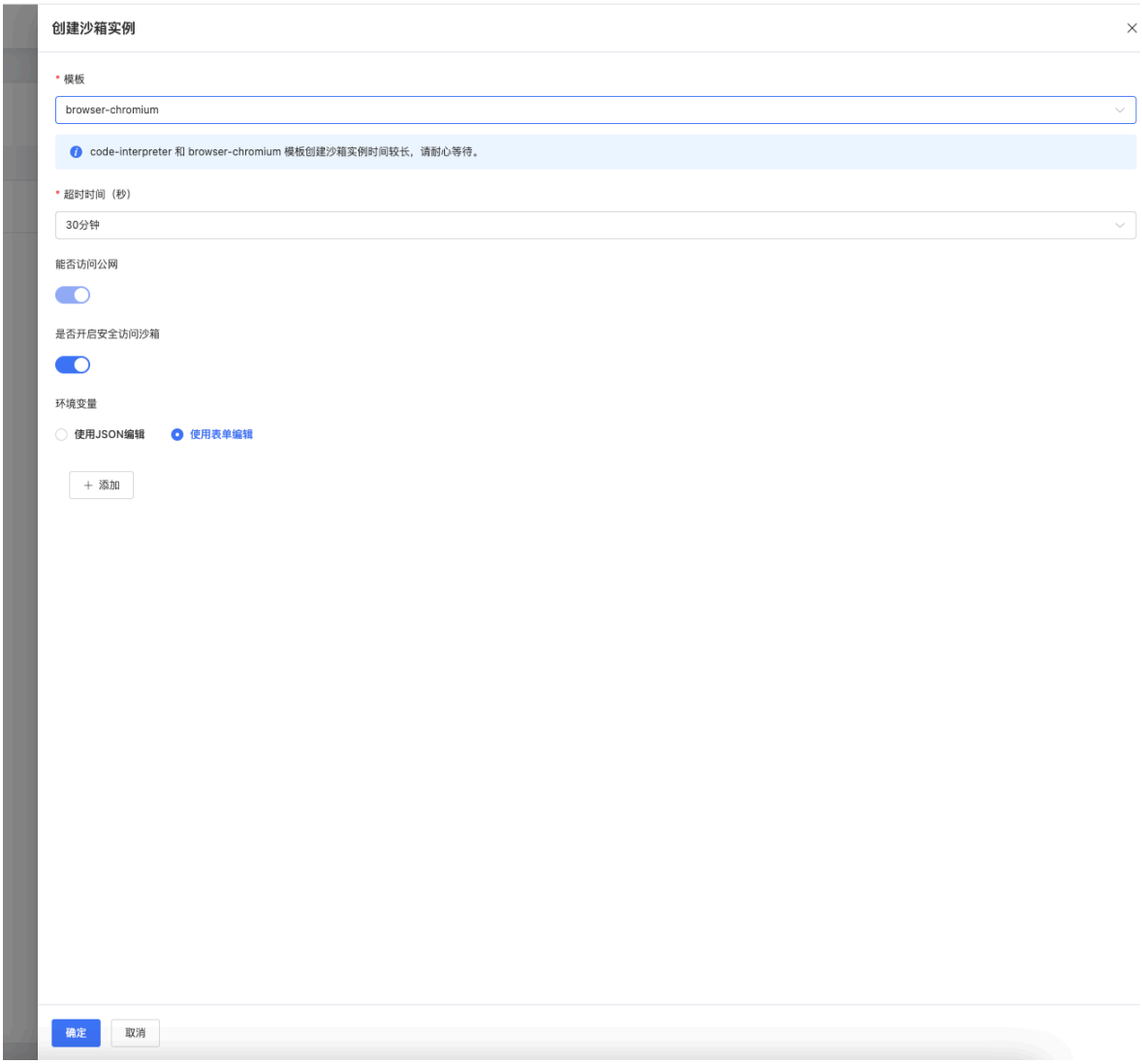
操作步骤

### 步骤一：进入沙箱实例页面

1. 登录 Agent 沙箱控制台。
2. 在左侧导航中进入**沙箱实例**。

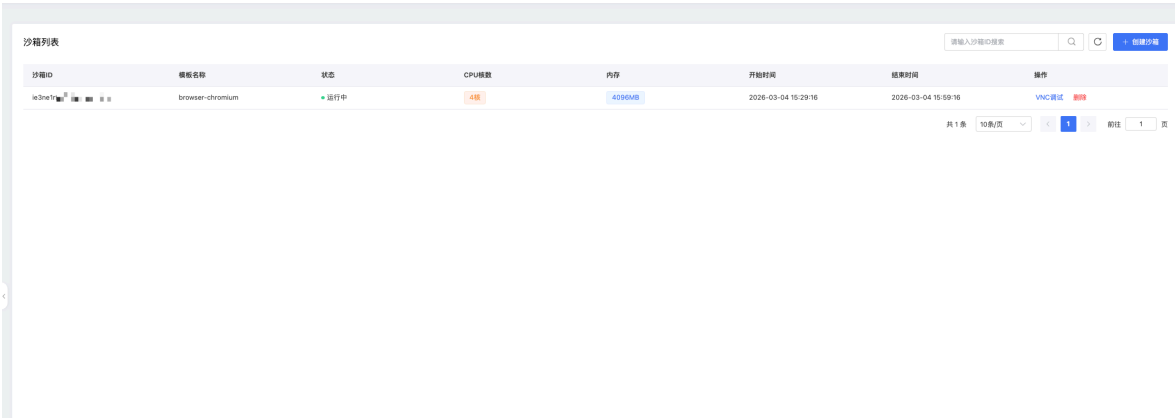
### 步骤二：创建浏览器实例

1. 点击**创建实例**。
2. 选择浏览器模板，可按需修改模板参数。
3. 点击**确认**。
4. 



