



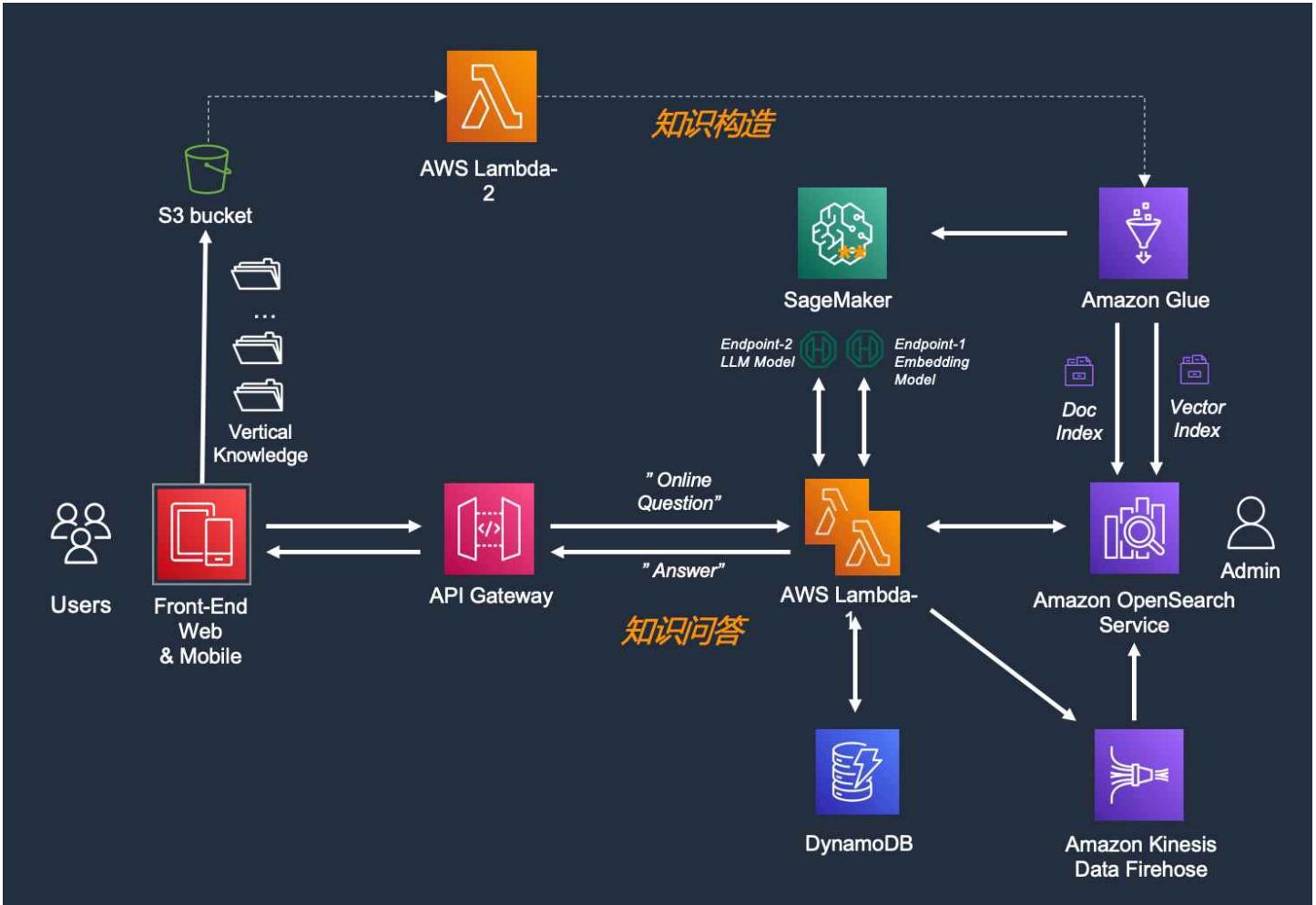
RAG+LLM Chatbot 部署指南

修订记录：

版本	修订时间	说明
v1.1.3	2023/11/7	跟release 1.1.3 对应

1. 整体架构说明

- 采用serverless架构，主要使用的AWS 服务如下图所示
- 前端和后端独立部署，通过api gateway或者lambda function调用。如果前后端都在同账号/Region下，通过lambda function call直接调用，如果跨账号或者region，则通过api gateway endpoint调用。



2. 环境准备

2.1 AWS账号准备

区域说明：

本手册涉及创建的资源可放到中国区或者global区，请确保您的账号满足如下要求。

可用资源数量：

- VPC: 1 个 VPC，因此，您需要确认当前区域 VPC 的额度足够（默认上限为 5 / Region）。
- EIP: 2 个 EIP，因此，您需要确认当前区域 EIP 的额度足够（默认上限为 5 / Region）。

2.2 创建EC2实例用于运行部署CDK代码,并且host前端代码

(如果本地机器有开发环境，且有配置AWS管理员账号的ak/sk，也可以直接在本机上操作部署)

操作系统：Amazon Linux 2023

实例类型：t3.large 或者c5.large

EBS卷：gp3 40G

是否创建密钥对：否，如果需要使用本地terminal登录则选是

网络安全组要求：开启22端口，能SSH到该实例

EC2 附加一个 IAM role,

对于开发人员，赋予**AdministratorAccess**，会避免各种权限引起的错误。

1. 选中ec2实例，修改IAM角色

The screenshot shows the AWS EC2 Instances page. A single instance named 'botdeploy' (ID: i-03be99b4c85898dd4) is selected. In the bottom right corner of the instance card, there is a blue-bordered button labeled '修改 IAM 角色' (Modify IAM Role), which is the target of a red box.

