Professor: Danilo Sibov

**Cloud Computing | Aula 16**

**Backups/Snapshots de Instâncias EC2 com AWS Lambda**

**Referência Bibliográfica**

* Acesso ao Conteúdo AWS

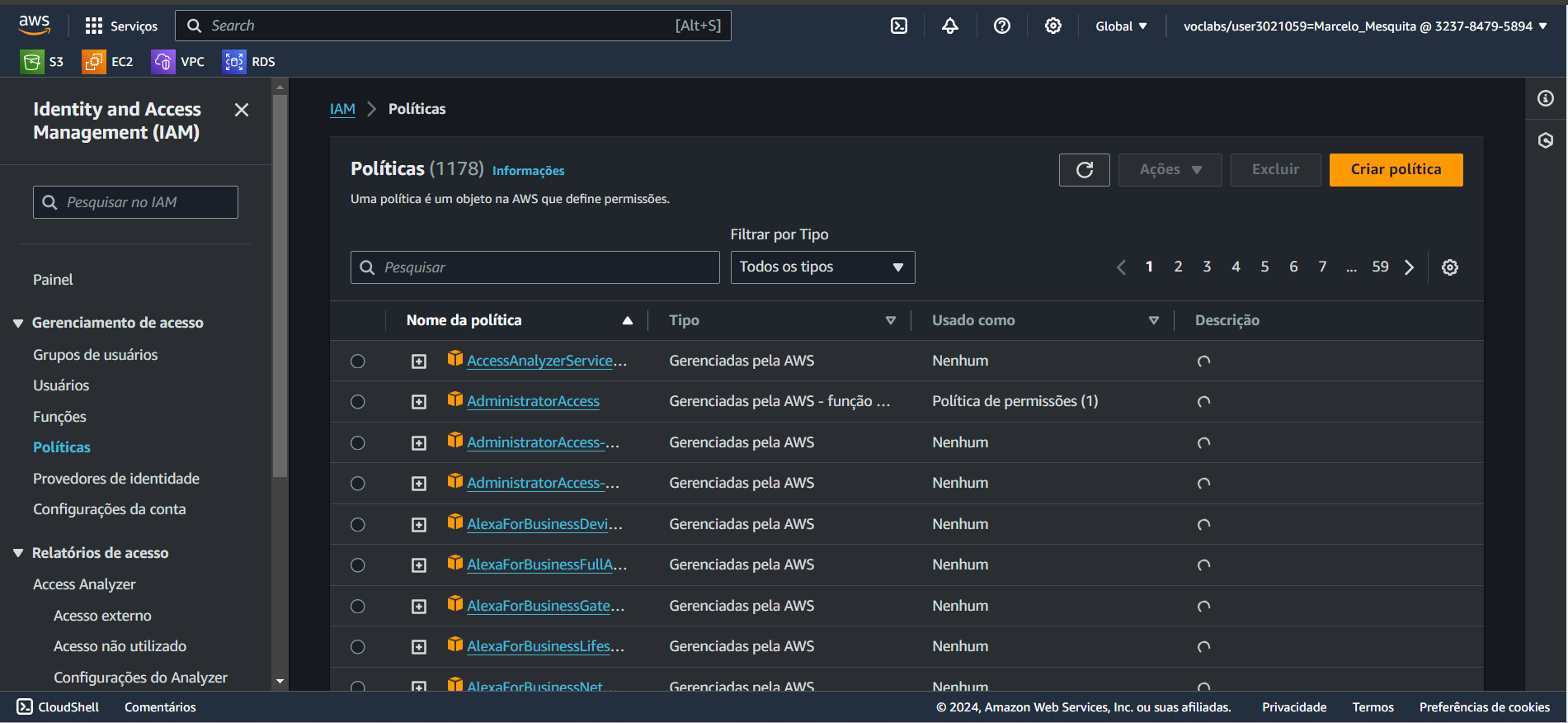
<https://docs.aws.amazon.com/pt_br/cli/latest/userguide/getting-started-install.html>

* Baixar e executar o instalador MSI da AWS CLI para Windows  
  <https://awscli.amazonaws.com/AWSCLIV2.msi>

