------------------ 环境搭建过程 -----------------

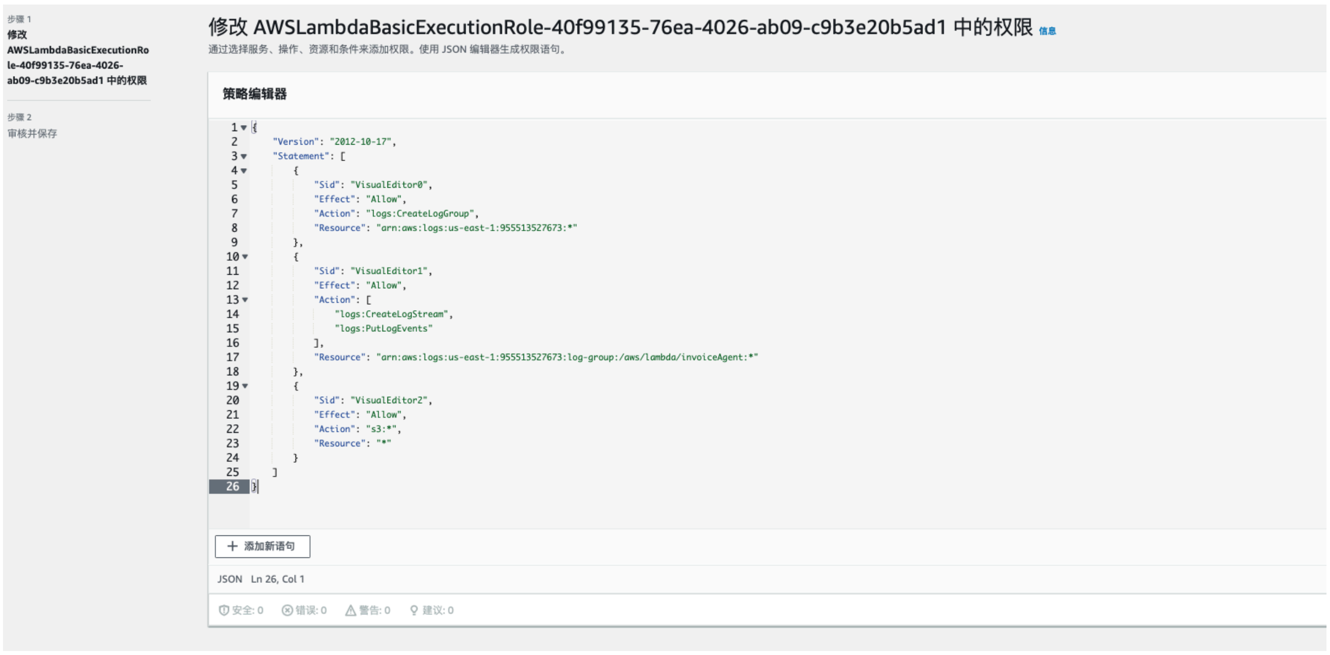
0. 金蝶测试特有的环境准备

1. 创建存储桶，上传product_code_name_map.txt，product_38_code_name_map.txt，product_47_to_38_map.txt

在后面创建的Lambda函数中加入环境变量BUCKET_NAME

2. Lambda的权限采用了默认，只有对cloudwatch的上传权限。金蝶案例中需要操作S3，增加对S3的操作权限：
   方法：创建了Lambda之后，在Lambda-配置-权限处对应的角色名称进入，增加对S3的操作权限

   ```
   {
           "Sid": "VisualEditor2",
           "Effect": "Allow",
           "Action": "s3:*",
           "Resource": "*"
       }
   ```



步骤 1
修改 AWSLambdaBasicExecutionRole-40f99135-76ea-4026-ab09-c9b3e20b5ad1 中的权限

步骤 2
审核并保存

修改 AWSLambdaBasicExecutionRole-40f99135-76ea-4026-ab09-c9b3e20b5ad1 中的权限 信息

通过选择服务、操作、资源和条件来添加权限。使用 JSON 编辑器生成权限语句。

策略编辑器

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:us-east-1:955513527673:*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:us-east-1:955513527673:log-group:/aws/lambda/invoiceAgent:*"
        },
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*"
        }
    ]
}
```

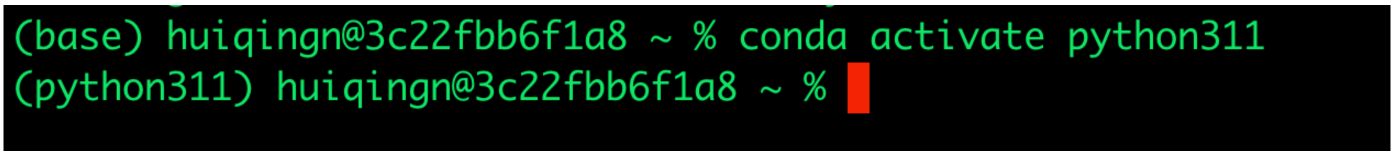
1. 使用Lambda创建Tools

1.1 构建Lambda所需的依赖包

这里体现了使用Bedrock agent的优势。Bedrock agent负责对LLM和调用流程的处理，因此构建Lambda依赖包非常简单。只需要添加构建function时需要的依赖即可。而如果使用Langchain或openAI function calling自行构建agent时，需要考虑Langchain的版本，需要安装。以Langchain为例，开源的版本变化非常快，需要考虑版本的更新、Langchain版本使用的LLM版本的匹配性问题，都给我们后续的开发维护带来难度；而Bedrock agent则通过AWS平台负责维护，不需要客户负责。

如下是开票业务所需要的依赖库。其他场景根据需要选择依赖库打包。

注意构建依赖包时使用与Lambda runtime相同的环境版本（Linux x86-64 ）。下面示例中使用python3.11进行环境构建

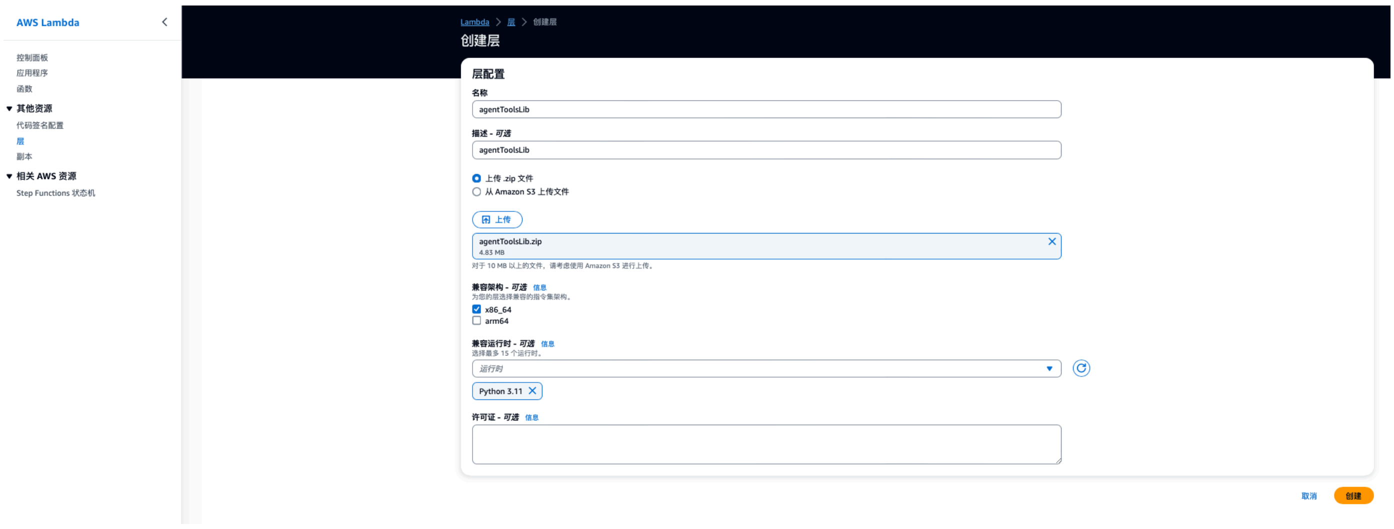


