Roteiro spike Bitbucket + Sagemaker

# Step 1: Set up an AWS CodeCommit Repository

The first step is to create an AWS CodeCommit repository, which will serve as the bridge between Bitbucket and SageMaker. Follow these steps:

* Log in to the AWS Management Console and navigate to the CodeCommit service.

.\saml2aws.exe console -a eec-aws-br-eits-devhub-sandbox

* Create a new repository or select an existing one that you want to connect to Bitbucket.

aws codecommit create-repository --repository-name integracao-sagemaker --repository-description "Integracao Bitbucket com Sagemaker" --profile devhub-sandbox --no-verify-ssl

* Take note of the repository URL, as you will need it in the next steps.

{

"repositoryMetadata": {

"accountId": "071087690196",

"repositoryId": "f60bbc43-3871-4243-a942-91c97f172b66",

"repositoryName": "integracao-sagemaker",

"repositoryDescription": "Integracao Bitbucket com Sagemaker",

"lastModifiedDate": "2023-10-19T10:24:53.510000-03:00",

"creationDate": "2023-10-19T10:24:53.510000-03:00",

"cloneUrlHttp": "https://git-codecommit.sa-east-1.amazonaws.com/v1/repos/integracao-sagemaker",

"cloneUrlSsh": "ssh://git-codecommit.sa-east-1.amazonaws.com/v1/repos/integracao-sagemaker",

"Arn": "arn:aws:codecommit:sa-east-1:071087690196:integracao-sagemaker"

}

}

# Step 2: Configure Bitbucket Repository

Next, configure your Bitbucket repository to integrate with AWS CodeCommit. Here’s how:

Go to your Bitbucket repository’s settings.

Under the “Settings” tab, find and click on “Webhooks” or “Hooks” (depending on your Bitbucket version).

Create a new webhook or hook.

Configure the webhook to trigger events related to your desired actions, such as pushing commits or creating pull requests.

Set the URL of the webhook to the AWS CodeCommit repository URL obtained in Step 1.

Save the webhook configuration.

## My steps:

➊Go to Repository: [Settings for EITS Engineering and Quality / experian-qes - Bitbucket Global](https://code.experian.local/projects/ENANQU/repos/experian-qes/settings)

➋Create Webhook:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Settings:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Copy URL from the repository I’ve just created in CodeCommit: <https://sa-east-1.console.aws.amazon.com/codesuite/codecommit/repositories/integracao-sagemaker/setup?region=sa-east-1>

Additional Settings

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Finally, mark the webhook as active and save it. (there’s a test button if you want to check it)

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

# Step 3: Configure AWS Identity and Access Management (IAM) Role

To enable SageMaker to access the necessary resources, such as S3 buckets or other AWS services, you need to configure an IAM role. Follow these steps:

* Create an IAM role in the AWS Management Console.
* Assign policies to the IAM role that grant necessary permissions for accessing the required resources.
* Ensure the IAM role is associated with your SageMaker notebook instance.

## My Steps:

I’ve created a ROLE SageMaker-Datacientist with the recommended policies in document: [SageMaker Roles - Amazon SageMaker](https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-roles.html) and add an AWS Managed Policy to get access to CodeCommit

# Step 4: Set up SageMaker Notebook Instance

Before connecting SageMaker with Bitbucket, set up a SageMaker notebook instance or use an existing one. Here’s what you need to do:

* Access the AWS Management Console and navigate to the SageMaker service.
* Create a new notebook instance or use an existing one that you want to connect with Bitbucket.
* Ensure the IAM role associated with the notebook instance has appropriate permissions to access the required resources.

## My Steps:

➊Create a Lifecycle Configuration to a Notebook Instance – This is necessary to adjust route to access SERASA internal services:

#!/bin/bash

set -e

sudo ip route add 10.0.0.0/8 via 10.120.134.1 dev eth2

Create a notebook instance with the following data:

* Name: DevHub
* Lifecycle configuration: arn:aws:sagemaker:sa-east-1:071087690196:notebook-instance-lifecycle-config/Adjusting-route-to-SERASA
* Notebook instance type: ml.t3.medium
* IAM role ARN: arn:aws:iam::071087690196:role/service-role/SageMaker-Datacientist
* Subnet(s): subnet-01ae076c3421d58a9
* Security Group(s): sg-0976d87fb4c4776fe
* Direct internet access: Enabled

# Step 5: Connect SageMaker with Bitbucket

Now that you have completed the necessary configurations, it’s time to connect SageMaker with Bitbucket. Follow these steps:

* Open your SageMaker notebook instance and clone the Bitbucket repository using the AWS CodeCommit repository URL and Git commands.
* Make changes to your code or notebooks as required within the SageMaker environment.
* Use Git commands, such as git commit and git push, to commit and push your changes back to the Bitbucket repository.
* Set up a workflow where your SageMaker notebooks can pull the latest changes from Bitbucket using Git commands like git pull.