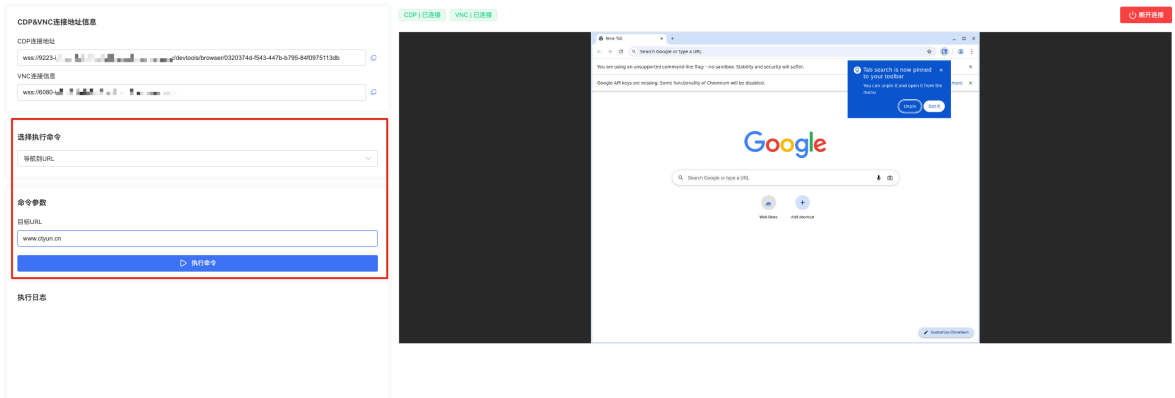

### 步骤三：进入 VNC 调试

1. 在沙箱实例列表中，选择目标实例的操作列点击**VNC 调试**。
2. 在打开的调试窗口中观察浏览器运行画面并进行交互。




### 步骤四：执行 CDP 指令

1. 在**选择执行命令**下拉列表中选择**导航到 URL**。
2. 在**命令参数**中输入目标 URL （比如 www.ctyun.cn）。
3. 点击**执行命令**，查看右侧VNC窗口中的浏览器变化。