```
(base) huiqingn@3c22fbb6f1a8 ~ % conda activate python311
(python311) huiqingn@3c22fbb6f1a8 ~ %
```

```
(python311) huiqingn@3c22fbb6f1a8 ~ % mkdir agentToolsLib
(python311) huiqingn@3c22fbb6f1a8 ~ % cd agentToolsLib
(python311) huiqingn@3c22fbb6f1a8 agentToolsLib %pip install cryptography -t ./
(python311) huiqingn@3c22fbb6f1a8 agentToolsLib %pip install requests -t ./
(python311) huiqingn@3c22fbb6f1a8 agentToolsLib %cd ..
(python311) huiqingn@3c22fbb6f1a8 ~ %zip -r agentToolsLib.zip ./agentToolsLib
```

1.2 构建Layer

使用构建的依赖包创建Lambda的Layer。



创建层

层配置

名称: agentToolsLib

描述 - 可选: agentToolsLib

[✓ 上传 .zip 文件] 从 Amazon S3 上传文件

agentToolsLib.zip 4.83 MB

对于 10 MB 以上的文件，请考虑使用 Amazon S3 进行上传。

兼容架构 - 可选 信息

为您的层选择兼容的指令集架构。

[✓ x86_64] arm64

兼容运行时 - 可选 信息

选择最多 15 个运行时。

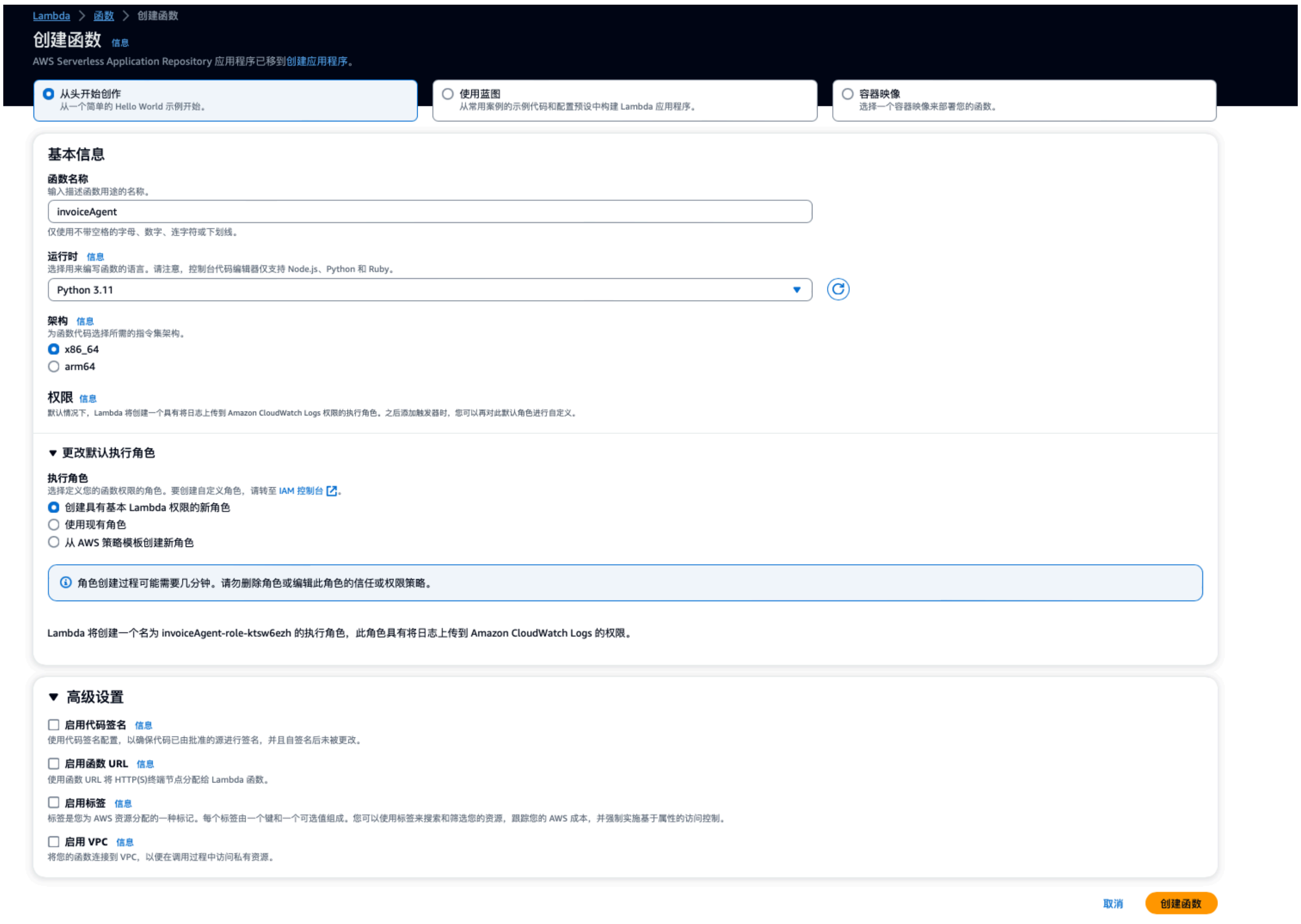
运行时: Python 3.11

许可证 - 可选 信息

1.3 创建Lambda函数

1.3.1 创建Lambda

- 创建Lambda函数，指定所需要的runtime，基本权限



Lambda > 函数 > 创建函数

创建函数 信息

AWS Serverless Application Repository 应用程序已移到创建应用程序。

[✓ 从头开始创作] 从一个简单的 Hello World 示例开始。
使用蓝图 从常用案例的示例代码和配置预设中构建 Lambda 应用程序。
容器映像 选择一个容器映像来部署您的函数。

基本信息

函数名称
输入描述函数用途的名称。
invoiceAgent
仅使用不带空格的字母、数字、连字符或下划线。

运行时 信息
选择用来编写函数的语言。请注意，控制台代码编辑器仅支持 Node.js、Python 和 Ruby。
Python 3.11

架构 信息
为函数代码选择所需的指令集架构。
[✓ x86_64]
arm64

权限 信息
默认情况下，Lambda 将创建一个具有将日志上传到 Amazon CloudWatch Logs 权限的执行角色。之后添加触发器时，您可以再对此默认角色进行自定义。

▼ 更改默认执行角色

执行角色
选择定义您的函数权限的角色。要创建自定义角色，请转至 IAM 控制台。
[✓ 创建具有基本 Lambda 权限的新角色]
使用现有角色
从 AWS 策略模板创建新角色

角色创建过程可能需要几分钟。请勿删除角色或编辑此角色的信任或权限策略。

Lambda 将创建一个名为 invoiceAgent-role-ktsw6ezh 的执行角色，此角色具有将日志上传到 Amazon CloudWatch Logs 的权限。

▼ 高级设置

☐ 启用代码签名 信息
使用代码签名配置，以确保代码已由批准的源进行签名，并且自签名后未被更改。

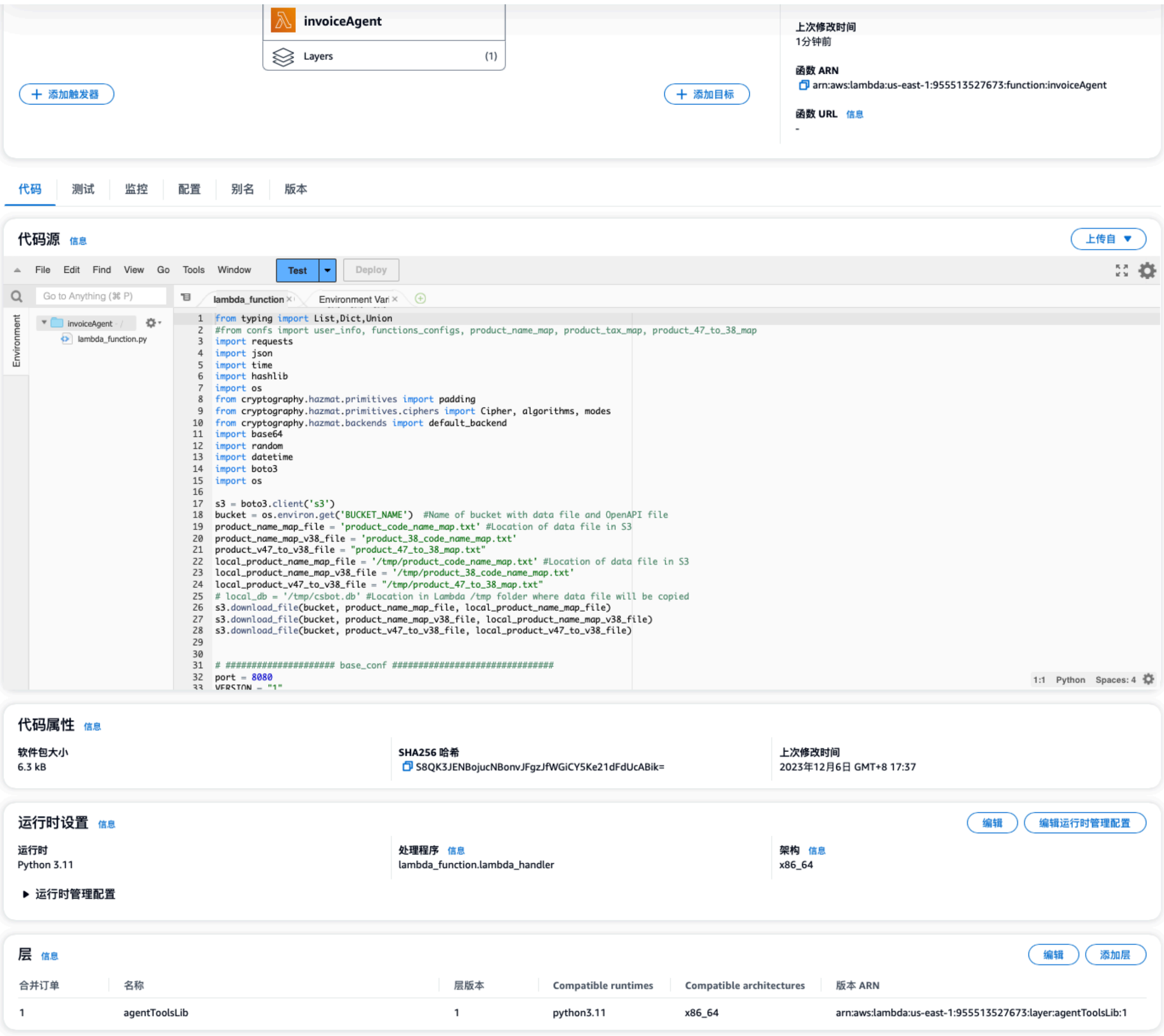
☐ 启用函数 URL 信息
使用函数 URL 将 HTTP(S)终端节点分配给 Lambda 函数。

☐ 启用标签 信息
标签是您为 AWS 资源分配的一种标记。每个标签由一个键和一个可选值组成。您可以使用标签来搜索和筛选您的资源，跟踪您的 AWS 成本，并强制实施基于属性的访问控制。

☐ 启用 VPC 信息
将您的函数连接到 VPC，以便在调用过程中访问私有资源。

对该Lambda函数添加Layer

点击图标的Layer，加入前面创建的依赖Layer



上次修改时间
1分钟前

函数 ARN
arn:aws:lambda:us-east-1:955513527673:function:invoiceAgent

函数 URL 信息
-

代码源 信息

