Red Hat Enterprise Linux AI 1.5

构建和维护您的环境

创建帐户、初始化 RHEL AI、下载/组织模型以及服务/聊天自定义

Last Updated: 2025-06-27

Red Hat Enterprise Linux AI 1.5 构建和维护您的环境

创建帐户、初始化 RHEL AI、下载/组织模型以及服务/聊天自定义

法律通告

摘要

本文档提供了有关如何初始化和设置 RHEL AI 环境的说明

目录

第 1 章 为 RHEL AI 配置帐户

在与 RHEL AI 交互前，您需要设置几个帐户。

创建红帽帐户

您可以通过注册红帽网站来创建红帽帐户。您可以按照 Register for a Red Hat account 中的步骤操作。

创建 Red Hat registry 帐户

在从 Red Hat registry 下载模型前，您需要创建一个 registry 帐户并使用 CLI 登录。您可以通过在网页中选择 **Regenerate Token** 按钮来查看您的帐户用户名和密码。

1. 您可以通过选择 Registry Service Accounts 页面中的 **New Service Account** 按钮来创建 Red Hat registry 帐户。
2. 您可以通过 CLI 登录 registry 帐户的方法有几种。按照 Red Hat Container Registry 身份验证中的步骤在您的机器上登录。

为混合云部署配置 Red Hat Insights

Red Hat Insights 是一个产品，可让您了解您要部署的环境。这个平台还可帮助识别系统中的操作和漏洞风险。有关 Red Hat Insights 的更多信息，请参阅 Red Hat Insights 数据和应用程序安全性 。

- 您可以按照 查看 激活码中的步骤，使用激活码和机构参数创建 Red Hat Insights 帐户。然后您可以运行以下命令来在机器上配置您的帐户：

```
$ sudo rhc connect --organization <org id> --activation-key <created key>
```

要在断开连接的环境中运行 RHEL AI，或选择不使用 Red Hat Insights，请运行以下命令：

```
$ sudo mkdir -p /etc/ilab
$ sudo touch /etc/ilab/insights-opt-out
```

- 您还可以运行以下命令来启用持久性凭证并保持登录到 registry：

```
$ podman login registry.redhat.io
```

然后，将 **auth.json** 文件添加到 **/etc/ostree/** 目录中。

```
$ sudo cp /run/user/1000/containers/auth.json /etc/ostree/
```

这可让您在升级 Red Hat Enterprise Linux AI 后登录到 Red Hat registry。

注意

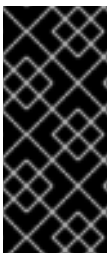
如果您的系统被配置为 root 用户，则在运行命令时不需要使用 sudo。

第 2 章 初始化 INSTRUCTLAB

您必须初始化 InstructLab 环境，才能开始使用 Red Hat Enterprise Linux AI 模型。

2.1. 创建 RHEL AI 环境

您可以通过初始化 InstructLab 环境，开始与 LLM 和 RHEL AI 工具交互。



重要

AMD 机器的系统配置集只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

先决条件

- 已使用可引导容器镜像安装了 RHEL AI。
- 在机器上具有 root 用户访问权限。

流程

1. 可选：您可以运行以下命令来查看您的机器信息：

```
$ ilab system info
```

2. 运行以下命令来初始化 InstructLab ：

```
$ ilab config init
```

3. RHEL AI CLI 开始设置您的环境和 **config.yaml** 文件。CLI 会自动检测机器的硬件并根据 GPU 类型选择系统配置文件。系统配置集会根据检测到的硬件使用正确的参数值填充 **config.yaml** 文件。

配置集自动检测示例

```
Generating config file and profiles:
    /home/user/.config/instructlab/config.yaml
    /home/user/.local/share/instructlab/internal/system_profiles/

We have detected the NVIDIA H100 X4 profile as an exact match for your system.

-------------------------------------------
  Initialization completed successfully!
You're ready to start using `ilab`. Enjoy!
-------------------------------------------
```