### 步骤五：删除浏览器沙箱实例

1. 任务完成后，返回沙箱实例列表，在目标实例的操作列点击**删除**。
2. 在弹出的确认框中点击**确认**。

# 域名管理

## 自定义域名管理

### 功能说明

自定义域名管理用于配置沙箱访问地址中的域名后缀。Agent 沙箱的实际访问地址由平台分配前缀（包含实例标识等信息）与您配置的自定义域名共同组成，并通过 HTTPS 提供安全访问。

通过该功能，您可以完成以下操作：

- 新增自定义域名，并通过该域名访问沙箱。
- 绑定或更换 HTTPS 证书。
- 查看域名接入状态并进行删除管理。

**典型应用场景**

- 统一业务入口标识：将可识别的业务域名作为访问地址后缀，便于对外发布与内部协作识别。
- 多环境域名隔离：为开发、测试、生产配置不同自定义域名后缀，避免环境混用和误调用。

**使用限制**

- 不支持中文域名，自定义域名必须为泛域名，格式如 *.example.com。
- 由于域名解析本身是不区分大小写，因此自定义域名生效时会被处理成全小写。
- 必须配置 DNS 中的 CNAME 解析，若未配置会创建失败。
- 证书必须与自定义域名匹配。
- 域名总长度最大支持 128 个字符。
- 每级子域名至少包含 1 个字符，可使用字母、数字（0-9）或连字符（-）。
- 每级子域名首字符不能为连字符（-）。
- 顶级域名长度至少 2 个字符，且必须为字母。

操作步骤

### 步骤一：进入控制台并创建自定义域名

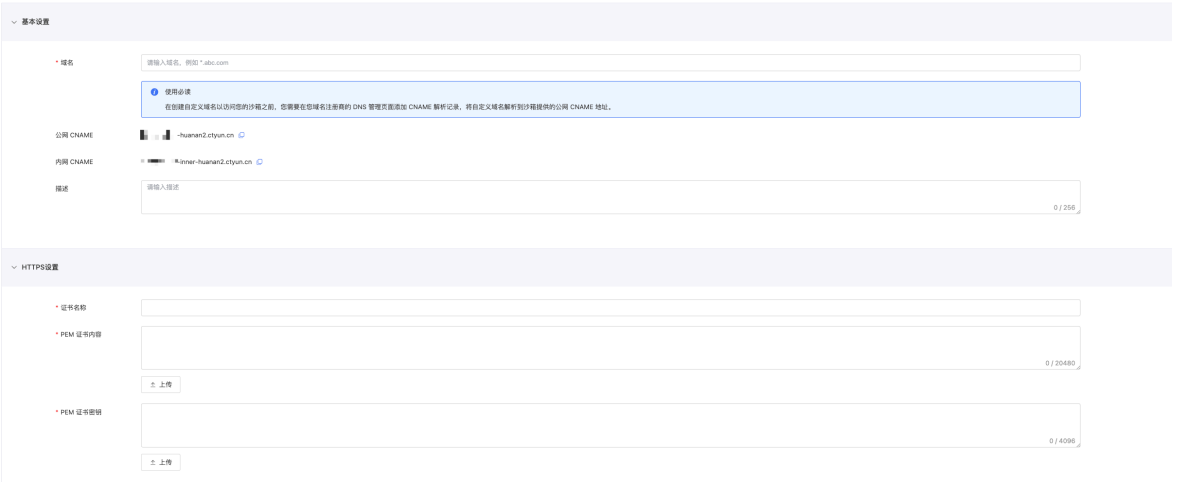
1. 登录 Agent 沙箱控制台，在左侧导航栏中选择**域名管理**。
2. 点击**创建域名**，进入创建页面。