```python
from typing import List,Dict,Union
#from confs import user_info, functions_configs, product_name_map, product_tax_map, product_47_to_38_map
import requests
import json
import time
import hashlib
import os
from cryptography.hazmat.primitives import padding
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.backends import default_backend
import base64
import random
import datetime
import boto3
import os

s3 = boto3.client('s3')
bucket = os.environ.get('BUCKET_NAME')  #Name of bucket with data file and OpenAPI file
product_name_map_file = 'product_code_name_map.txt' #Location of data file in S3
product_name_map_v38_file = 'product_38_code_name_map.txt'
product_v47_to_v38_file = "product_47_to_38_map.txt"
local_product_name_map_file = '/tmp/product_code_name_map.txt' #Location of data file in S3
local_product_name_map_v38_file = '/tmp/product_38_code_name_map.txt'
local_product_v47_to_v38_file = "/tmp/product_47_to_38_map.txt"
# local_db = '/tmp/csbot.db' #Location in Lambda /tmp folder where data file will be copied
s3.download_file(bucket, product_name_map_file, local_product_name_map_file)
s3.download_file(bucket, product_name_map_v38_file, local_product_name_map_v38_file)
s3.download_file(bucket, product_v47_to_v38_file, local_product_v47_to_v38_file)


# ###################### base_conf ###############################
port = 8080
VERSION = "1"
```

1:1 Python Spaces: 4

代码属性 信息

软件包大小	SHA256 哈希	上次修改时间
6.3 kB	S8QK3JENBojucNBonvJFgzJfWGiCY5Ke21dFdUcABik=	2023年12月6日 GMT+8 17:37

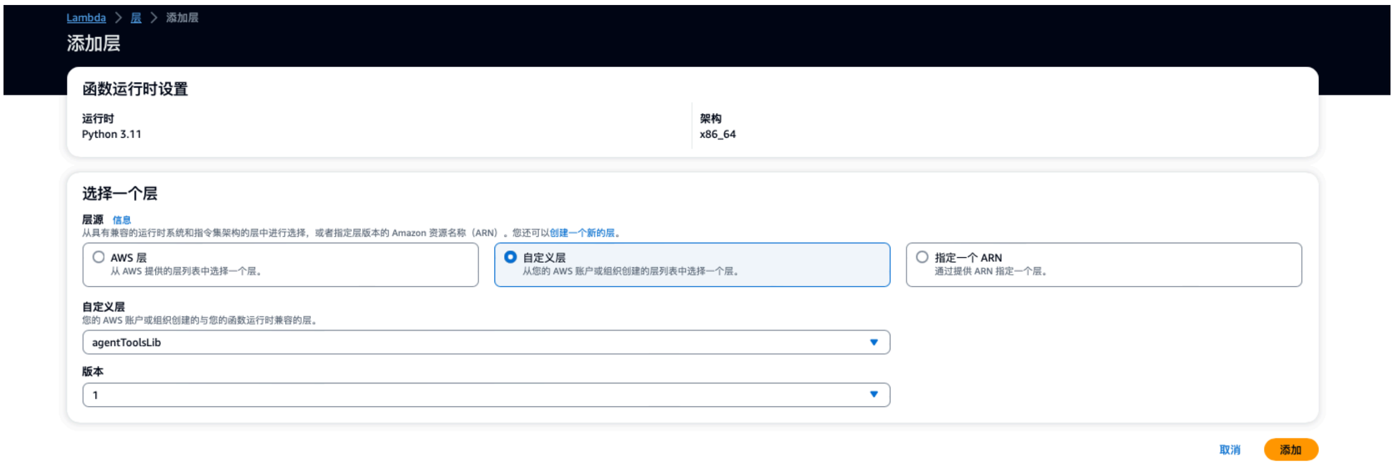
运行时设置 信息

运行时	处理程序 信息	架构 信息
Python 3.11	lambda_function.lambda_handler	x86_64

▶ 运行时管理配置

层 信息

合并订单	名称	层版本	Compatible runtimes	Compatible architectures	版本 ARN
1	agentToolsLib	1	python3.11	x86_64	arn:aws:lambda:us-east-1:955513527673:layer:agentToolsLib:1



添加层

函数运行时设置

运行时	架构
Python 3.11	x86_64

选择一个层

层源 信息
从具有兼容的运行时系统和指令集架构的层中进行选择，或者指定层版本的 Amazon 资源名称（ARN）。您还可以创建一个新的层。

- ○ AWS 层 — 从 AWS 提供的层列表中选择一个层。
- ● 自定义层 — 从您的 AWS 账户或组织创建的层列表中选择一个层。
- ○ 指定一个 ARN — 通过提供 ARN 指定一个层。

自定义层
您的 AWS 账户或组织创建的与您的函数运行时兼容的层。
agentToolsLib

版本
1

1.3.2 Lambda函数Event内容解析

LLM在判断需要使用Tools时会触发Lambda，产生的event示例如下。从如下的event可以看到，判断使用generatePreviewInvoiceImage 函数，以及对应的参数内容。 详情请参考官方文档：参考https://docs.aws.amazon.com/zh_cn/bedrock/latest/userguide/agents-create.html

```
{
    "agent": {
        "alias": "TSTALIASID",
        "name": "KingdeeAgent",
        "version": "DRAFT",
        "id": "NORJI2BS5I"
    },
    "sessionId": "955513527673393",
    "inputText": "公司名称：金蝶有礼，识别号：91440300MA5FAE9E4P，用户ID：300，产品信息： 小麦，1010101020000000000，9000",
    "sessionAttributes": {},
    "promptSessionAttributes": {},
    "apiPath": "/generatePreviewInvoiceImage",
    "httpMethod": "POST",
    "messageVersion": "1.0",
    "actionGroup": "issueInvoice",
    "parameters": [
        {
            "name": "buyer_company_name",
            "type": "string",
            "value": "金蝶有礼"
        },
        {
            "name": "product_detail",
            "type": "array",
            "value": "[\"{\"name\":\"小麦\",\"code\":\"1010101020000000000\",\"money\":\"9000\"}\"]"
        },
        {
            "name": "user_id",
            "type": "string",
            "value": "300"
        },
        {
            "name": "buyer_tax_number",
            "type": "string",
            "value": "91440300MA5FAE9E4P"
        }
    ]
}
```

1.3.3 Lambda函数编写

Lambda函数编写分两部分。第一部分为客户根据业务需求做相应业务函数的编写。如下面sendInvoiceEmail、generatePreviewInvoiceImage、issueInvoice等函数编写。 第二部分为Lambda入口函数编写，根据event中携带的api_path调用相应的业务函数。

Lambda函数的返回值格式，请参考如下示例并结合官方文档https://docs.aws.amazon.com/zh_cn/bedrock/latest/userguide/agents-create.html