2. 创建新角色

This screenshot shows the 'Modify IAM Role' dialog for the selected EC2 instance. On the left, under 'IAM 角色' (IAM Role), there is a dropdown menu set to '无 IAM 角色' (No IAM Role). To its right is a blue-bordered button labeled '创建新的 IAM 角色' (Create New IAM Role). A red box highlights this button. Below the dropdown, a yellow warning box asks if the user wants to remove all existing IAM roles from the instance if they choose to add a new one. At the bottom right of the dialog is a yellow '更新 IAM 角色' (Update IAM Role) button.

On the right, the 'Select Trusted Entity' dialog is open. It lists four options: '亚马逊云科技 - 服务 (Amazon Web Services - Service)' (selected), '亚马逊云科技 - 用户 (Amazon Web Services - User)' (disabled), 'SAML 2.0 身份 (SAML 2.0 Identity)' (disabled), and 'OpenID Connect 客户端 (OpenID Connect Client)' (disabled). Below this is a section titled '使用案例' (Use Cases) with a note about EC2. At the bottom, there is a 'EC2' service selection dropdown and a 'EC2 Role for AWS Systems Manager' checkbox, which is checked and highlighted with a red box.

3. 追加admin权限

This screenshot shows the IAM Policies page. Under the '权限策略' (Policies) tab, there are two policies listed: 'AdministratorAccess' and 'AmazonSSMManagedInstanceCore'. The 'AdministratorAccess' policy is highlighted with a red box. The 'AmazonSSMManagedInstanceCore' policy is also visible below it.

4. 重新回到EC2 console，刷新角色列表，选中新增的角色

修改 IAM 角色 信息

将 IAM 角色附加到您的实例。

实例 ID
i-03be99b4c85898dd4 (botdeploy)

IAM 角色
选择要附加到您的实例的 IAM 角色，如果您尚未创建任何 IAM 角色，也可以新建一个。您选择的角色会取代当前附加到您的实例的任何角色。

adminrole-for-ec2 [创建新的 IAM 角色](#)

2.3 登陆EC2， 安装环境

1. 在EC2 控制台使用EC2 connect to instance登录

EC2 > Instances > i-03be99b4c85898dd4 > Connect to instance

Connect to instance Info

Connect to your instance i-03be99b4c85898dd4 (botdeploy) using any of these options

[EC2 Instance Connect](#) [Session Manager](#) [SSH client](#) [EC2 serial console](#)

2. 安装nodejs 18

```
1 sudo yum install https://rpm.nodesource.com/pub_18.x/nodistro/repo/nodesource-re
2 sudo yum install nodejs -y --setopt=nodejs.module_hotfixes=1
```

3. 安装 &启动 docker

```
1 sudo yum install docker -y
2 sudo service docker start
3 sudo chmod 666 /var/run/docker.sock
```

4. 安装git

```
1 sudo yum install git -y
```

5. 安装aws-cdk， 依次运行下面三条命令

```
1 npm install  
2 sudo npm install -g aws-cdk  
3 sudo npm install --global yarn
```

3. 部署说明-后端部分

3.1 下载代码

3.1.1 通用方式

登陆到EC2以后，通过如下命令下载代码

```
1 git clone https://github.com/aws-samples/private-llm-qa-bot.git
```

3.1.2 中国区Wordaround

如果因为在中国区，网络原因导致下载失败，可以执行下面代码先下载到本地再通过S3 Bucket进行中转。

1. 下载repo到本地，并配置好aws的credentials

```
1 cat ~/.aws/credentials  
2  
3 [default]  
4 aws_access_key_id = *****  
5 aws_secret_access_key = *****  
6  
7 [BJS_ACC]  
8 aws_access_key_id = *****  
9 aws_secret_access_key = *****
```

2. 本地命令行执行如下脚本，同步代码到S3中，脚本中的region需要进行相应修改

```
1 region="cn-northwest-1"  
2 profile_name="BJS_ACC"  
3 timestamp=$(date +%s)  
4 bucket_name="code-temp-dir-$timestamp"  
5 aws s3 mb "s3://$bucket_name" --region $region --profile $profile_name  
6  
7 aws s3 sync ./private-llm-qa-bot "s3://$bucket_name/private-llm-qa-bot" --  
profile "$profile_name" --region "$region"
```

3. 切换到Ec2，执行如下脚本，从S3上同步下来，注意bucket_name和region需要手动设定

```
1 aws s3 sync "s3://$bucket_name/private-llm-qa-bot" private-llm-qa-bot --region $region
```