**Passo a Passo para Agendar Backups/Snapshots de Instâncias EC2 com AWS Lambda:**

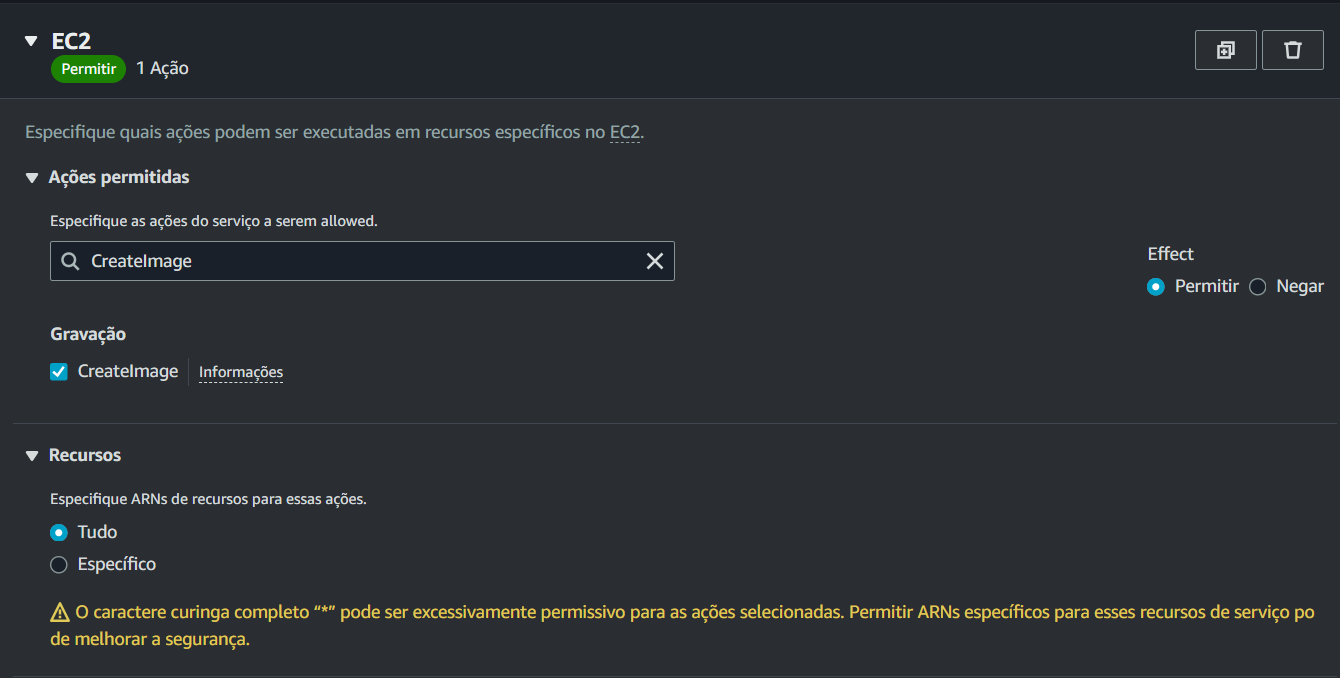
1. **Acesso à AWS:**
2. **Acesso ao IAM:**

* Vá para o Console IAM.
* No painel de navegação à esquerda, clique em "Políticas" e, em seguida, "Criar política".

