

Manual de Sistema

Serviços utilizados:

- **Alexa Skill Kit**

O Alexa Skill Kit (ASK) é um conjunto de recursos que fornece meios para desenvolver habilidades para a Alexa. Habilidades são aplicativos para a Alexa. Através do serviço de voz da Alexa o usuário pode interagir com suas habilidades.

- **Alexa Voice Service**

Com o Alexa Voice Service, os fabricantes de dispositivos conseguem integrar a Alexa aos seus produtos. Isso significa que os aparelhos conectados passarão a contar com controle de voz.

- **AWS amplify**

O AWS Amplify é um conjunto de ferramentas e serviços que podem ser usados em conjunto ou individualmente para permitir que desenvolvedores de front-end da web e de dispositivos móveis criem aplicações seguras e escaláveis full-stack, com tecnologia desenvolvida pela AWS.

- **Amazon Cognito**

O Amazon Cognito oferece funções de autenticação e gerenciamento de usuários para proteger a API do back-end.

- **Amazon Api Gateway**

O Amazon API Gateway é um serviço onde é possível criar, publicar e gerenciar APIs com facilidade. Funciona como uma porta de entrada, com ele você consegue criar APIs HTTP RESTful, que são a melhor escolha quando estamos falando de APIs web, ou, também, a criação de APIs do tipo WebSocket.

Com o API Gateway conseguimos administrar todo o recebimento e processamento de centenas de milhares de chamadas simultâneas. Sendo possível gerenciar todo o tráfego HTTP, configurar CORS, adicionar controle de acesso e até criar um controle de versões da nossa API.

- **AWS lambda**

É um serviço de computação sem servidor(serverless computing) que executa código em resposta a eventos e gerencia automaticamente os recursos de computação exigidos por esse código.

- **AWS IoT core**

O AWS IoT é uma plataforma de nuvem gerenciada que permite a interação fácil e segura de dispositivos conectados, ou seja, carros, lâmpadas, grades sensoras, etc., com aplicativos de nuvem e outros dispositivos.

- **IAM - Authorization for IoT Services**

É responsável por controlar e criar usuários de seus serviços web. Além disso, neste console você pode definir funções e políticas para restringir ou autorizar o que seus usuários podem usar ou não.

Conectando placa com serviço AWS IoT core

Criando uma coisa no AWS IoT core

1. Acessar AWS Management Console → Acessar serviço IoT core;
2. Na sessão do IoT core acesse guia Gerenciar → Coisas → Criar;
3. Selecione a opção de criar um único item → Dê o nome a coisa → Avançar;
4. Criar certificado com um clique → Baixe o certificado e a chave privada para cada dispositivo e também uma CA raiz;
5. Ative o certificado clicando em ativar → Concluído;

Criando e anexando política ao certificado

1. Clique na guia Proteger → Políticas → Criar;
2. Insira o nome da política → Preencha as opções Ação e ARN com "*" → Marque a opção permitir → Selecione Criar;
3. Selecione a guia Certificados → No certificado criado clique nos três pontos → Selecione anexar política → Anexe a política recém criada;

Converter Certificado baixados de .PEM para .DER

1. Baixar OpenSSL para seu sistema operacional;
2. Separe os arquivos Certificate, Private Key, Root CA em uma pasta, abra o console do sistema operacional e execute os comandos:
> openssl x509 -in xxxxxxxxxx-certificate.pem -out cert.der -outform DER
> openssl rsa -in xxxxxxxxxx-private.pem.key -out private.der -outform DER
> openssl x509 -in AmazonRootCA1.pem -out ca.der -outform DER
3. Separe os arquivos .DER gerados para uma pasta Data;