```
def get_named_parameter(event, name):
    return next(item for item in event['parameters'] if item['name'] == name)['value']

def sendInvoiceEmail(event):
    invoice_code = get_named_parameter(event, 'invoice_code')
    invoice_number = get_named_parameter(event, 'invoice_number')
    email_address = get_named_parameter(event, 'email_address')
    ## 处理逻辑省略 ##
    ## 调用后端API
    encrypt_key = functions_configs[function_name]["encrypt_key"]
    data = {
        "invoiceCode":invoice_code,
        "invoiceNo":invoice_number,
        "email":email_address
    }
    # 数据加密
    base64_cipher = encrypt(encrypt_key=encrypt_key, data=data)
    response = requests.post(
        url=url,
        data=json.dumps(base64_cipher),
        headers=header,
        params=params
    )

    result = response.json()
    #定义输出
    res = {}
    res["input_args"] = {}
    res["input_args"]["invoice_code"] = invoice_code
    res["input_args"]["invoice_number"] = invoice_number
    res["input_args"]["email_address"] = email_address
    if result["errcode"] == "0000":
        res["status"] = "success"
        res["results"] = "邮件发送成功"
    else:
        res["status"] = "fail"
        res["results"] = "邮件发送失败,请稍后尝试重新发送."
    return res

def generatePreviewInvoiceImage(event):
```

```python
  ## 处理逻辑省略 ##
def issueInvoice(event):
  ## 处理逻辑省略 ##
def lambda_handler(event, context):

    result = ''
    response_code = 200
    action_group = event['actionGroup']
    api_path = event['apiPath']

    print ("lambda_handler == > api_path: ",api_path)

    if api_path == '/generatePreviewInvoiceImage':
        result = generatePreviewInvoiceImage(event)
    elif api_path == '/issueInvoice':
        result = issueInvoice(event)
    elif api_path == '/sendInvoiceEmail':
        result = sendInvoiceEmail(event)
    else:
        response_code = 404
        result = f"Unrecognized api path: {action_group}::{api_path}"

    response_body = {
        'application/json': {
            'body': json.dumps(result)
        }
    }

    session_attributes = event['sessionAttributes']
    prompt_session_attributes = event['promptSessionAttributes']

    print ("Event:", event)
    action_response = {
        'actionGroup': event['actionGroup'],
        'apiPath': event['apiPath'],
        # 'httpMethod': event['HTTPMETHOD'],
        'httpMethod': event['httpMethod'],
        'httpStatusCode': response_code,
        'responseBody': response_body,
        'sessionAttributes': session_attributes,
        'promptSessionAttributes': prompt_session_attributes
    }

    api_response = {'messageVersion': '1.0', 'response': action_response}

    return api_response
```

2. Bedrock Agent的构建

2.1 创建function对应的schema描述

创建agent的action group时，需要提供该action group对应的API schema。Bedrock agent提供了两种方式进行开发：

1. 编写自己的json描述，上传S3
2. 提供了in-line的OpenAPI editor，可以直接在线编写

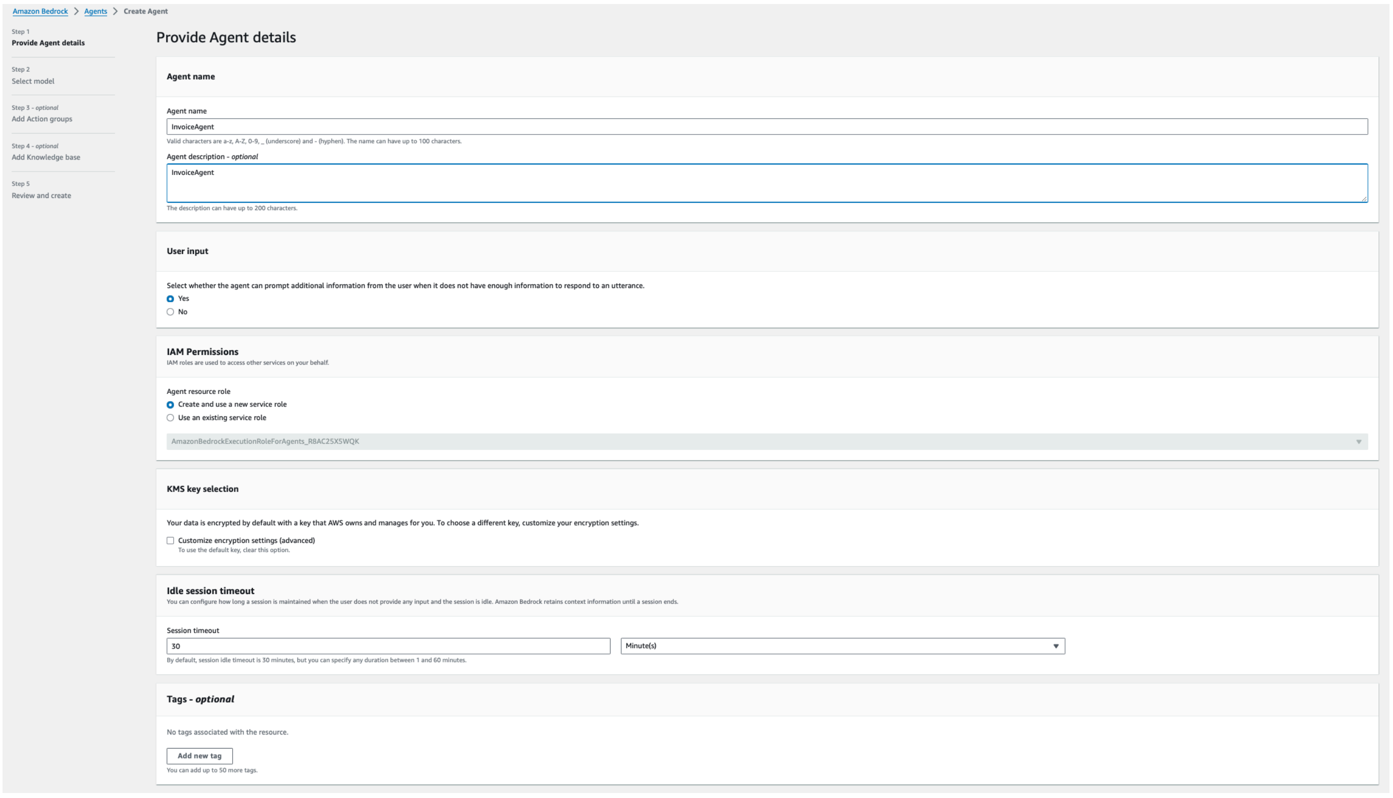
   具体要求参见官方文档https://docs.aws.amazon.com/zh_cn/bedrock/latest/userguide/agents-create.html。其中给出了当前版本的示例https://github.com/OAI/OpenAPI-Specification/tree/main/examples/v3.0

   下面给出了例子程序中使用第一种方式的json文件示例片段