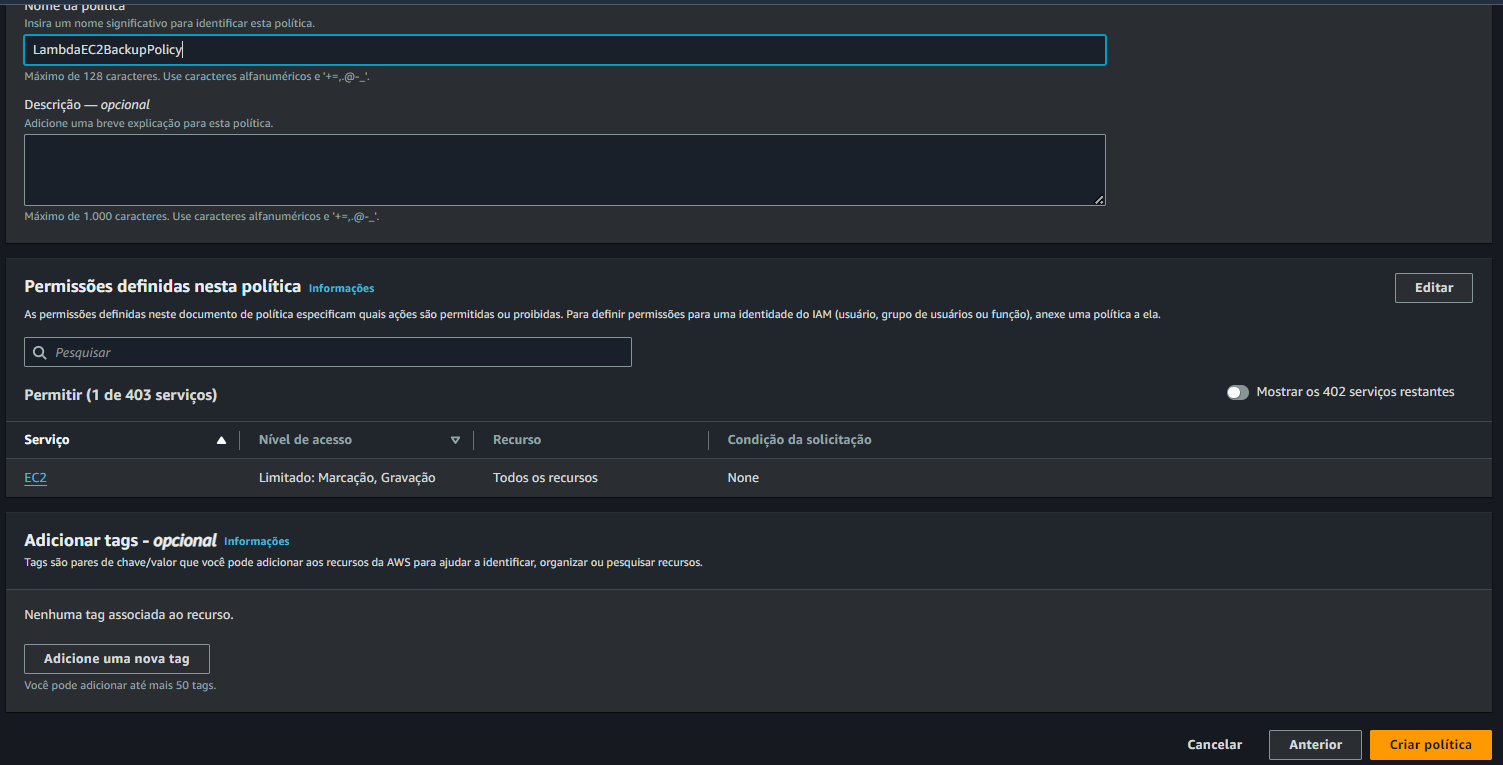


1. **Criar Política Customizada:**

* Selecione o serviço EC2 e adicione as permissões CreateImage e CreateTags. Recursos: Todos

