Contents

[Prerequisites 1](#_Toc145841863)

[Configure CLI User 1](#_Toc145841864)

[Permissions 2](#_Toc145841865)

[GitHub 2](#_Toc145841866)

[Install CDK 4](#_Toc145841867)

[Verify Installation 4](#_Toc145841868)

[CDK Init 4](#_Toc145841869)

[Templates 4](#_Toc145841870)

[Init Base App 5](#_Toc145841871)

[Bootstrap 5](#_Toc145841872)

[Alias 5](#_Toc145841873)

[Do It! 5](#_Toc145841874)

[Adjust User permissions 6](#_Toc145841875)

[Verify CDK Stack 6](#_Toc145841876)

[Commit 6](#_Toc145841877)

# Prerequisites

* Java 8+
* Maven
* NodeJs
* AWS CLI

## Configure CLI User

[This section](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html) explains how to configure the settings that the AWS Command Line Interface (AWS CLI) uses to interact with AWS.

$ aws configure --profile mig  
AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ABCD]:  
AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WxyZ]:  
Default region name [eu-central-1]:  
Default output format [json]:

# To help us not confuse other profiles setup on our env, we create a profile specific to the mig  
$ aws sts get-caller-identity --profile mig  
{

"UserId": "…",

"Account": "0000000000",

"Arn": "arn:aws:iam::0000000000:user/mig"

}

# But then to make it quicker to use, and more explicit to our profile, we can create an alias for it

$ alias awsm='aws --profile mig'

$ awsm sts get-caller-identity

{

"UserId": "AIDAZVRHILQVRHCEL4Y2E",

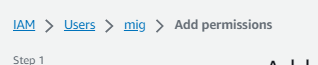
"Account": "000000000000",

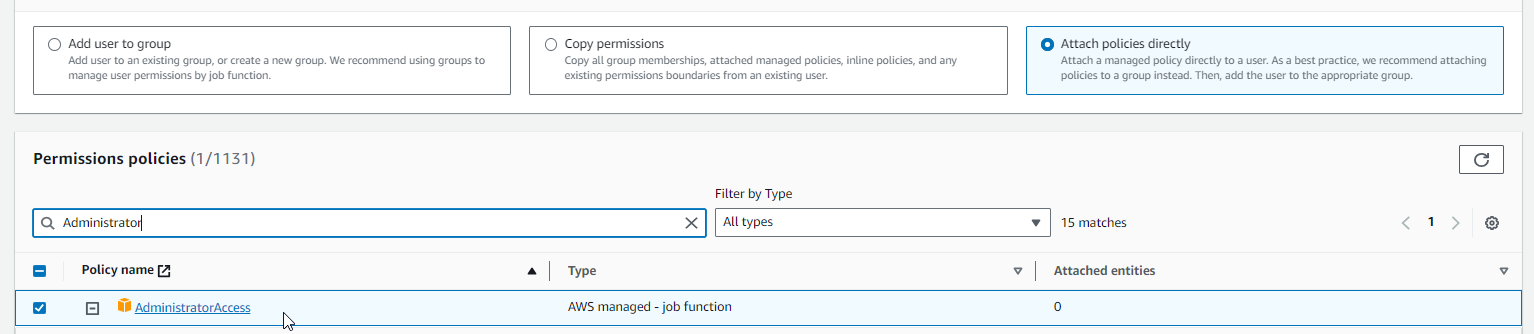
"Arn": "arn:aws:iam::000000000000:user/mig"

}

### Permissions

In order to run the bootstrap, it need set of permissions. To set these manually, is out of scope for this session, so we’ll temporarily give it admin access.

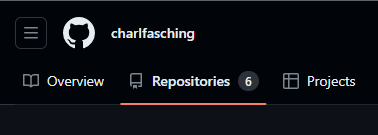




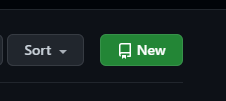
**Not something you should leave on your CLI user long term, It’s dangerous**

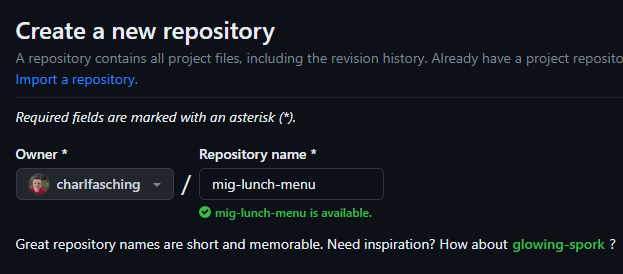
# GitHub

Let’s log into [github](https://github.com/login)