4. 如果 CLI 不会检测到系统的确切匹配项，则在提示时手动选择系统配置文件。选择与您的系统匹配的硬件厂商和配置。

选择系统配置集的输出示例

```
Please choose a system profile to use.
System profiles apply to all parts of the config file and set hardware specific defaults for each
command.
First, please select the hardware vendor your system falls into
[0] NO SYSTEM PROFILE
[1] NVIDIA
Enter the number of your choice [0]: 4
You selected: NVIDIA
Next, please select the specific hardware configuration that most closely matches your
system.
[0] No system profile
[1] NVIDIA H100 X2
[2] NVIDIA H100 X8
[3] NVIDIA H100 X4
[4] NVIDIA L4 X8
[5] NVIDIA A100 X2
[6] NVIDIA A100 X8
[7] NVIDIA A100 X4
[8] NVIDIA L40S X4
[9] NVIDIA L40S X8
Enter the number of your choice [hit enter for hardware defaults] [0]: 3
```

完成的 ilab 配置 init 运行 的输出示例。

```
You selected:
/Users/<user>/.local/share/instructlab/internal/system_profiles/nvidia/H100/h100_x4.yaml

-------------------------------------------
   Initialization completed successfully!
  You're ready to start using `ilab`. Enjoy!
-------------------------------------------
```

5. 如果要使用框架工具箱（包括两个技能和一个知识 **qna.yaml** 文件），您可以克隆 skeleton 存储库并将其放置在 **taxonomy** 目录中：

```
rm -rf ~/.local/share/instructlab/taxonomy/ ; git clone https://github.com/RedHatOfficial/rhelai-sample-taxonomy.git ~/.local/share/instructlab/taxonomy/
```

6. 如果系统配置集被自动探测到，您可以运行以下命令：

```
$ ilab config init --profile <path-to-system-profile>
```

其中

<path-to-system-profile>

指定到正确的系统配置文件的路径。您可以在 **~/.local/share/instructlab/internal/system_profiles** 路径中找到系统配置文件。

配置集选择命令示例

```
$ ilab config init --profile
~/.local/share/instructlab/internal/system_profiles/amd/mi300x/mi300x_x8.yaml
```

InstructLab 环境的目录结构

```
├─ ~/.config/instructlab/config.yaml 1
├─ ~/.cache/instructlab/models/ 2
├─ ~/.local/share/instructlab/datasets 3
├─ ~/.local/share/instructlab/taxonomy 4
├─ ~/.local/share/instructlab/phased/<phase1-or-phase2>/checkpoints/ 5
```

1. **~/.config/instructlab/config.yaml**: 包含 **config.yaml** 文件。
2. **~/.cache/instructlab/models/** ：包含所有下载的大型语言模型，包括您使用 RHEL AI 生成的已保存的输出。
3. **~/.local/share/instructlab/datasets/** ：包含 SDG 阶段的数据输出，基于对 taxonomy 存储库的修改而构建。
4. **~/.local/share/instructlab/taxonomy/**: 包含技术和知识数据。
5. **~/.local/share/instructlab/phased/<phase1-or-phase2>/checkpoints/**: 包含多阶段培训进程的输出

验证

1. 您可以运行以下命令来查看完整的 **config.yaml** 文件

```
$ ilab config show
```

2. 您还可以运行以下命令来手动编辑 **config.yaml** 文件：

```
$ ilab config edit
```

第 3 章 下载大型语言模型

Red Hat Enterprise Linux AI 允许您使用由红帽和 IBM 提供的和构建的各种 Large Language Models (LLM)自定义或聊天。您可以从 Red Hat RHEL AI registry 下载这些模型。您可以将任何自定义模型上传到 S3 存储桶。

表 3.1. Red Hat Enterprise Linux AI 版本 1.5 LLMs

大型语言模型 (LLM)	类型	Size	用途	模型系列	NVIDIA 加速器支持	AMD 加速器支持	Intel 加速器支持
granite-3.1-8b-starter-v2.1	LAB 调优的 granite 初学者模型	16.0 GB	默认 Granite 3.1 基本模型的版本 2，用于自定义和微调	Granite 3.1	正式发布	正式发布	不可用
granite-3.1-8b-lab-v2.1	LAB 调优的 granite 模型	16.0 GB	用于 inference 服务的默认 Granite 3.1 模型的版本 2	Granite 3.1	正式发布	正式发布	不可用
granite-3.1-8b-starter-v2	LAB 调优的 granite 初学者模型	16.0 GB	默认 Granite 3.1 基本模型的版本 2，用于自定义和微调	Granite 3.1	不可用	不可用	技术预览
granite-3.1-8b-lab-v2	LAB 调优的 granite 模型	16.0 GB	用于 inference 服务的默认 Granite 3.1 模型的版本 2	Granite 3.1	不可用	不可用	技术预览
granite-8b-code-instruct	LAB 调优的 granite 代码模型	15.0 GB	LAB 调优的 granite 代码模型用于 inference 服务	Granite Code 模型	技术预览	技术预览	技术预览
granite-8b-code-base	Granite 调优的代码模型	15.0 GB	Granite 代码模型用于 inference 服务	Granite Code 模型	技术预览	技术预览	技术预览
mixtral-8x7b-instruct-v0-1	默认指导模型	87.0 GB	用于运行 Synthetic 数据生成(SDG)的默认指导模型。	Mixtral	正式发布	正式发布	技术预览
llama-3.3-70b-Instruct	可选指导模型	74.0 GB	用于运行 Synthetic 数据生成(SDG)的可选指导模型。	llama	技术预览	不可用	不可用