3.2 自动化部署

3.2.1 通用方式

在命令行执行如下代码

```
1 # 进入到deploy目录
2 cd private-llm-qa-bot/deploy/
3
4 #设置当前region
5 export region=${region}
6
7 #生成环境变量
8 sh gen_env.sh ${region}
9
10 #执行cdk部署程序
11 aws iam create-service-linked-role --aws-service-name es.amazonaws.com
12
13 #如果这个region有其他项目执行过这个bootstrap，会提示报错，忽略即可
14 cdk bootstrap
15 cdk synth
16 cdk deploy --require-approval never --all
```

注意：整个部署时间约25-30分钟，部署完毕以后，可以在CloudFormation中看到如下Stack信息

The screenshot shows the AWS CloudFormation console with the 'Stacks' section open. A specific stack, 'QAChatDeployStack', is selected. On the right, the 'Outputs' tab is active, displaying a list of outputs with their keys and values. The outputs include:

- APIgatewayendpointurl: https://acwltxw7.execute-api.us-west-2.amazonaws.com/prod/
- DownloadKeyCommand: aws secretsmanager get-secret-value --secret-id ec2-ssh-key/cdk-keypair/private --query SecretString --output text > cdk-key.pem && chmod 400 cdk-key.pem
- embeddingendpoint: 9.128.111.123:3007-23-05-18-07-05-20-embedding
- embeddingmodelname: 9.128.111.123:3007-23-05-18-07-05-20-embedding
- GlueJobName: chatbotfroms3taoosf96BA633-9qj0y5Wm8qvz
- KinesisFirehoseRole: QAChatDeployStack-chatbotkinesisfirehoseAAD8C891-7QSO7TEE715K
- llmchatglmendpoint: 9.128.111.123:3007-23-05-18-07-05-20-llm-chatglm
- modelname: 9.128.111.123:3007-23-05-18-07-05-20-llm-chatglm
- OpenSearchEC2ProxyAddress: http://34.214.15.111:8081/_dashboards/
- opensearchendpoint: vpc-domain66ac69e0-p189l5ymkgar-v64op5yrjkmn3ptemmnjhwkne.us-west-2.es.amazonaws.com
- region: us-west-2
- sshcommand: ssh -i cdk-key.pem -o IdentitiesOnly=yes ec2-user@ec2-34-214-15-111.us-west-2.compute.amazonaws.com
- UPLOADBUCKET: 7-23-05-18-07-05-20-bucket

3.2.2 中国区Wordaround

如果是在中国区部署，包安装很可能会失败，请参考下面建议：

在多个private-llm-qa-bot/code/{main, query_rewriter, intention_detect, chat_agent}目录中 Dockfile 文件，需要进行如下改动指定包安装的源

```

1 ...
2 COPY requirements.txt .
3 +RUN pip3 config set global.index-url https://mirrors.ustc.edu.cn/pypi/web/simple
4 RUN pip3 install -r requirements.txt --target "${LAMBDA_TASK_ROOT}"
5 ...

```

3.3 构建知识库

1. 创建Amazon OpenSearch索引

a. 进入EC2服务，并连接EC2 proxy实例

The screenshot shows the AWS EC2 Instances page. A search bar at the top is set to 'Instances (1/2) Info'. Below it, a table lists instances. One instance, 'QAChatDeployStack/Ec2Stack/ProxyInstance', is highlighted with a red box. Its details are as follows:

- Name: QAChatDeployStack/Ec2Stack/ProxyInstance
- Instance ID: i-0fbfb5ad34387790
- Instance state: Running
- Instance type: t3.micro
- Status check: 2/2 checks passed
- Alarm status: No alarms
- Availability Zone: us-west-2a
- Public IPv4 DNS: ec2-34-214-15-111.us-west-2.amazonaws.com
- Elastic IP: 34.214.15.111
- IPv6 IPs: -
- Monitoring: disabled
- Security group: aws-cloud9-chatbot

b. 获取知识库创建Script

```
1 curl -LJO https://raw.githubusercontent.com/aws-samples/private-llm-qa-
```

```
bot/main/deploy/setup_knowledgebase.sh
```

【中国区Wordaround】如果因为网络原因，上面的脚本下载失败，可以从S3 download

```
1 aws s3 cp s3://$bucket_name/private-llm-qa-
bot/deploy/setup_knowledgebase.sh . --region $region
```

c. 执行知识库创建Script

```
1 # OpenSearch的endpoint，可以从上面的Cloudformation的output中获取
2 # Dimension 是向量模型的输出维度，由选择的模型确定
3 sh setup_knowledgebase.sh ${OpensearchEndpoint} ${Dimension}
```

2. 配置OpenSearch Dashboard (可选)

OpenSearch Dashboard主要用于获取OpenSearch索引的内部细节，是进行效果深入调优的方式。

a. 安装nginx

```
1 sudo yum update
2 sudo amazon-linux-extras install nginx1 -y
```

b. 创建nginx 配置文件, 进行反向代理

```
1 cd /etc/nginx/conf.d
2 sudo vim default.conf
```

c. 修改如下, domain_endpoint需要设置为创建的Amazon OpenSearch Endpoint(在Cloudformation获取)

```
1 # ${OpenSearch Endpoint} 需要从cloudformation的输出中获取;
2 server {
3     listen 80;
4     server_name localhost;
5
6     location / {
7         proxy_pass https://vpc-domain66ac69e0-pl89l5ymkgar-v64op6yrjkmn3phter;
```

```

8     proxy_pass https://${OpenSearch Endpoint};
9 }
10 }