```
{
    "openapi": "3.0.0",
    "info": {
      "title": "InvoiceService API",
      "description": "APIs for invoice service",
      "version": "1.0.0"
    },
    "paths": {
      "/generatePreviewInvoiceImage": {
        "post": {
          "description": "Generate a temporary preview invoice image.",
          "operationId": "generatePreviewInvoiceImage",
          "parameters": [
            {
              "name": "user_id",
              "in": "query",
              "description": "Id of user",
              "required": true,
              "schema": {
                "type": "string",
                "default": "000001"
              }
            },
            {
              "name": "product_detail",
              "in": "query",
              "description": "'name','code','money' for the product",
              "required": true,
              "schema": {
                "type": "array",
                "items": {
                    "type": "dict"
                }
              },
              "example": [
                {"name": "实木茶几","code": "1050201010000000000", "money": 1000},
                {"name": "餐饮费用", "code": "3070401000000000000", "money": 500}
              ]
            },
            {
              "name": "buyer_company_name",
              "in": "query",
              "description": "The name of buyer company",
              "required": true,
              "schema": {
                "type": "string"
              },
              "example": "广东唯一网络科技有限公司"
            },
            {
                "name": "buyer_tax_number",
                "in": "query",
                "description": "The tax number of buyer company",
                "required": true,
                "schema": {
                  "type": "string"
                },
                "example": "91450923MA5L7W2C1W"
              },
            {
              "name": "invoice_type",
              "in": "query",
              "description": "The type of invoice",
              "schema": {
                "type": "string",
                "default": "全电普通发票",
                "enum": ["全电普通发票","全电专用发票"]
              }
            },
            {
                "name": "remark",
                "in": "query",
                "description": "Remarks on the invoice",
                "schema": {
                  "type": "string"
                }
              }
          ],