大型语言模型 (LLM)	类型	Size	用途	模型系列	NVIDIA 加速器支持	AMD 加速器支持	Intel 加速器支持
prometheus-8x7b-v2-0	评估判断模型	87.0 GB	judge 模式用于多阶段培训和认证	Prometheus 2	正式发布	正式发布	技术预览

重要

使用 granite-8b-code-instruct 或 granite-8b-code-base Large Language 模型(LLM)只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

自定义 Granite LLM 所需的模型

- **granite-7b-starter** 或 **granite-8b-starter-v1** 基本 LLM，具体取决于您的硬件供应商。
- SDG 的 **mixtral-8x7b-instruct-v0-1** 教授器模型。
- **prometheus-8x7b-v2-0** judge 模型用于培训和认证。

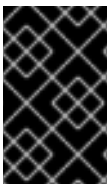
自定义 LLM 所需的其他工具

Low-rank adaptation (LoRA)适应者提高了 Synthetic Data Generation (SDG)流程的效率。

- SDG 的 **skills-adapter-v3** LoRA 分层技能适配器。
- SDG 的 **knowledge-adapter-v3** LoRA 分层知识适配器。

下载适应器的命令示例

```
$ ilab model download --repository docker://registry.redhat.io/rhelai1/knowledge-adapter-v3 --release latest
```



重要

LoRA 分层适配器不会显示在 **ilab model list** 命令的输出中。您可以在 **ls ~/.cache/instructlab/models** 文件夹中看到 **skills-adapter-v3** 和 **knowledge-adapter-v3** 文件。

3.1. 从红帽软件仓库下载模型

您可以下载由红帽和 IBM 创建的其他可选模型。

先决条件

- 已使用可引导容器镜像安装了 RHEL AI。

- 您初始化了 InstructLab。
- 您创建了红帽 registry 帐户并登录您的机器中。
- 在机器上具有 root 用户访问权限。

流程

1. 要下载额外的 LLM 模型，请运行以下命令：

```
$ ilab model download --repository docker://<repository_and_model> --release <release>
```

其中：

<repository_and_model>

指定模型和模型的存储库位置。您可以从 **registry.redhat.io/rhelai1/** 仓库访问模型。

<release>

指定模型的版本。对于 RHEL AI 版本 **1.5** 支持的模型，设置为 1.5。对于 **最新版本的** 模型，设置为 latest。

示例命令

```
$ ilab model download --repository docker://registry.redhat.io/rhelai1/granite-3.1-8b-starter-v1
--release latest
```

验证

1. 您可以使用以下命令查看系统上所有下载的模型，包括培训后的新模型：

```
$ ilab model list
```

输出示例

```
+-----------------------------------+--------------------+---------+
| Model Name                        | Last Modified      | Size    |
+-----------------------------------+--------------------+---------+
| models/prometheus-8x7b-v2-0       | 2024-08-09 13:28:50 |  87.0 GB|
| models/mixtral-8x7b-instruct-v0-1 | 2024-08-09 13:28:24 |  87.0 GB|
| models/granite-3.1-8b-starter-v1  | 2024-08-09 14:28:40 |  16.6 GB|
| models/granite-3.1-8b-lab-v1      | 2024-08-09 14:40:35 |  16.6 GB|
+-----------------------------------+--------------------+---------+
```

2. 您还可以通过运行以下命令来列出 **ls ~/.cache/instructlab/models** 文件夹中下载的模型：

```
$ ls ~/.cache/instructlab/models
```