```

d. 重启nginx

```

1 sudo nginx -t
2 sudo systemctl restart nginx

```

e. 访问OpenSearch Dashboard

i. 打开Dashboard网页，可以参见下图(Cloudformation Output)找到对应的网址

Key	Value	Description
APIGatewayEndpointUrl	https://acwixwmd7.execute-api.us-west-2.amazonaws.com/prod/	-
DownloadKeyCommand	aws secretsmanager get-secret-value --secret-id ec2-ssh-key/cdk-keypair/private --query SecretString --output text > cdk-key.pem && chmod 400 cdk-key.pem	-
embeddingEndpoint	9.....3-05-18-07-05-20-embedding-endpoint	-
embeddingModelName	9.....3-05-18-07-05-20-embedding	-
GlueJobName	chatbotfromstodoaf988AG33-9qjeySWm8qvz	-
KinesisFirehoseRole	QAChatDeployStack-chatbotkinesisfirehoseAADBC891-7Q5Q7TE715K	-
llmChatGLMEndpoint7-23-05-18-07-05-20-llm-chatglm-endpoint	-
modelName7-23-05-18-07-05-20-llm-chatglm	-
OpenSearchEC2ProxyAddress	http://34.2.14.15.111.8081/_dashboards/	-
opensearchEndpoint	yoc-domain66ac69e0-p18915ymkqr-v5406vrkmn9ohemmz1hwkne.us-west-2.es.amazonaws.com	-

ii. 访问页面后，可以对OpenSearch进行操作，比如查询索引的schema，如图

```

1+ {
2+   "cluster": "index": {
3+     "aliases": {},
4+     "mappings": {
5+       "properties": {
6+         "content": {
7+           "type": "text",
8+           "analyzer": "ik_max_word",
9+           "search_analyzer": "ik_smart"
10+          },
11+         "doc": {
12+           "type": "text",
13+           "analyzer": "ik_max_word",
14+           "search_analyzer": "ik_smart"
15+         },
16+         "doc_author": {
17+           "type": "keyword"
18+         },
19+         "doc_category": {
20+           "type": "keyword"
21+         },
22+         "doc_title": {
23+           "type": "keyword"
24+         },
25+         "doc_type": {
26+           "type": "keyword"
27+         }
28+       },
29+       "embedding": {
30+         "type": "knn_vector",
31+         "dimension": 1024,
32+         "method": "l2",
33+         "engine": "mlslib",
34+         "space_type": "cosinesimil",
35+         "max_top_k": 100,
36+         "parameters": {
37+           "ef_construction": 128,
38+           "m": 16
39+         }
40+       }
41+     }
42+   }
43+ }

```

3.4 串联整个流程

3.4.1 模型部署

i. 模型部署环境

A. [Global Region] 图示步骤

B. [China Region] 图示步骤

ii. 模型部署步骤

A. LLM模型部署

使用这个[目录](#)内notebook进行部署。如果使用Bedrock，或者其他SAAS化的LLM API，这步可跳过。默认选择[baichuan2_13b_gptq_deploy.ipynb](#)进行部署

B. Embedding模型部署

使用这个[目录](#)内notebook进行部署，默认选择[bge_zh_deploy.ipynb](#)进行部署

3.4.2 设置Lambda函数的环境变量

共有五个Lambda的环境变量需要设置为部署的模型Endpoint, 分别是:

- Ask_Assistant
- Trigger_Ingestion
- Detect_Intention
- Query_Rewrite
- Chat_Agent

具体请参考下面两个图, 环境变量名称为embedding_endpoint和llm_model_endpoint。

The screenshot shows the 'Endpoints' section in the SageMaker Studio interface. It lists two endpoints: '370400458231-23-05-24-08-34-42-llm-chaining-endpoint' and '370400458231-23-05-24-08-34-42-embedding-endpoint'. Both endpoints have their ARNs displayed next to them.

SageMaker中部署的模型Endpoint

The screenshot shows the 'Configuration' tab for a Lambda function. The 'Environment variables' section is highlighted. It lists several variables, with 'embedding_endpoint' and 'llm_model_endpoint' being the ones specifically highlighted with red boxes around their values.

Variable	Value
embedding_endpoint	bge-zh-15-2023-09-17-01-00-27-06-endpoint
llm_model_endpoint	106039800180-23-10-09-15-08-15-llm-default-endpoint

Lambda Ask_Assistant 环境变量设置

3.5 日志回流检索

参考[workshop](#)部分章节。

3.6 安全防范措施

3.6.1 删掉OpenSearch Proxy Ec2的安全组 80端口

入站规则 (5)										
	Name	安全组规则 ID	IP 版本	类型	协议	端口范围	源	描述		
<input type="checkbox"/>	-	sgr-028041fd4bdeb04...	IPv4	HTTPS	TCP	443	10.22.0.0/16	from 10.22.0.0/16:443		
<input type="checkbox"/>	-	sgr-067b9092260d45...	IPv4	SSH	TCP	22	0.0.0.0/0	Allow SSH Access		
<input type="checkbox"/>	-	sgr-08c798a3119000...	IPv4	自定义 TCP	TCP	8081	0.0.0.0/0	Allow HTTP 8081 port Access		
<input type="checkbox"/>	-	sgr-0a715b0a2c4c2fc08	-	所有流量	全部	全部	sg-0e41cef06e0322cb...	Allow self traffic		
<input type="checkbox"/>	-	sgr-0940c7f1fe535d9496	IPv4	HTTP	TCP	80	0.0.0.0/0	Allow HTTP Access		