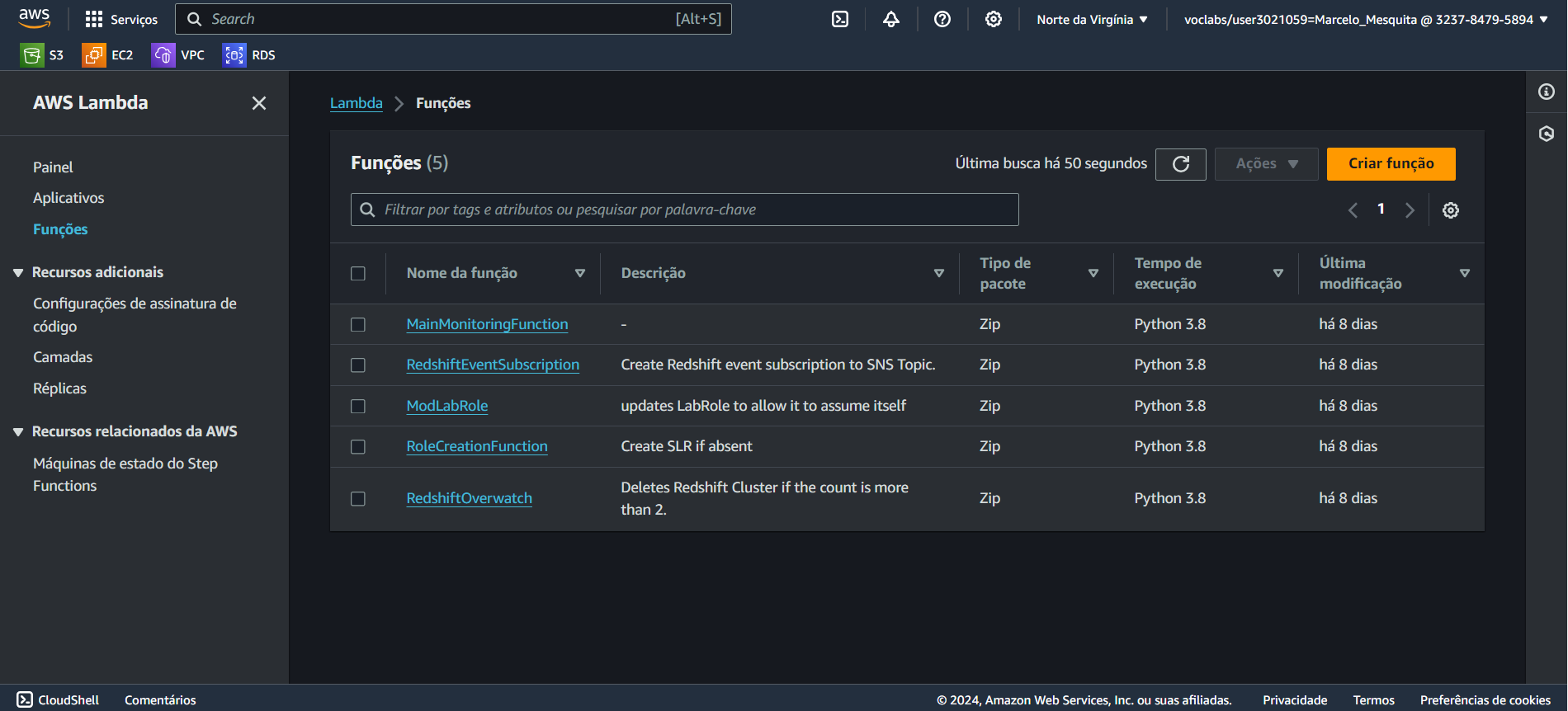


* Clique em "Próximo" e, em seguida, em "Revisar política".
* Dê um nome (por exemplo, "LambdaEC2BackupPolicy") e clique em "Criar política".