输出示例

```
granite-3.1-8b-starter-v1
granite-3.1-8b-lab-v1
```

第 4 章 模型管理

您可以在 RHEL AI 上组织和管理您的自定义或下载模型的不同方法。

4.1. 将模型上传到 REGISTRY

微调模型后，您可以将模型上传到外部 registry。RHEL AI 目前支持将模型上传到 AWS S3 存储桶。

先决条件

- 您已在首选平台上安装了 RHEL AI。
- 您初始化了 InstructLab。
- 登录到您首选的 registry。

流程

1. 您可以使用以下命令将模型上传到特定 registry

```
$ ilab model upload --model <name-of-model> --destination <registry-location> --dest-type
<registry-type>
```

其中：

<name-of-model>
指定您要上传的检查点名称。例如,- **-model samples_0801**。您还可以指定检查点的路径。

<registry-location>
指定您要上传模型的位置。例如，- **-destination example-s3-bucket**

<registry-type>
指定 model 类型。有效值包括： **s3**。

ilab 模型上传 命令示例到 s3 存储桶

```
$ ilab model upload --model samples_0801 --destination example-s3-bucket --dest-type s3
```

第 5 章 使用模型提供和聊天

要与 Red Hat Enterprise Linux AI 上的各种模型交互，您必须提供在服务器中托管它的模型，然后您可以使用模型进行聊天。

5.1. 提供模型

要与模型交互，您必须首先通过服务激活机器中的模型。**ilab 模型服务** 命令启动 vLLM 服务器，允许您与模型进行聊天。

先决条件

- 已使用可引导容器镜像安装了 RHEL AI。
- 您初始化了 InstructLab。
- 已安装您首选的 Granite LLM。
- 在机器上具有 root 用户访问权限。

流程

1. 如果没有指定模型，您可以通过运行以下命令来提供默认模型 **granite-7b-redhat-lab** ：

```
$ ilab model serve
```

2. 要服务特定模型，请运行以下命令

```
$ ilab model serve --model-path <model-path>
```

示例命令

```
$ ilab model serve --model-path ~/.cache/instructlab/models/granite-8b-code-instruct
```

模型提供并就绪时的输出示例

```
INFO 2024-03-02 02:21:11,352 lab.py:201 Using model 'models/granite-8b-code-instruct'
with -1 gpu-layers and 4096 max context size.
Starting server process
After application startup complete see http://127.0.0.1:8000/docs for API.
Press CTRL+C to shut down the server.
```

5.1.1. 可选：运行 **ilab 模型** 作为服务

您可以设置 **systemd** 服务，以便 **ilab 模型服务** 命令作为正在运行的服务运行。**systemd** 服务在后台运行 **ilab model serving** 命令，并在其崩溃或失败时重新启动。您可以将服务配置为在系统启动时启动。

先决条件

- 您在裸机上安装了 Red Hat Enterprise Linux AI 镜像。
- 您初始化的 InstructLab

- 您下载了您首选的 Granite LLMs。
- 在机器上具有 root 用户访问权限。

流程.

1. 运行以下命令，为您的 **systemd** 用户服务创建一个目录：

```
$ mkdir -p $HOME/.config/systemd/user
```

2. 使用以下示例配置创建 **systemd** 服务文件：

```
$ cat << EOF > $HOME/.config/systemd/user/ilab-serve.service
[Unit]
Description=ilab model serve service

[Install]
WantedBy=multi-user.target default.target (1)

[Service]
ExecStart=ilab model serve --model-family granite
Restart=always
EOF
```

(1) 指定在引导时默认启动。

3. 运行以下命令来重新载入 **systemd** Manager 配置：

```
$ systemctl --user daemon-reload
```

4. 运行以下命令启动 **ilab 模型服务 systemd** 服务：

```
$ systemctl --user start ilab-serve.service
```

5. 您可以使用以下命令检查该服务是否正在运行：

```
$ systemctl --user status ilab-serve.service
```

6. 您可以运行以下命令来检查服务日志：

```
$ journalctl --user-unit ilab-serve.service
```

7. 要允许服务在引导时启动，请运行以下命令：

```
$ sudo loginctl enable-linger
```

8. 可选：您可以运行几个可选命令来维护 **systemd** 服务。
   - 您可以运行以下命令来停止 ilab-serve 系统服务：

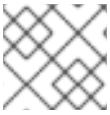
```
$ systemctl --user stop ilab-serve.service
```