### 步骤二：配置 DNS CNAME 解析

1. 在创建页面查看平台提供的 CNAME 目标值（公网或内网）。
2. 前往您的 DNS 服务商控制台，为自定义泛域名添加 CNAME 记录并指向上述目标值。
3. 完成解析后返回 Agent 沙箱控制台继续配置。

### 步骤三：填写必要参数

1. 在**域名**中填写自定义泛域名（例如 *.example.com）。
2. 在 HTTPS 设置中填写**证书名称**、**PEM 证书内容**与**PEM 证书密钥**。
3. 点击**确认**完成创建。




# 凭证管理

## 概述

本章节主要介绍身份凭证管理相关功能。

### 身份凭证管理

## 查看身份凭证

### 操作步骤

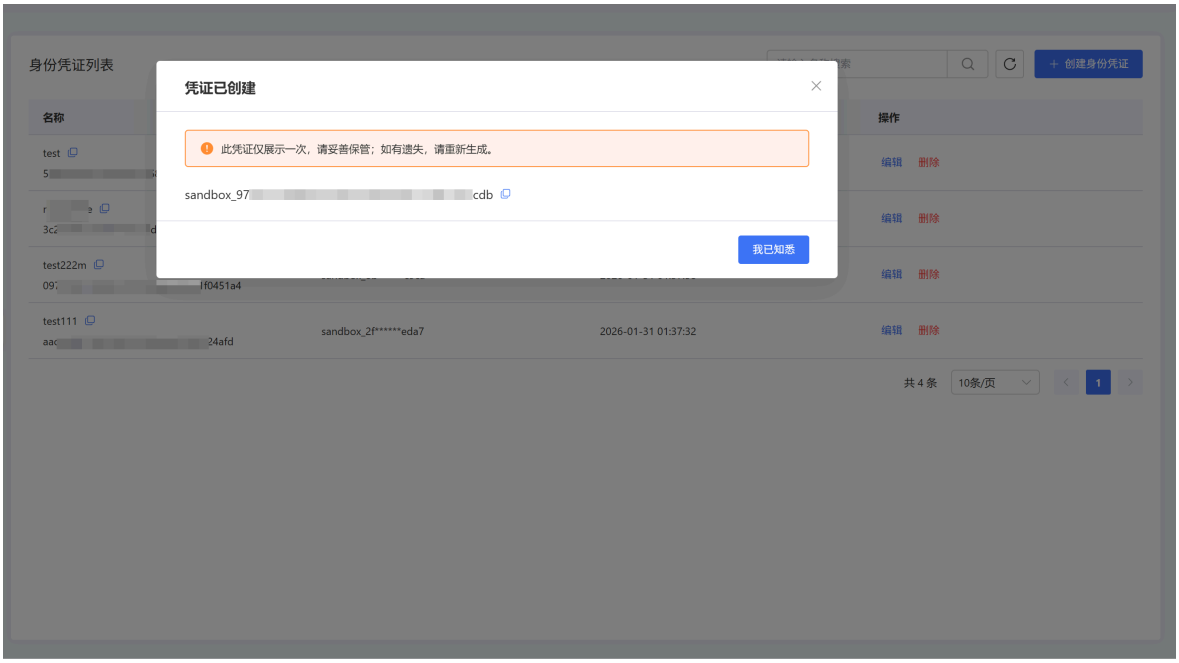
1. 登录Agent沙箱控制台。
2. 左侧菜单选择身份凭证，查看身份凭证。 

## 创建身份凭证

### 操作步骤

1. 登录Agent沙箱控制台。
2. 在身份凭证菜单下，点击创建身份凭证，填写API Key名称，点击确定。 

3. 查看API Key的值。




## 编辑身份凭证

### 操作步骤

1. 登录Agent沙箱控制台。
2. 在身份凭证菜单下，找到需要编辑的身份凭证，在操作列点击编辑，修改API Key名称，点击确定。

## 删除身份凭证

### 操作步骤

1. 登录Agent沙箱控制台。
2. 在身份凭证菜单下，找到需要删除的身份凭证，在操作列点击删除，确认无误后，点击确定。

# SDK使用指南

## 浏览器操作

### 功能说明

浏览器操作能力用于在 Agent 沙箱中启动浏览器实例，并通过自动化调试地址完成页面访问和程序化控制。

**典型应用场景**

- 页面自动化测试：在隔离沙箱中执行页面打开、点击、输入、断言等操作。
- 任务流程编排：在业务流程中通过代码驱动浏览器完成网页访问与信息采集。

- 可视化调试排障：通过控制台的 VNC 调试观察浏览器状态，复现和定位问题。

**前提条件**

- 已完成 SDK 运行环境与环境变量配置。
- 已在 Agent 沙箱服务控制台创建可用的浏览器沙箱模板，并记录模板 template_id。
- 本地已安装 Python 及 Playwright 依赖，并具备网络访问目标页面的权限。

使用限制

- template 参数必须填写已创建且可用的浏览器沙箱 template_id。
- cdp_url 依赖访问令牌，请妥善保管，避免泄露。
- 浏览器实例的可用时长受沙箱生命周期和配额策略限制。
- 当页面需要额外认证（如登录态、验证码）时，自动化脚本可能需要配套处理逻辑。

操作步骤

### 步骤一：在控制台创建浏览器沙箱模板并记录 template_id

登录 Agent 沙箱服务控制台创建浏览器沙箱模板，创建完成后记录该模板的 template_id，用于代码中创建实例。

### 步骤二：运行代码并通过 VNC 监控浏览器

运行下列代码后，在 Agent 沙箱控制台进入对应实例，打开 **VNC 调试** 查看浏览器实时画面；代码中使用 cdp_url 进行自动化操作。