这个80端口主要是用于从外网登陆OpenSearch Dashboard，用于定制化修改对应的知识库 Schema信息。一般情况下无需打开，如果有访问需要，可以限制源为指定的外网IP。

4. 部署说明-前端部分（可选）

4.1 下载代码

如果是中国区，因为网络原因导致下载失败，可以先下载到本地再通过S3 Bucket进行中转，参考前面的3.1章节

```
1 git clone https://github.com/xiehust/chatbotFE.git
2 cd chatbotFE/deploy
```

进入chatbotFE/deploy目录后，修改env.sample。只需修改以下4项，根据自己的账号修改：

其中 **MAIN_FUN_ARN** 为后端部署完成之后主lambda的arn，可以在lambda控制台获取，也可以按以下格式拼写。**UPLOAD_BUCKET**是部署后端时定义的文档存放S3桶名

例如：

 **CDK_DEFAULT_ACCOUNT**=6310xxxxx15

CDK_DEFAULT_REGION=cn-northwest-1

如果是中国区部署：

MAIN_FUN_ARN=arn:**aws-cn**:lambda:{region}:{aws account id}:function:Ask_Assistant

非中国区域：

MAIN_FUN_ARN=arn:**aws**:lambda:{region}:{aws account id}:function:Ask_Assistant

UPLOAD_BUCKET=是部署后端时定义的文档存放S3桶名

```
CDK_DEFAULT_ACCOUNT=631023...  
CDK_DEFAULT_REGION=us-...  
TOKEN_KEY=bot_chat_test_20230502  
OPENAI_API_KEY=  
START_CMD=/rs  
UPLOAD_BUCKET=6310...atbot-bucket  
UPLOAD_OBJ_PREFIX=ai-content/  
MAIN_FUN_ARN=arn:aws:lambda:us-...-2:6310...15:function:Ask_Assistant  
embedding_endpoint=  
sd_endpoint_name=  
all_in_one_api=
```

再另存为.env 文件

```
1 cp env.sample .env
```

4.2 运行CDK

1. 仍然在chatbotFE/deploy目录中，执行

```
1 npm install  
2 cdk synth  
3 cdk deploy --require-approval never
```

2. 执行成功之后，会在账号中部署2个api gateway，其中https开始的是rest apigateway，wss开始的是websocket apigateway.

```
✨ Deployment time: 101.66s  
Outputs:  
BackendCdkStack.APIGatewayendpointurl = https://g1w...execute-api.cn-northwest-1.amazonaws.com.cn/prod/  
BackendCdkStack.ChatBotWsApiURL = wss://1df...cn-northwest-1.amazonaws.com.cn/Prod
```

3. 回到上一级目录，在chatbotFE目录下，修改env.sample文件

```
1 cd ..
```

📌 **REACT_APP_API_http**=上一步的中输出的https开头的地址

REACT_APP_API_socket=上一步的中输出的wss开头的地址

REACT_APP_DEFAULT_UPLOAD_BUCKET=是部署后端时定义的文档存放S3桶名

```
app > chatbotFE >  .env
REACT_APP_API_http=https://dv[REDACTED].execute-api.us-west-2.amazonaws.com/prod
REACT_APP_API_socket=wss://gje7[REDACTED].execute-api.us-west-2.amazonaws.com/Prod
REACT_APP_DEFAULT_UPLOAD_BUCKET=631023[REDACTED]-chatbot-bucket
```

4. 另存为.env 文件

```
1 cp env.sample .env
```

5. Build 前端js文件，在目录chatbotFE/下运行

```
1 yarn install
2 yarn build
```

4.3 部署hosting

前端js code可以托管到S3，也可以继续在此EC2上部署。

如果托管S3：

需要开启S3的static website hosting, 设置 index document, permissions. 可以参考更多 [AWS doc Reference](#).

1. Create an S3 bucket named **bucket-name** on the Amazon S3 console.
2. Enable static website hosting of this bucket. In Index document, enter the file name of the index document `index.html`.
3. By default, the S3 bucket blocks public access. You need to change the setting by unchecking the option in the "Permissions" tab of the bucket detail page.
4. Add the policy below to the bucket policy to allow public access.

```
1 {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Sid": "PublicReadGetObject",
6             "Effect": "Allow",
7             "Principal": "*",
8             "Action": [
9                 "s3:GetObject"
10            ],
11            "Resource": [
```

```
12           "arn:aws:s3:::bucket-name/*"
13       ]
14   }
15 ]
16 }
```

5. 把chatbotFE/build 目录的内容上传到s3

```
1 cd chatbotFE
2 aws s3 sync ./build/ s3://bucket-name/
```

最后可以通过以下两种url格式访问：

- <http://bucket-name.s3-website-Region.amazonaws.com>
- <http://bucket-name.s3-website.Region.amazonaws.com>