          "responses": {
            "200": {
              "description": "Generate a temporary preview invoice image
successfully",
              "content": {
                "application/json": {
                  "schema": {
                    "type": "object",
                    "properties": {
                      "status": {
                        "type": "string"
                      },
                      "results": {
                        "type": "string"
                      }
                    }
                  }
                }
              }
            }
          }
        }
      },
```

2.2 创建Agent



Amazon Bedrock > Agents > Create Agent

Step 1
Provide Agent details

Step 2
Select model

Step 3 - *optional*
Add Action groups

Step 4 - *optional*
Add Knowledge base

Step 5
Review and create

Provide Agent details

Agent name

Agent name
InvoiceAgent
Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). The name can have up to 100 characters.

Agent description - *optional*
InvoiceAgent
The description can have up to 200 characters.

User input

Select whether the agent can prompt additional information from the user when it does not have enough information to respond to an utterance.
[✓ Yes] No

IAM Permissions
IAM roles are used to access other services on your behalf.

Agent resource role
[✓ Create and use a new service role] Use an existing service role
AmazonBedrockExecutionRoleForAgents_R8AC25X5WQK

KMS key selection

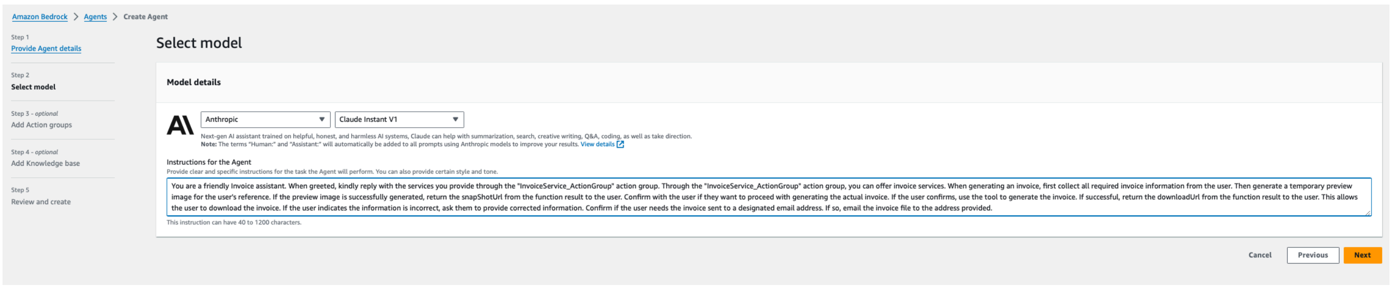
Your data is encrypted by default with a key that AWS owns and manages for you. To choose a different key, customize your encryption settings.
☐ Customize encryption settings (advanced)
To use the default key, clear this option.

Idle session timeout
You can configure how long a session is maintained when the user does not provide any input and the session is idle. Amazon Bedrock retains context information until a session ends.

Session timeout
30 Minute(s)
By default, session idle timeout is 30 minutes, but you can specify any duration between 1 and 60 minutes.

Tags - *optional*

No tags associated with the resource.
Add new tag
You can add up to 50 more tags.



Amazon Bedrock > Agents > Create Agent

Step 1
Provide Agent details

Step 2
Select model

Step 3 - *optional*
Add Action groups

Step 4 - *optional*
Add Knowledge base

Step 5
Review and create

Select model

Model details

Anthropic
Claude Instant V1
Next-gen AI assistant trained on helpful, honest, and harmless AI systems, Claude can help with summarization, search, creative writing, Q&A, coding, as well as take direction.