```
from e2b import Sandbox
from playwright.sync_api import sync_playwright


# 请替换为您在控制台记录的template_id
template_id = "4xxxx3e9-xxxx-xxxx-xxx-f22xxx50xxxx"

# 创建浏览器沙箱实例
sandbox = Sandbox.create(template=template_id)

# 拼接 cdp url，用于通过 Playwright 操作浏览器
cdp_url = f"https://{sandbox.get_host(9223)}"
print("cdp_url:", cdp_url)
with sync_playwright() as playwright:
    # 连接到沙箱浏览器实例
    browser = playwright.chromium.connect_over_cdp(
        cdp_url,
        headers={"X-Access-Token": str(sandbox._envd_access_token)}
    )

    # 获取第一个上下文与页面并访问目标站点
    context = browser.contexts[0]
    page = context.pages[0]
```

```
page.goto("https://www.ctyun.cn")

# 输出页面标题，确认操作成功
print("title:", page.title())
```

# 常见问题

## 通用类

### 如何开通智能体引擎产品？服务收费吗？

前往天翼云官网智能体引擎产品详情页，点击【开通服务】并根据页面指引进行开通即可；本产品免费开通，公测期间服务免费。

#### 如何运行Agent沙箱？

方式一：用户可以通过控制台创建 API Key 和 自定义域名，通过web控制台创建并运行沙箱；

方式二：通过客户端安装SDK、配置环境变量后直接启动沙箱；同时，它兼容E2B使用方式，只需替换域名和API Key，即可丝滑迁移现有业务代码，快速接入并运行沙箱。

## 设置沙箱超时时间

每个沙箱都有一个 timeout（“超时时间”）配置，当沙箱的运行时间超过该配置值时，沙箱将自动关闭并释放。

您可以参考如下方式在启动的时候显式指定 timeout 的配置值，如果未指定，则默认值为 5 分钟。推荐按需设置 timeout 的值，以避免不必要的资源浪费。

```
from dotenv import load_dotenv
from e2b_code_interpreter import Sandbox

load_dotenv()

sbx = Sandbox.create(template="code-interpreter", timeout=60) # 单位为秒。

# 代码执行
execution = sbx.run_code("print('hello world')")
print(execution.logs)
```