如果在EC2上部署：

1. 安装pm2工具

在chatbotFE目录中，启动pm2

```
1 sudo yarn global add pm2
2 cd chatbotFE
3 pm2 start yarn --name "chatbotFE" -- start
```

查看后台进程是否加入了运行

```
1 pm2 list
```

id	name	namespace	version	mode	pid	uptime	σ	status	cpu	mem	user	watching
0	chatbotFE	default	N/A	fork	32277	82s	0	online	0%	77.6mb	ec2-user	disabled

2. 设置默认启动

```
1 pm2 startup systemd
```

3. 设置ALB

a. 在EC2控制台的负载平衡中，创建一个目标组，并选择之前的ec2实例

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol	Port
HTTP	: 80 1-65535

HTTPS

b. 注册目标端口为3000

EC2 > Target groups > chatbotfe > Register targets

Register targets

Select instances, specify ports, and add the instances to the list of pending targets. Repeat to add additional combinations of instances and ports to the list of pending targets. Once you are satisfied with your selections, click Register pending targets.

Available instances (1/1)						
Instance ID	Name	Status	Security groups	Zone	Public IPv4 address	Subnet ID
<input checked="" type="checkbox"/> i-04e3a834d78983141	chatbotFServer	Running	launch-wizard-1	us-west-2c	54.149.207.153	subnet-084270d413271ff39

1 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

1-65535 (separate multiple ports with commas)

Review targets

c. 创建一个application load balancer

- 📌 1. ALB的VPC选择跟EC2在同一个VPC，安全组打开80端口公开访问。
- 2. 再配置原来EC2的安全规则，使得ALB的安全组可以访问EC2安全组的所有流量同一个安全组



d. ALB的目标组是上一步创建的目标组。

A screenshot of the AWS Application Load Balancer Listener configuration page. The main title is 'Listeners and routing Info'. A sub-section titled '▼ Listener HTTP:80' is expanded, showing the following details:

Protocol	Port	Default action	Action
HTTP	: 80 1-65535	Forward to chatbotfe Target type: Instance, IPv4	HTTP <input checked="" type="button"/>

Below the listener configuration, there is a section for 'Listener tags - optional' with a note: 'Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.' It includes a 'Add listener tag' button and a note: 'You can add up to 50 more tags.'

At the bottom left, there is a 'Add listener' button.

e. ALB的dns，即是访问chatbot的地址

4.4 配置管理员账户

1. 去dynamodb控制台，找到usertable.

DynamoDB > 浏览项目 > BackendCdkStack-usertable46247387-N1PB6ZO2JYQ1

表 (3) >

任何标签键

任何标签值

按表名称查找表

BackendCdkStack-usertable46247387-N1PB6ZO2JYQ1

chat_user_info

http-crud-tutorial-items

已完成. 已使用的读取容量单位: 0.5

返回的项目 (0)

无项目

没有要显示的项目。

操作 > 创建项目

2. 点击创建项目，添加以下6个字段。

username:yourname
password:yourpassword
email:youremail@xx.com
groupname:admin
status:active
createtime:2023-06-26T16:30:37.120Z

DynamoDB > Explore items: BackendCdkStack-usertable46247387-1EM3CNWDPKDKN > Create item

Create item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#)

Attributes		Add new attribute ▾
Attribute name	Value	Type
username - Partition key	admin	String
password	-	String
email	admin@aws.com	String
groupname	admin	String
status	active	String
createtime	2023-06-26T16:30:37.120Z	String

Cancel **Create item**

访问ALB的dns或者S3托管的endpoint，进入login登陆页面。



Sign in

Username *

Password *

Remember me

SIGN IN

SIGN UP

[Forgot password?](#)

4.5 使用流式输出

前端部署的websocket apigateway用于流式输出，这时还需要后端有权限对这个apigateway发送消息。

因此需要把后端部署的CDK wss接口参数更新一下，再执行一下后端的CDK部署代码，对wss接口授权。

1. 修改private-llm-qa-bot/deploy/.env文件

```
1 cd private-llm-qa-bot/deploy/  
2 vim .env
```

2. 加入以下3个变量：

📌 **use_wss=1**

wss_apiId={apigateway id}

如果是中国区：

wss_resourceArn=arn:aws-cn:execute-api:{region}:{aws account id}:{apigateway id}/*//@connections/

非中国区：

wss_resourceArn=arn:aws:execute-api:{region}:{aws account id}:{apigateway id}/*//@connections/

例如：

```
use_wss=1  
wss_apiId=gje7lsl  
wss_resourceArn=arn:aws:execute-api:us-west-2:631023274615:gje: /csl/*/*@connections/*
```