Note: The terms "Human:" and "Assistant:" will automatically be added to all prompts using Anthropic models to improve your results. View details

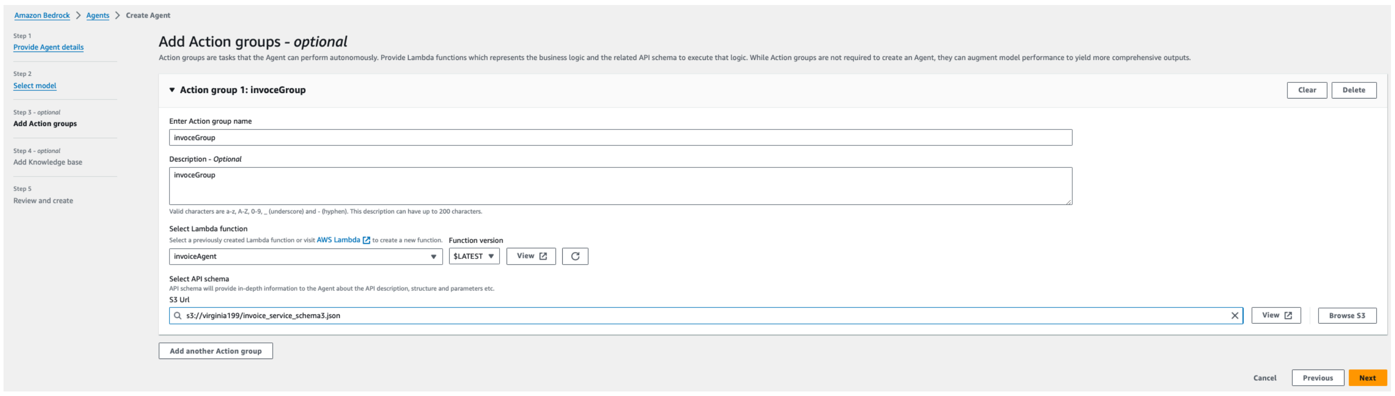
Instructions for the Agent
Provide clear and specific instructions for the task the Agent will perform. You can also provide certain style and tone.

You are a friendly Invoice assistant. When greeted, kindly reply with the services you provide through the "InvoiceService_ActionGroup" action group. Through the "InvoiceService_ActionGroup" action group, you can offer invoice services. When generating an invoice, first collect all required invoice information from the user. Then generate a temporary preview image for the user's reference. If the preview image is successfully generated, return the snapShotUrl from the function result to the user. Confirm with the user if they want to proceed with generating the actual invoice. If the user confirms, use the tool to generate the invoice. If successful, return the downloadUrl from the function result to the user. This allows the user to download the invoice. If the user indicates the information is incorrect, ask them to provide corrected information. Confirm if the user needs the invoice sent to a designated email address. If so, email the invoice file to the address provided.

This instruction can have 40 to 1200 characters.

instruction示例：

You are a friendly Invoice assistant. When greeted, kindly reply with the services you provide through the "InvoiceService_ActionGroup" action group. Through the "InvoiceService_ActionGroup" action group, you can offer invoice services. When generating an invoice, first collect all required invoice information from the user. Then generate a temporary preview image for the user's reference. If the preview image is successfully generated, return the snapShotUrl from the function result to the user. Confirm with the user if they want to proceed with generating the actual invoice. If the user confirms, use the tool to generate the invoice. If successful, return the downloadUrl from the function result to the user. This allows the user to download the invoice. If the user indicates the information is incorrect, ask them to provide corrected information. Confirm if the user needs the invoice sent to a designated email address. If so, email the invoice file to the address provided.


Amazon Bedrock > Agents > Create Agent

Step 1
Provide Agent details

Step 2
Select model

Step 3 - *optional*
Add Action groups

Step 4 - *optional*
Add Knowledge base

Step 5
Review and create

Add Action groups - *optional*