- 您可以通过从 **$HOME/.config/systemd/user/ilab-serve.service** 文件中删除 **"WantedBy=multi-user.target default.target"** 来防止服务在引导时启动。

5.1.2. 可选：允许从安全端点访问模型

您可以服务 inference 端点，并通过创建 **systemd** 服务并设置公开安全端点的 nginx 反向代理，让其他人与 Red Hat Enterprise Linux AI 在安全连接上交互。这可让您与其他端点共享安全端点，以便它们可以通过网络与模型进行聊换。

以下流程使用自签名认证，但建议使用由可信证书颁发机构(CA)发布的证书。



注意

只有在裸机平台上才支持以下步骤。

先决条件

- 您已在裸机上安装了 Red Hat Enterprise Linux AI 镜像。
- 您初始化的 InstructLab
- 您下载了您首选的 Granite LLMs。
- 在机器上具有 root 用户访问权限。

流程

1. 运行以下命令，为您的证书文件和密钥创建一个目录：

```
$ mkdir -p `pwd`/nginx/ssl/
```

2. 运行以下命令，使用正确的配置创建 OpenSSL 配置文件：

```
$ cat > openssl.cnf <<EOL
[ req ]
default_bits = 2048
distinguished_name = <req-distinguished-name> (1)
x509_extensions = v3_req
prompt = no

[ req_distinguished_name ]
C  = US
ST = California
L  = San Francisco
O  = My Company
OU = My Division
CN = rhelai.redhat.com

[ v3_req ]
subjectAltName = <alt-names> (2)
basicConstraints = critical, CA:true
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
```

```
[ alt_names ]
DNS.1 = rhelai.redhat.com 3
DNS.2 = www.rhelai.redhat.com 4
```

1 为您的要求指定可分辨名称。

2 为您的要求指定备用名称。

3 4 指定 RHEL AI 的服务器通用名称。在示例中，服务器名称为 **rhelai.redhat.com**。

3. 使用以下命令生成启用了 Subject Alternative Name (SAN)的自签名证书：

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
`pwd`/nginx/ssl/rhelai.redhat.com.key -out `pwd`/nginx/ssl/rhelai.redhat.com.crt -config
openssl.cnf
```

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
```

4. 运行以下命令，创建 Nginx 配置文件并将其添加到 **'pwd**/nginx/conf.d' 中：

```
mkdir -p `pwd`/nginx/conf.d

echo 'server {
    listen 8443 ssl;
    server_name <rhelai.redhat.com> 1

    ssl_certificate /etc/nginx/ssl/rhelai.redhat.com.crt;
    ssl_certificate_key /etc/nginx/ssl/rhelai.redhat.com.key;

    location / {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
' > `pwd`/nginx/conf.d/rhelai.redhat.com.conf
```

1 指定服务器的名称。在示例中，服务器名称为 **rhelai.redhat.com**

5. 运行以下命令，使用新配置运行 Nginx 容器：

```
$ podman run --net host -v `pwd`/nginx/conf.d:/etc/nginx/conf.d:ro,Z -v
`pwd`/nginx/ssl:/etc/nginx/ssl:ro,Z nginx
```

如果要使用端口 443，则必须以 root 用户身份运行 **podman run** 命令。

6. 现在，您可以使用安全端点 URL 连接到服务 ilab 机器。示例命令：

```
$ ilab model chat -m /instructlab/instructlab/granite-7b-redhat-lab --endpoint-url
```

7. 您还可以使用以下命令连接到服务 RHEL AI 机器：

```
$ curl --location 'https://rhelai.redhat.com:8443/v1' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer <api-key>' \
--data '{
  "model": "/var/home/cloud-user/.cache/instructlab/models/granite-7b-redhat-lab",
  "messages": [
      {
          "role": "system",
          "content": "You are a helpful assistant."
      },
      {
          "role": "user",
          "content": "Hello!"
      }
  ]
}' | jq .
```