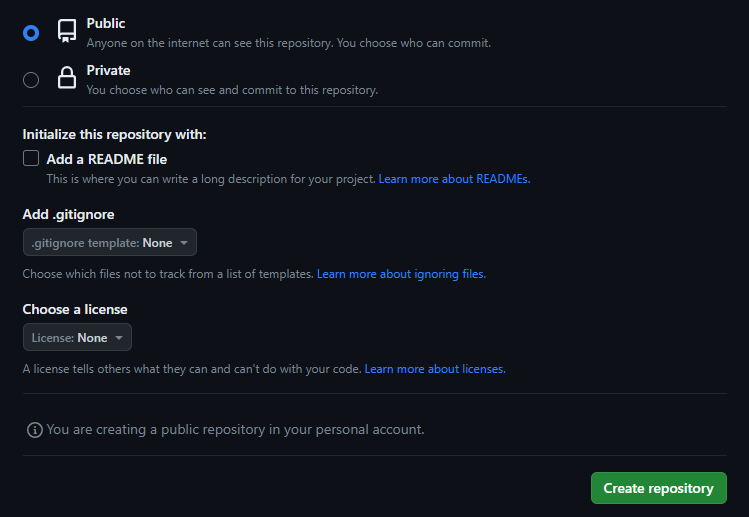


Go to Repositories, and select the green “new” button



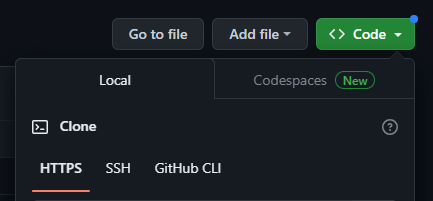


We need to create a new repo called mig-lunch-menu



Choose which Protocol:

* HTTPS
* SSH



$ git clone https://github.com/charlfasching/mig-lunch-menu.git

Cloning into 'mig-lunch-menu'...

warning: You appear to have cloned an empty repository.

$ cd mig-lunch-menu

# Let’s make a folder to place our AWS infra into

$ mkdir infra

# Install CDK

We can install cdk cli using npm

$ npm install -g aws-cdk

*Packages: +1*

*+ Progress: resolved 230, reused 229, downloaded 0, added 0, done*

*+ aws-cdk 2.95.1*

*Done in 1.6s*

## Verify Installation

$ cdk version

2.96.2 (build xxx)

To see the available command for the cdk cli, provide the help parameter.

$ cdk –help

…

To start with a new project, we can ask for the different templates available.

Here is the [official repo](https://github.com/aws-samples/aws-cdk-examples) for cdk to see more info on it

# CDK Init

Init is used to help start a project, it creates the basic configuration needed to run a CDK project.

## Templates

These provide frequently used patterns for easier use

$ cdk init --list

Available templates:

\* **app**: Template for a CDK Application

└─ cdk init app --language=[csharp|fsharp|go|java|javascript|python|typescript]

\* lib: Template for a CDK Construct Library

└─ cdk init lib --language=typescript

\* sawsmle-app: Exawsmle CDK Application with some constructs

└─ cdk init sawsmle-app --language=[csharp|fsharp|go|java|javascript|python|typescript]

## Init Base App

We need to setup an empty dir to run cdk init in

**Note!** Before running next command, make sure **java** and **maven** are set correct on the $Path

$ cdk init app --language=java

# Welcome to your CDK Java project!

….

## Useful commands

\* `mvn package` compile and run tests

\* `cdk ls` list all stacks in the app

\* `cdk synth` emits the synthesized CloudFormation template

\* `cdk deploy` deploy this stack to your default AWS account/region

\* `cdk diff` compare deployed stack with current state

\* `cdk docs` open CDK documentation

$ ls

cdk.json pom.xml README.md src/ target/

## Bootstrap

Bootstrapping is the process of provisioning resources for the AWS CDK before you can deploy AWS CDK apps into an AWS environment. (An AWS environment is a combination of an AWS account and Region).

These resources include an Amazon S3 bucket for storing files and IAM roles that grant permissions needed to perform deployments.

*( Taken from* [*AWS docs*](https://docs.aws.amazon.com/cdk/v2/guide/bootstrapping.html) *)*

### Alias

In order for us to use easily use cdk with our mig profile let’s create another alias, call it cdkm

$ alias cdkm="cdk --profile mig"

### Do It!

Make sure you are within the folder where we ran the cdk init.

$ cdkm bootstrap

⏳ Bootstrapping environment aws://000000000000/eu-central-1...

….

✅ Environment aws://000000000000/eu-central-1 bootstrapped.

### Adjust User permissions

When done bootstrapping we need to reduce the permissions of mig user.

We need to specify policy for mig user to assume cdk role.

$ cat > assume-role.json << EOF

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"sts:AssumeRole"

],

"Resource": [

"arn:aws:iam::\*:role/cdk-\*"

]

}

]

}

EOF

$ awsm iam put-user-policy --user-name mig --policy-name Assume-CDK --policy-document file://assume-role.json

# Next, we’ll add permission for doing Cloudformation activities

$ awsm iam attach-user-policy --user-name mig --policy-arn arn:aws:iam::aws:policy/AWSCloudFormationFullAccess

# Lastly, we need to remove AdminAccess

$ awsm iam detach-user-policy --user-name mig --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

## Verify CDK Stack

Now that we’ve setup our Stack, let’s verify it

$ cdkm list stack   
MigLunchMenuStack

# By running a diff, it will see if there is something locally to deploy

$ cdkm diff  
… some meta data …

✨ Number of stacks with differences: 1

$ cdkm deploy

MigLunchMenuStack

✨ Deployment time: 17.07s

Stack ARN:

arn:aws:cloudformation:eu-central-1:000000000000

✨ Total time: 26.5s

Well Done! You have successfully bootstrapped an empty CDK App.

## Commit

To commit our empty stack into Github, we go up one folder, and add commit, you should see the following

$ ls   
infra/

$ git commit -m   
git commit -m "Bootstrap CDK"  
…

$ git push   
…

Tada!