Action groups are tasks that the Agent can perform autonomously. Provide Lambda functions which represents the business logic and the related API schema to execute that logic. While Action groups are not required to create an Agent, they can augment model performance to yield more comprehensive outputs.

Action group 1: invoceGroup

Enter Action group name
invoceGroup

Description - *Optional*
invoceGroup

Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen). This description can have up to 200 characters.

Select Lambda function
Select a previously created Lambda function or visit AWS Lambda to create a new function.
invoiceAgent

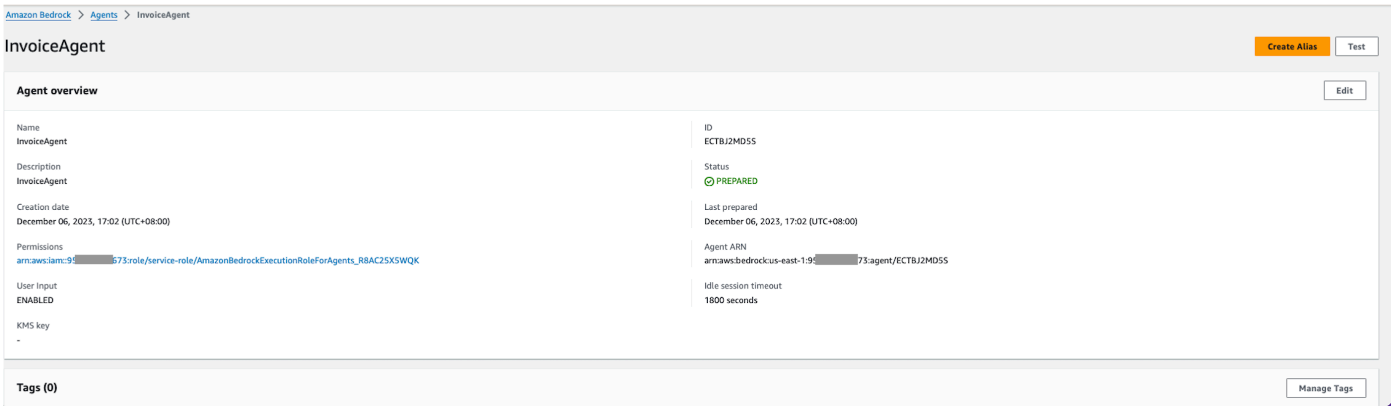
Function version
$LATEST

Select API schema
API schema will provide in-depth information to the Agent about the API description, structure and parameters etc.
S3 Url
s3://virginia199/invoice_service_schema3.json

如果不需要KnowledgeBase可以跳过

2.3 编辑Lambda资源策略，允许Agent访问

创建好agent之后，获得Agent的ARN：


Amazon Bedrock > Agents > InvoiceAgent

InvoiceAgent

Agent overview

Name
InvoiceAgent

Description
InvoiceAgent

Creation date
December 06, 2023, 17:02 (UTC+08:00)

Permissions
arn:aws:iam::9[redacted]673:role/service-role/AmazonBedrockExecutionRoleForAgents_R8AC25X5WQK

User Input
ENABLED

KMS key
-

ID
ECTBJ2MD5S

Status
PREPARED

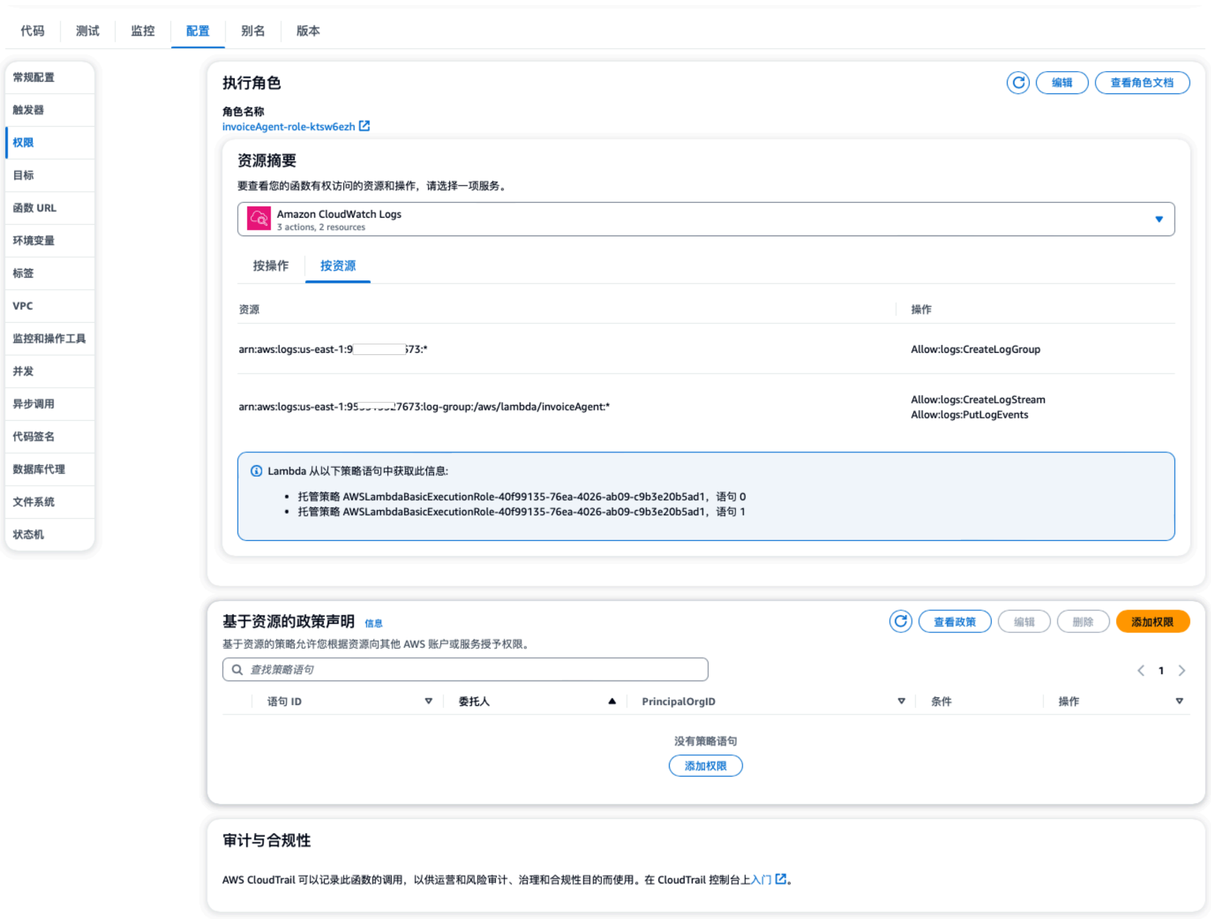
Last prepared
December 06, 2023, 17:02 (UTC+08:00)

Agent ARN
arn:aws:bedrock:us-east-1:9[redacted]73:agent/ECTBJ2MD5S

Idle session timeout
1800 seconds

Tags (0)

在Lambda上构建资源策略，允许Agent访问



执行角色

角色名称
invoiceAgent-role-ktsw6ezh

资源摘要

要查看您的函数有权访问的资源和操作，请选择一项服务。

Amazon CloudWatch Logs
3 actions, 2 resources

资源	操作
arn:aws:logs:us-east-1:9[redacted]573:*	Allow:logs:CreateLogGroup
arn:aws:logs:us-east-1:95[redacted]7673:log-group:/aws/lambda/invoiceAgent:*	Allow:logs:CreateLogStream Allow:logs:PutLogEvents

Lambda 从以下策略语句中获取此信息:

- 托管策略 AWSLambdaBasicExecutionRole-40f99135-76ea-4026-ab09-c9b3e20b5ad1，语句 0
- 托管策略 AWSLambdaBasicExecutionRole-40f99135-76ea-4026-ab09-c9b3e20b5ad1，语句 1

基于资源的政策声明 信息

基于资源的策略允许您根据资源向其他 AWS 账户或服务授予权限。

查找策略语句

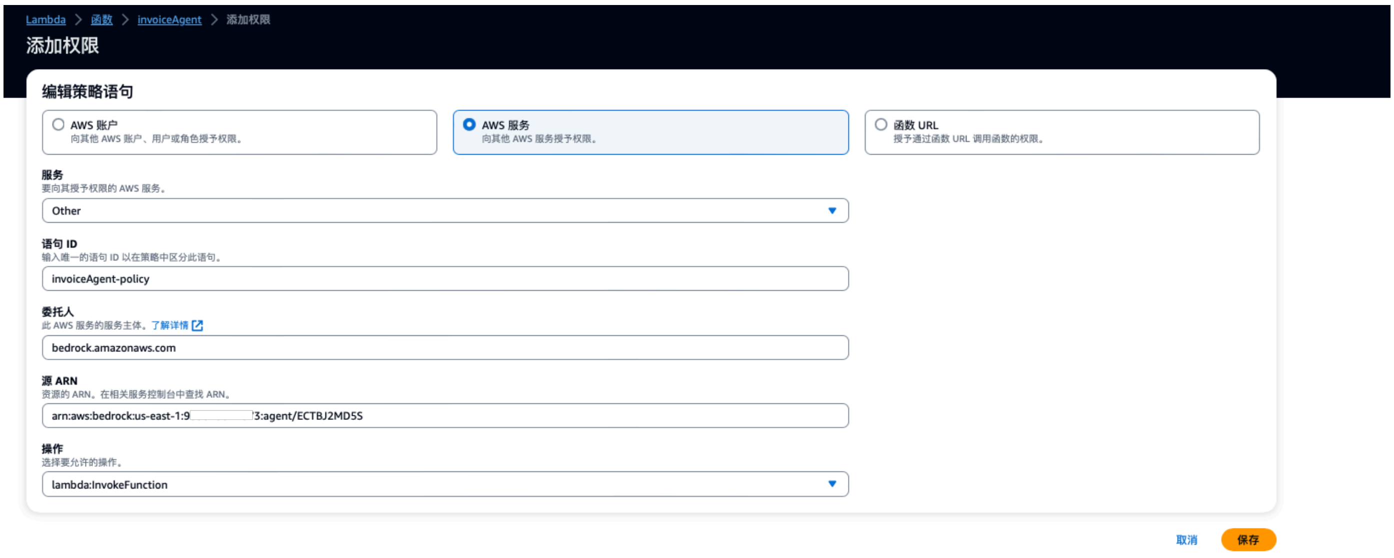
语句 ID	委托人	PrincipalOrgID	条件	操作

没有策略语句

添加权限

审计与合规性

AWS CloudTrail 可以记录此函数的调用，以供运营和风险审计、治理和合规性目的而使用。在 CloudTrail 控制台上入门。



添加权限

编辑策略语句

[ ] AWS 账户 向其他 AWS 账户、用户或角色授予权限。
[✓ AWS 服务 向其他 AWS 服务授予权限。]
[ ] 函数 URL 授予通过函数 URL 调用函数的权限。

服务
要向其授予权限的 AWS 服务。
Other

语句 ID
输入唯一的语句 ID 以在策略中区分此语句。
invoiceAgent-policy

委托人
此 AWS 服务的服务主体。了解详情
bedrock.amazonaws.com

源 ARN
资源的 ARN。在相关服务控制台中查找 ARN。
arn:aws:bedrock:us-east-1:9[redacted]3:agent/ECTBJ2MD5S

操作
选择要允许的操作。
lambda:InvokeFunction

3. 测试

Bedrock agent在console上提供了测试功能，可以直接进行测试