Configurar IDE Arduino

1. Baixe e instale a IDE Arduino;
2. Ao abrir a IDE vá em arquivos → Preferências → Em "URL do gerenciador de placa adicional" → Adicione:
"http://arduino.esp8266.com/stable/package_esp8266com_index.json"
3. Vá na opção Ferramentas → Placa → Gerenciador de placa → Procure e Instale ESP8266;
4. Selecione a placa correta, usamos a Node MCU 1.0 (ESP-12E module), com a seguinte configuração:
Tamanho do Flash: "4M (2M SPIFFS)"
Porta de depuração: "Desativada"
Nível de depuração: "Nenhum"
Variante IWIP: "V2 Lower Memory"
Frequência da CPU: "80Mhz"
Velocidade de upload: "115200"
Apagar Flash: "Sketch On"

Porta: "Porta COM disponível" (porta do dispositivo)

5. Coloque a pasta Data com os Certificados no mesmo diretório do projeto Arduino e carregue-os para a placa na opção → Ferramentas → ESP8266 Sketch data upload.

Criando Regras para armazenando dados no DynamoDB (AWS IoT Rule Actions)

1. Acessar console IAM → Role → selecionar Lambda → Anexar permissões AmazonDynamoDBFullAccess e AWSIoTFullAccess → nomear a função → criar
2. Acessar console do AWS IoT core → Guia agir → Regras → Criar;
3. Insira um nome e uma descrição (opcional) para a regra;
4. Em Instrução da consulta da regra → Selecione a versão mais recente do SQL;
5. Insira uma Instrução da consulta da regra por exemplo `SELECT * FROM 'iot/topic'` (Seleciona todos os dados '*' enviados no tópico 'iot/topic');
6. Em Adicionar Ação → Lambda → Clique em criar novo recurso;
7. Ao abrir o console do AWS lambda → selecione a blueprint dynamodb-process-stream;
8. Selecione DynamoDB Table -> Selecione sua tabela → Em Role selecione escolher função existente → Selecione a função criada no passo anterior → Dê um nome para a função lambda → Selecione Próximo e insira o código da função lambda;
9. Voltando para o console do AWS IoT selecione a função lambda recém criada→ Clique em adicionar ação → Por fim selecione Criar Regra;

Criação De uma aplicação WEB

Instalando o Amplify CLI

1. Como pré-requisito tenha o Node.js versão 10.x e NPM versão 6.x instalados na máquina;
2. Instale o amplify CLI executando na linha de comando "npm install -g @aws-amplify/cli";
3. Instale também o AWS CLI;
4. Como estamos fazendo uso de uma conta do Aws Educate com recursos limitados não é possível criar novos usuários IAM, então faremos uso do usuário Root. Após instalar o AWS CLI acesse o diretório /.aws na raiz do sistema, e altere o conteúdo do arquivo Credentials com suas credenciais da aws.

Configurando projeto JavaScript

1. Abra a linha de comando de seu sistema operacional → Execute o comando “mkdir -p amplify-js-app/src && cd amplify-js-app” para criar pasta do projeto → Em seguida execute o comando “touch package.json index.html webpack.config.js src/app.js” para criar arquivos essenciais ao projeto;
2. Adicione o conteúdo ao package.json:

```
{
  "name": "amplify-js-app",
  "version": "1.0.0",
  "description": "Amplify JavaScript Example",
  "dependencies": {
    "aws-amplify": "latest"
  },
  "devDependencies": {
    "webpack": "^4.17.1",
    "webpack-cli": "^3.1.0",
    "copy-webpack-plugin": "^6.1.0",
    "webpack-dev-server": "^3.1.5"
  },
  "scripts": {
    "start": "webpack && webpack-dev-server --mode development",
    "build": "webpack"
  }
}
```
3. Execute o comando “npm install” na linha de comando para instalar as dependências;
4. Após implementar o index.html e configurar o webpack.config.js, rode o comando “npm start” para iniciar um servidor de desenvolvimento acessando através <http://localhost:8080>.