4. 修改保存后，再在private-llm-qa-bot/deploy 目录下执行，进行更新部署。更新部署过程加快，大约几分钟就完成

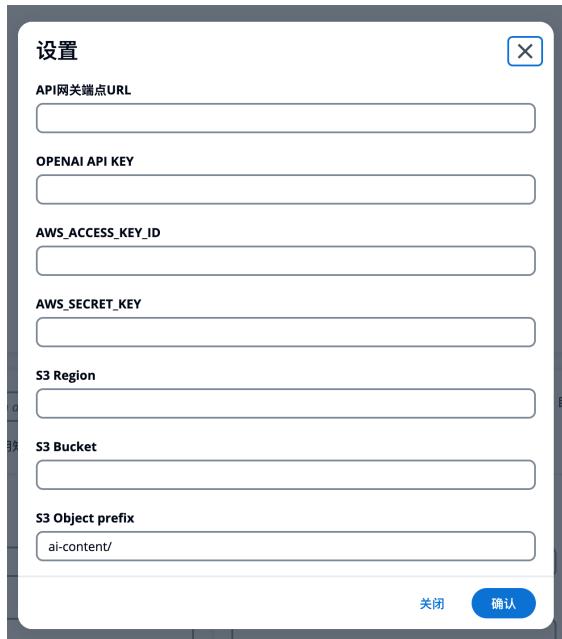
```
1 cdk synth  
2 cdk deploy --require-approval never
```

5. 功能介绍说明

5.1 基本功能介绍

The screenshot shows the AWS Chat Portal interface. At the top, there's a navigation bar with the AWS logo, 'AWS Chat Portal' title, a user icon (admin@admin), and a settings gear icon. Below the navigation is a sidebar with '聊天区' (Chats) and '管理' (Management) sections. Under '管理', there are links for '文档库' (Document Library), '提示词模板' (Prompt Template), '反馈管理' (Feedback Management), 'Few shot示例管理' (Few-shot Examples Management), and '用户' (Users). A red box highlights the '文档库' link with the number '5'. The main content area is titled 'AWS智能问答' (AI Assistant) and has a sub-section '对话' (Conversation). It features a search bar with placeholder 'Input your question and press enter', a 'Stream' button (highlighted with a red box, labeled '3'), and other options like '使用知识库回答' (Answer from Knowledge Base), '多轮会话' (Multi-turn Conversation), '隐藏引用' (Hide References), and '跟踪日志' (Track Log). Below these are '更多设置' (More Settings) options: 'LLM模型' (Model) set to 'claude', '最大Token数量' (Max Tokens) set to '3000', 'Temperature' set to '0.01', '系统角色名' (System Role Name) empty, '系统角色提示词' (System Role Prompt) empty, and '提示词模板' (Prompt Template) set to 'sso-chatbot-1102'. A red box highlights the '更多设置' section with the number '4'. At the bottom right are '发送' (Send) and '新对话' (New Conversation) buttons.

1. 语言切换，支持中英文切换
2. 系统设置，默认不用填。如果要调用别的账户的后端，则需要配置其对应的API网关端点URL,AWS_ACCESS_KEY_ID,AWS_SECRET_KEY,S3 Region,S3 Bucket



3. 控制开启流式输出，是否启用知识库，多轮会话，显示引用文档，输出追踪日志功能。
4. 展开更多设置，可以选择模型，提示词模板，模型参数等。

5.2 上传知识

管理->文档库，点击创建，支持txt, pdf等类型，也可以下载csv模板上传faq类型知识。

也选中之后删除知识。

5.3 管理提示词模板

管理->提示词模板，可以创建，删除，修改提示词模板，注意例子中{}的是保留字段，不能自行修改，具体解释可以见[说明](#)



添加模板

聊天区

AWS智能问答

管理

文档库

提示词模板

反馈管理

Few shot示例管理

用户

模板名称

Required

模板

Keywords: {system_role_prompt},{question},{role_bot},{chat_history},{context} [使用说明](#)

```

1 {system_role_prompt} {role_bot}, 请根据反括号中的资料提取相关信息, 回答用户的各种
    问题
2 ``
3 {chat_history}
4 {context}
5 ``
6 用户:{question}
7 ▼ {role_bot}: 
```

Python

undefined: 0

△ undefined: 0



预览

, 请根据反括号中的资料提取相关信息, 回答用户的各种问题
``

{chat_history}
{context}
``

用户:{question}

备注

Optional

取消

确认

5.4 反馈管理

用户可以对chatbot回答进行反馈，纠错。

对话



ODCR预留实例怎么计费?

AN

只要交付到客户账户里, 无论是否预留实例是否运行, 都按等同的按需费率计费。如果客户没有使用预留, 将在客户的EC2账单中显示为未使用的预留。如果客户运行的实例属性与预留匹配, 则客户只需要为该实例付费, 不需要为预留付费。

纠正

具体的反馈在**反馈管理**中能够查看, 管理员角色用户, 可以双击单元格对反馈内容进行修改, 或者点击注入知识库。这样将会把这条内容直接注入到知识库中, 形成新的知识。生效时间大约1分钟。

AWS Chat Portal > 反馈管理

反馈管理 (1+)

Filter by text, property or value

Question	Original Answer	New Answer	Status	Timestamp
erik chen 是负责什么的	根据公开报道,Erik Chen是一位年轻的...	事实上 ✓ ✗	injected	2023-11-05 11:40:...

也可以用于逐条创建FAQ类型的知识

AWS Chat Portal > 反馈管理

反馈管理 (1+)

Filter by text, property or value

Question	Original Answer	New Answer	Status	Timestamp
erik chen 是负责什么的	根据公开报道,Erik Chen是一位年轻的...	事实上 ✓ ✗	injected	2023-11-05 11:40:...

