Configuring Gen AI Chatbot Project

You will need access to the following project:

lhasa-limited-products / gen-ai-chatbot — Bitbucket

Clone this project into your workspace.

You will also need access to the AWS project. Login to the following link and confirm you can access **PracticeAccount1**.

Accounts | AWS access portal

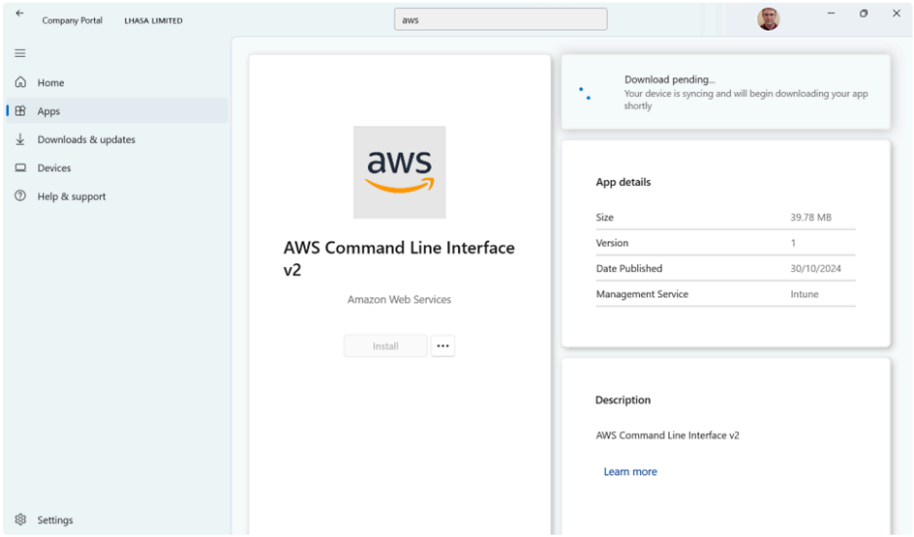
If you do not have access, request this directly from the Cloud team.

Instructions for installing AWS CLI and CDK applications are detailed below:

Windows

Installing AWS CLI

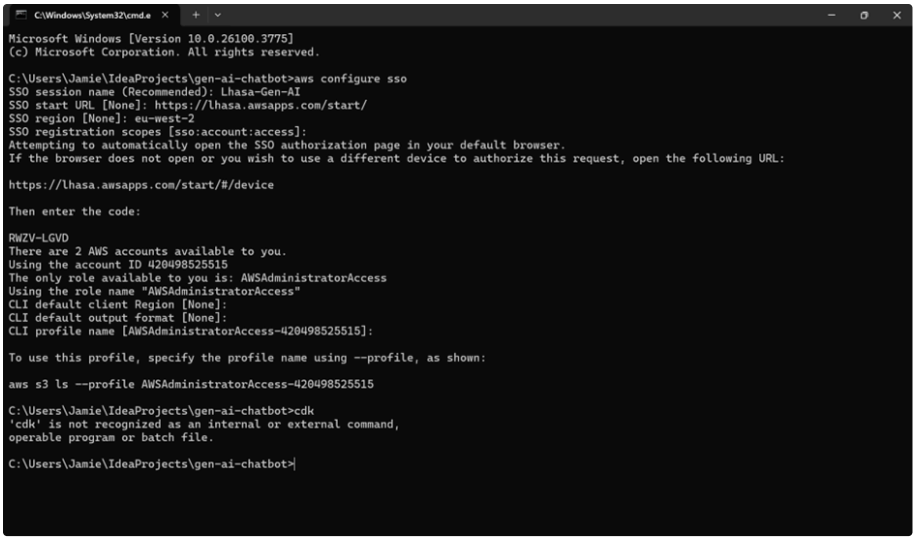
Install AWS Command Line Interface v2 from Company Portal



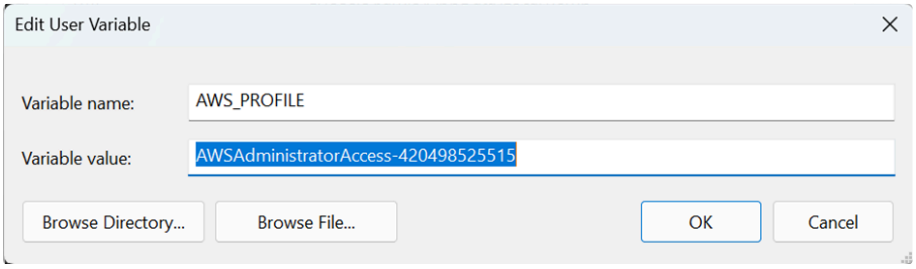
Once installed, open a new command prompt and run the following command:

```
aws configure sso
```

Enter the values as shown below, many values will be auto-populated



The profile to use in future will be shown in the output, in my case it was AWSAdministratorAccess-420498525515. To avoid having to enter this every time, edit your user account variables and add the parameter AWS_PROFILE with this value.



On subsequent interactions with the aws command, login with:

```
aws sso login
```

If this does not work, just go through the aws configure sso steps again.

Installing CDK

You must have NodeJS preinstalled, if you do not, install this using Company Portal.

Install CDK globally with the following command:

```
npm install -g aws-cdk
```

Now verify installation with the command:

```
cdk --version
```

If the version is shown, CDK was installed successfully. If however you see the error “**'cdk' is not recognized as an internal or external command, operable program or batch file.**”, do the following, run the command:

```
npm config get prefix
```

Copy the path that is returned and add it to the PATH variable in your User Environment Variables. Open a new command prompt, and try the version command again.

Linux

Installing AWS CLI

My installation was done using Ubuntu 24.04 in WSL. The following steps may need adapting depending on your situation.

First, try to install AWS CLI using apt install:

```
sudo apt install awscli
```

If this does not work, you will have to install manually by downloading the install package, unpacking and then installing. To do this you will need to ensure that unzip is installed. Perform the following instructions:

```
sudo apt install unzip
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

Confirm that the AWS CLI was installed by running the command:

```
aws --version
```

Configure the AWS CLI by running the following command and then doing the same as for the Windows install above.

```
aws configure sso
```

The profile to use will be displayed in the output. To prevent having to enter this every time, edit the ~/.bashrc file and add the parameter as below and save

AWS_PROFILE=<The profile that the output gives you>

Subsequent interactions with the aws cli should only need you to enter

aws sso login

However if this dos not work, just run aws configure sso again

Installing CDK

First install Node

```
sudo apt install nodejs npm -y
```

Then install CDK globally

```
sudo npm install -g aws-cdk
```

Finally, check CDK installation with

```
cdk –version
```

Using CDK

Navigate to the CDK folder within the cloned project.

Install all dependencies with the following command

```
npm ci
```

Now confirm you are able to build the project using the command, this will use the development environment to build. The cdk.json file will contain a list of environments that can be built.

```
cdk synth -c env=development
```

This should work, but if not, it is likely that setting the profile as an environment variable did not work. If this happens, you can try rerunning the command with an explicit profile

```
cdk synth -c env=development --profile <Your profile name>
```

To deploy the project to AWS, replace synth with deploy

```
cdk deploy -c env=development
```