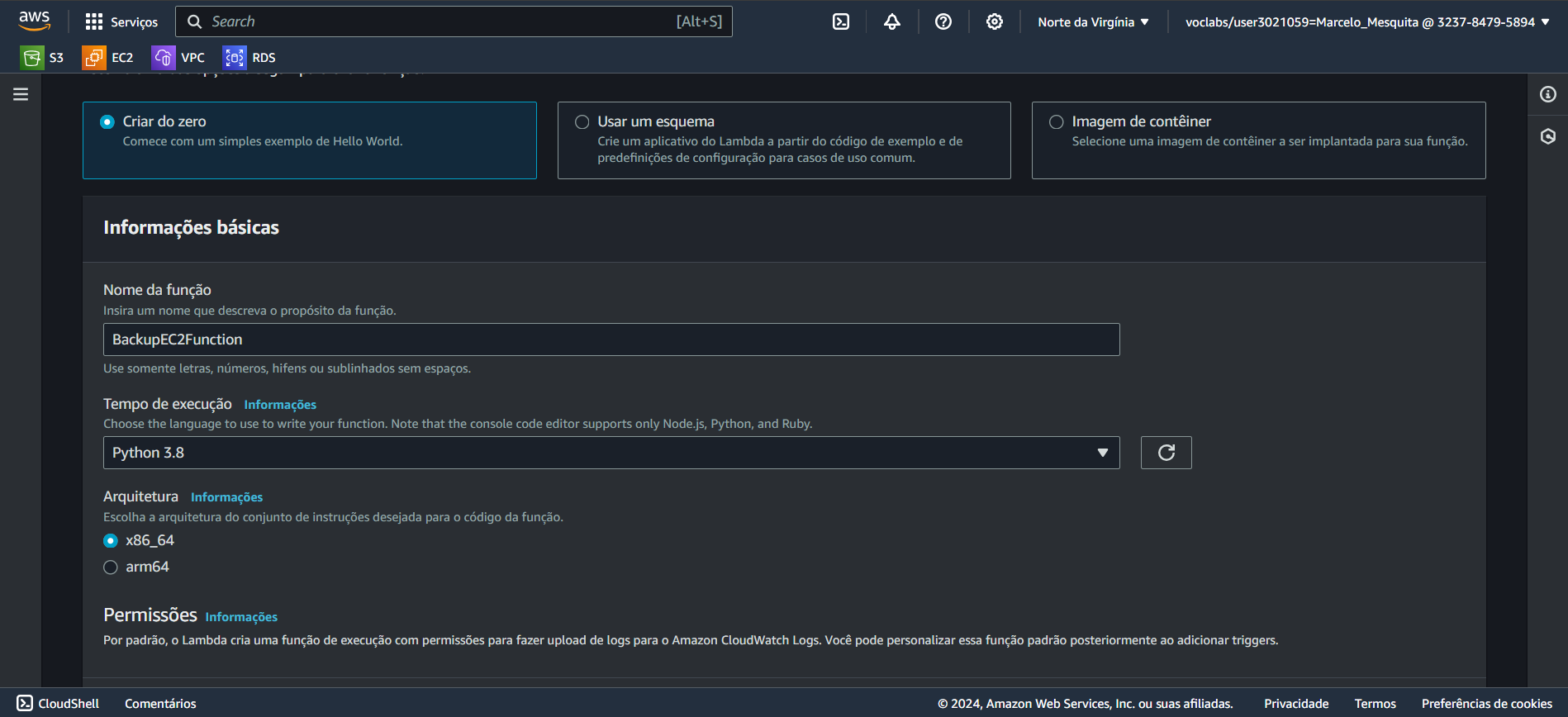


1. **Associar Política à Função Lambda:**

* Vá para o Console Lambda.
* Clique em "Criar função" no canto superior direito.