创建新的FAQ

问题
Your question

答案
Your answer

取消 提交

5.5 Fewshot 示例管理

跟文档库功能类似，上传的是意图识别的示例，具体模板可以参考[这里的文件](#)

AWS Chat Portal

Few shot示例管理 (5)

Find

索引名	Embedding模型	文档名称	创建者
chatbot-example-index	paraphrase-m-base-v2-2023-08-31-05-2...	ai-content/aws_service_owner.example	s3event
chatbot-example-index	paraphrase-m-base-v2-2023-08-31-05-2...	ai-content/emotion.example	s3event
chatbot-example-index	paraphrase-m-base-v2-2023-08-31-05-2...	ai-content/aws_service_status.example	s3event
chatbot-example-index	paraphrase-m-base-v2-2023-08-31-05-2...	ai-content/conversations.example	s3event
chatbot-example-index	paraphrase-m-base-v2-2023-08-31-05-2...	ai-content/admin/aws_faq.example	admin

5.6 用户管理

管理员可以在这里添加其他账户

首页 > 用户 > 添加用户

添加用户

用户名

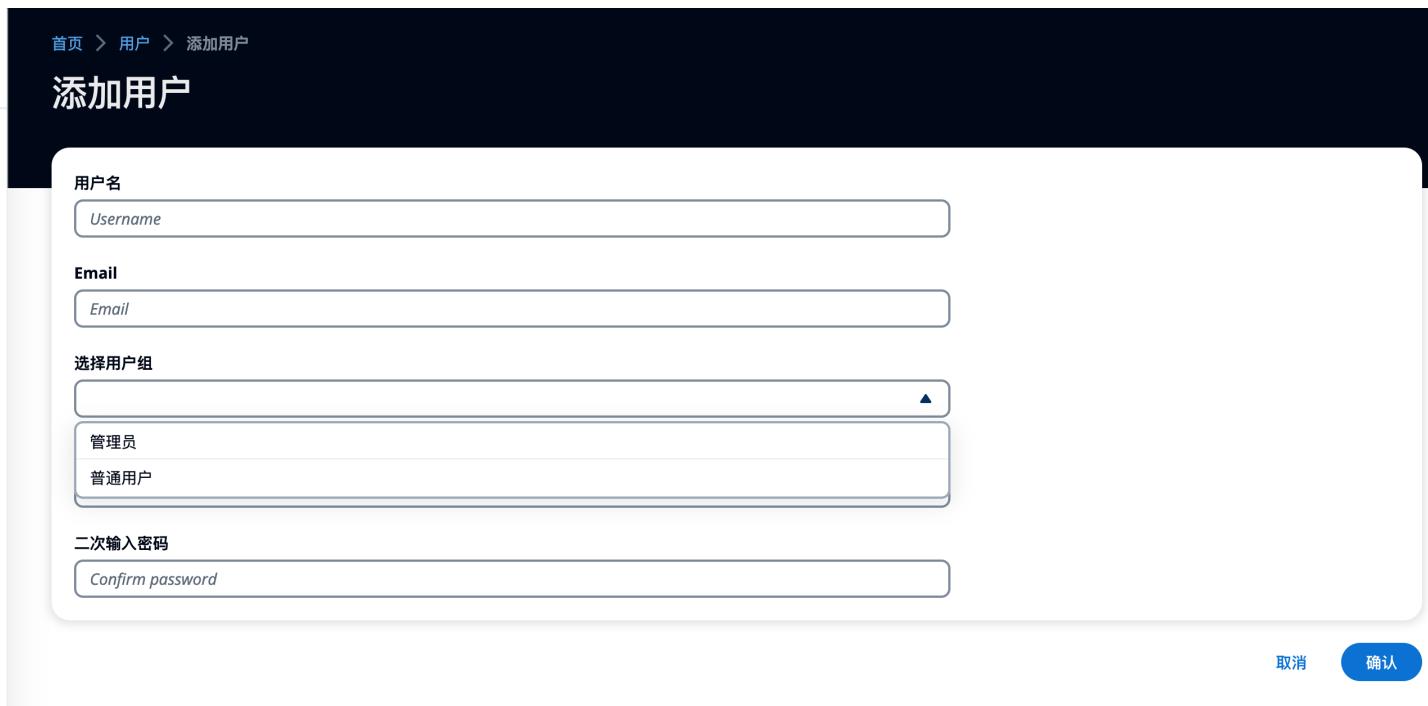
Email

选择用户组

管理员
普通用户

二次输入密码

取消 确认

A screenshot of a user addition form. At the top, there's a breadcrumb navigation: 首页 > 用户 > 添加用户. Below it is a title 添 加用 户. The form consists of several input fields: '用户名' (Username) with placeholder 'Username', 'Email' with placeholder 'Email', and a '选择用户组' (Select User Group) dropdown menu containing '管理员' (Administrator) and '普通用户' (Normal User). There's also a '二次输入密码' (Confirm Password) field with placeholder 'Confirm password'. At the bottom right are two buttons: '取消' (Cancel) and '确认' (Confirm).

6. Demo视频

https://www.bilibili.com/video/BV1HN4y1D7vy/?vd_source=2cb87d8dd3ca4ea778f5468be12405b3