其中

<api-key>

指定 API 密钥。您可以按照"创建 API 密钥以使用模型进行聊天"中的步骤创建自己的 API 密钥。

8. 可选： 您还可以获取服务器证书并将其附加到 Certifi CA 捆绑包中

   a. 运行以下命令来获取服务器证书：

   ```
   $ openssl s_client -connect rhelai.redhat.com:8443 </dev/null 2>/dev/null | openssl x509 -outform PEM > server.crt
   ```

   b. 将证书复制到您系统的可信 CA 存储目录中，并使用以下命令更新 CA 信任存储：

   ```
   $ sudo cp server.crt /etc/pki/ca-trust/source/anchors/
   ```

   ```
   $ sudo update-ca-trust
   ```

   c. 您可以运行以下命令来将证书附加到 Certifi CA 捆绑包：

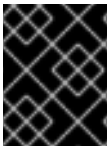
   ```
   $ cat server.crt >> $(python -m certifi)
   ```

   d. 现在，您可以使用自签名证书运行 **ilab 模型聊天**。示例命令：

   ```
   $ ilab model chat -m /instructlab/instructlab/granite-7b-redhat-lab --endpoint-url https://rhelai.redhat.com:8443/v1
   ```

5.2. 使用模型进行聊天

提供模型后，您现在可以使用模型进行聊天。



重要

要聊天的型号必须与您要提供的型号匹配。使用默认 **config.yaml** 文件，**granite-7b-redhat-lab** 模型是服务及聊天的默认模型。

先决条件

- 已使用可引导容器镜像安装了 RHEL AI。
- 您初始化了 InstructLab。
- 您下载了您首选的 Granite LLMs。
- 您提供模型。
- 在机器上具有 root 用户访问权限。

流程

1. 由于您在一个终端窗口中提供模型，您必须打开另一个终端来与模型进行聊天。
2. 要使用默认模型进行聊天，请运行以下命令：

```
$ ilab model chat
```

3. 要与特定模型进行聊天，请运行以下命令：

```
$ ilab model chat --model <model-path>
```

示例命令

```
$ ilab model chat --model ~/.cache/instructlab/models/granite-8b-code-instruct
```

chatbot 的输出示例

```
$ ilab model chat
╭──────────────────────────────────────────────────────────────────────────────────────────────────── system ────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ Welcome to InstructLab Chat w/ GRANITE-8B-CODE-INSTRUCT (type /h for help)                                                                                                                                  │
╰─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
>>>
[S][default]
```

+ 键入 **exit** 以离开 chatbot。

5.2.1. 可选：创建一个 API 密钥以使用模型进行聊天

默认情况下，**ilab** CLI 不使用身份验证。如果要将服务器公开给互联网，您可以创建一个 API 密钥，以按照以下流程连接到您的服务器。

先决条件

- 您在裸机上安装了 Red Hat Enterprise Linux AI 镜像。
- 您初始化的 InstructLab
- 您下载了您首选的 Granite LLMs。
- 在机器上具有 root 用户访问权限。

流程

1. 运行以下命令，创建一个在 **$VLLM_API_KEY** 参数中保留的 API 密钥：

```
$ export VLLM_API_KEY=$(python -c 'import secrets; print(secrets.token_urlsafe())')
```

2. 您可以运行以下命令来查看 API 密钥：

```
$ echo $VLLM_API_KEY
```

3. 运行以下命令来更新 **config.yaml** ：

```
$ ilab config edit
```

4. 将以下参数添加到 **config.yaml** 文件的 **vllm_args** 部分。

```
serve:
    vllm:
        vllm_args:
        - --api-key
        - <api-key-string>
```

其中

<api-key-string>
指定 API 密钥字符串。

5. 您可以运行以下命令来验证服务器是否使用 API 密钥身份验证：

```
$ ilab model chat
```

然后，查看显示未授权用户的以下错误。

```
openai.AuthenticationError: Error code: 401 - {'error': 'Unauthorized'}
```

6. 运行以下命令验证您的 API 密钥是否正常工作：

```
$ ilab model chat -m granite-7b-redhat-lab --endpoint-url https://inference.rhelai.com/v1 --api-key $VLLM_API_KEY
```

输出示例

```
$ ilab model chat
╭──────────────────────────────────────────────────────────────────────────────╮
│ system                                                                       │
╰──────────────────────────────────────────────────────────────────────────────╯
╭──────────────────────────────────────────────────────────────────────────────╮
│ Welcome to InstructLab Chat w/ GRANITE-7B-LAB (type /h for help)             │
╰──────────────────────────────────────────────────────────────────────────────╯
>>>
[S][default]
```