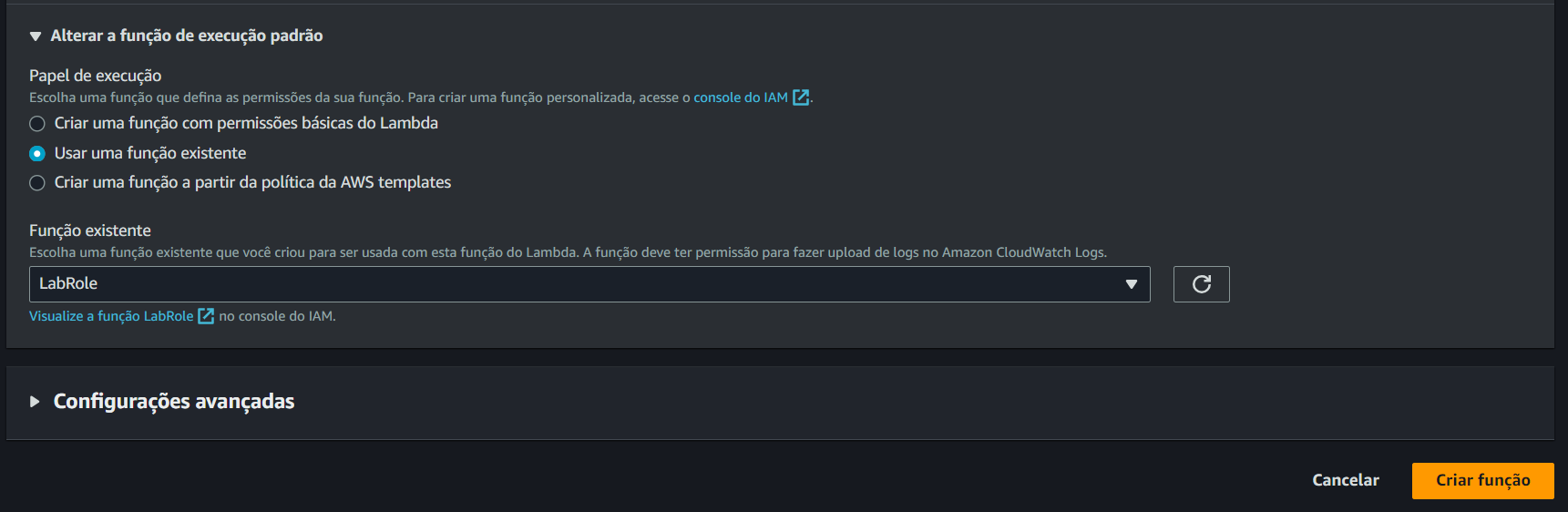
  
  
  
  
  
  
**Configurar Detalhes da Função:**

* Escolha "Nome da Função" e dê um nome para a função (por exemplo, "BackupEC2Function").
* Escolha "Python 3.8" ou uma versão mais recente como ambiente de execução.



* Em Alterar a função de execução padrão, clique em Usar Função existente e selecione a Opção LabRole

Criar Função

  
 **Associar Política à Função Lambda:**

* Depois de criar a função, Clique em Configuração e vá para a guia "Permissões".
* Clique em "Anexar políticas".
* Pesquise e anexe a política customizada que você criou anteriormente (por exemplo, "LambdaEC2BackupPolicy").
* Selecione a função Lambda criada anteriormente.
* Role até "Permissões da Função" e clique em "Anexar Políticas".
* Pesquise e anexe a política criada ("LambdaEC2BackupPolicy").

1. **Adicionar Código Python à Função Lambda:**

* Substitua o código da função Lambda com o Código Da Última Página
* Salve a função Lambda.

1. **Configurar Gatilho:**

* Na página da função Lambda, clique em "Adicionar Gatilho".
* Escolha "EventBridge (CloudWatch Events)" como tipo de gatilho.
* Configure uma regra do CloudWatch Events com uma expressão cron para agendar a execução (ex: cron(0 0 \* \* ? \*) para diariamente).

1. **Testar Manualmente:**

* Na página da função Lambda, clique em "Testar" para garantir que não há erros.
* Verifique os logs para mensagens de sucesso.

1. **Verificar Resultados:**

* Monitore os logs do CloudWatch para mensagens de sucesso.
* Vá para o Console do EC2 e confira se novas AMIs foram criadas**.**

Este passo a passo e código Python ajudarão a configurar um backup automático de instâncias EC2 com AWS Lambda. Lembre-se de adaptar conforme necessário, especialmente em relação à frequência de agendamento e permissões específicas. **Código Python para Backup de Instâncias EC2:**

import boto3

from datetime import datetime

def backup\_instances():

date\_format = "%d/%m/%Y\_%H\_%M\_%S"

created\_by\_value = "lambda-backup-script"

# Inicialize o cliente EC2

ec2\_client = boto3.client('ec2')

# Obtenha todas as instâncias EC2

instances = ec2\_client.describe\_instances()

for reservation in instances['Reservations']:

for instance in reservation['Instances']:

# Ignore instâncias com estados inválidos

if instance['State']['Name'] == "running" \

or instance['State']['Name'] == "stopping" \

or instance['State']['Name'] == "stopped":

date\_time = datetime.now().strftime(date\_format)

ami\_name = f"backup\_by\_lambda\_{instance['InstanceId']}\_{date\_time}"

ami\_desc = ami\_name

print(f"Fazendo backup da instância: {instance['InstanceId']} >> {ami\_desc}")

# Crie uma AMI (Amazon Machine Image) da instância

ami\_id = ec2\_client.create\_image(

InstanceId=instance['InstanceId'],

Name=ami\_name,

Description=ami\_desc,

NoReboot=True

)

# Adicione tags à AMI criada

ec2\_client.create\_tags(

Resources=[ami\_id['ImageId']],

Tags=[

{'Key': 'creation\_date', 'Value': date\_time},

{'Key': 'created\_by', 'Value': created\_by\_value}

]

)

def lambda\_handler(event, context):

print('Iniciando backup...')

backup\_instances()

print('Backup concluído com sucesso!')