Configurando serviços de back-end

1. Inicie a linha de comando e execute “amplify init” no diretório do projeto;
2. Ao iniciar o amplify será solicitada algumas informações. Após inicializar o projeto amplify é criado um diretório que contém definições e recursos do back-end. Também é criado um projeto na nuvem que pode ser acesso através do console do aws amplify;

Criando Banco de dados DynamoDB

1. Execute na linha de comando “amplify add storage” para adicionar um banco de dados ao projeto;
2. Será solicitado as informações referente a criação de uma tabela como, nome, chave de index, chave de classificação, etc.,

Criando API REST

1. Execute na linha de comando “amplify add api” para adicionar uma api ao projeto;
2. Será solicitado informações para a criação da api, como nome e caminho de acesso, durante o processo também será criada uma função lambda para acessar o storage criado anteriormente;
3. Para salvar as alterações realizadas execute o comando “amplify push” para enviar as alterações para a nuvem.

Criando Serviço de autenticação (Amplify usa Cognito como provedor de autenticação)

1. Na pasta raiz do projeto execute o comando “amplify add auth”;
2. Selecione Default configuration para usar configuração padrão → Selecione Username modo de autenticação;
3. Execute “amplify push” para salvar modificações na nuvem;

Estabelecendo Comunicação MQTT entre o site e AWS IoT

1. Importe as bibliotecas PubSub, AWSIoTProvider para o projeto no arquivo src/app.js;
2. Defina endpoint e região no arquivo src/app.js;
3. Anexe a política do dispositivo no aws IoT a identidade Amazon Cognito utilizando o AWS CLI;
4. Recupere o Cognito Identity Id no console do serviço Cognito..
5. Execute o comando na linha de comando “ aws iot attach-principal-policy --policy-name 'myIoTPolicy' --principal '<YOUR_COGNITO_IDENTITY_ID>' ”
6. Acesse o AWS Console e selecione o serviço CloudFormation → Procure pela pilha pai → Selecione a guia recursos → Selecione AuthRole → ID físico;
7. Será aberto o console do Aws IAM onde será anexada as políticas AWSIoTDataAccess e AWSIoTConfigAccess;

Conectando ambiente frontend a provedor de repositório e branch principal

1. Acesse o console amplify no AWS Management Console → Selecione a guia front → Selecione o provedor de repositório github;
2. Na tela de configurações de compilação → escolha um ambiente de back-end existente → Escolha dev na lista suspensa e conceda a função de serviço ao Console do Amplify → Escolha Salvar e implantar.]

Configurando a Alexa

Criando uma Skill no console da Alexa Skill

1. Fazer login no portal do desenvolvedor da Amazon → Navegue até Alexa → Seus consoles Alexa → Habilidades;
2. botão “Criar habilidade” → inserir um nome de habilidade → Selecione a guia “Personalizar” e crie a habilidade;
3. Em seguida, você será direcionado para a página Habilidades → Navegue para guia de invocação → fornecer o comando na seção de nome de invocação de habilidade;

4. Você pode criar uma nova intent usando o botão Adicionar intent → Fornecer nome para Intent → fornecer os enunciados de amostra para esse intent;
5. Na seção de expressões, você pode especificar o Slots necessário entre chaves. → Será solicitado a criação de um novo Slot → Selecione o tipo de Slot;
6. Após a criação da Intent é possível fazer o controle por voz na Alexa;

Criando função de backend Lambda

1. No console da AWS Acessar Lambda → Criar função → Insira um nome → selecione a linguagem Node.js 10.XX → selecione criar função a partir do modelo → selecione criar;
2. Insira o código da função lambda→ No console da AWS selecione IAM → Na seção funções procure pela função criada no passo anterior → selecione a função e anexe as políticas AmazonDynamoDBFullAccess e AWSIoTFullAccess;

Ligue a função lambda e a habilidade

1. Copie o valor ARN da função Lambda para a área de transferência;
2. Vá para sua habilidade personalizada e selecione a guia EndPoint e selecione o AWS Lambda ARN;
3. Insira o valor ARN na seção Região padrão;
4. Voltando para a função lambda selecione Adicionar gatilho “Alexa Skills Kit” → Indo para a seção Configurar do Alexa Skills Kit e adicione o ID de habilidade de sua habilidade personalizada.
5. Salve a função Lambda e a habilidade personalizada.
6. Em seguida, construa a habilidade personalizada usando o botão Construir modelo.