### **Setting up AWS**

If you’re already an expert in AWS, what you’ll need is

* IAM credentials (and stored locally too)
* A bucket (and region)

and you can skip the rest of this section and go directly to Writing our function.

If like me however, you’re mostly new to managing AWS, not only can it be extremely overwhelming, but the help is also overwhelming too. In this section I’ll briefly walk you through what you need to get the above set up.

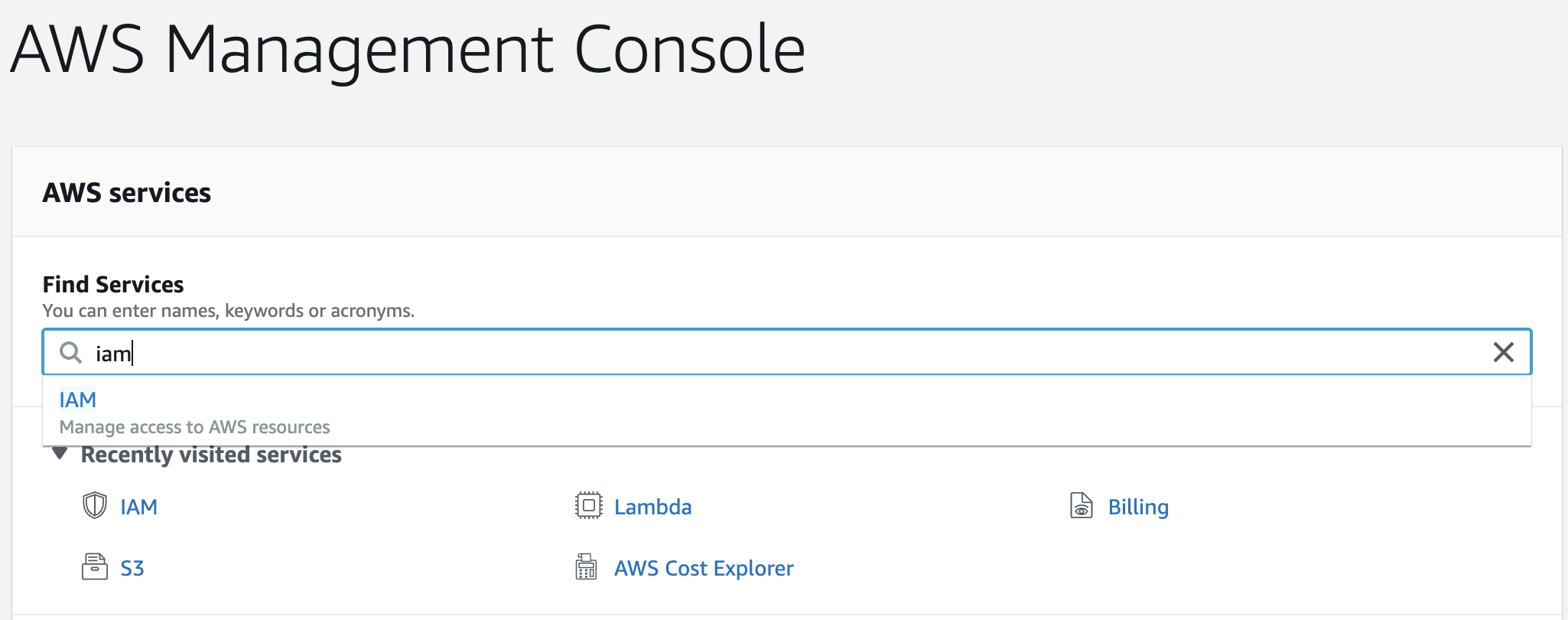
**Step 1 -** [**Create an AWS account**](https://portal.aws.amazon.com/billing/signup?nc2=h_ct&src=default&redirect_url=https%3A%2F%2Faws.amazon.com%2Fregistration-confirmation#/start).

You need to provide credit card details, but stick to the free tier and you should be fine. In addition, make sure you set up some alerts in case you suddenly go beyond what free provides. For these demos though you most certainly won’t hit anything beyond free!

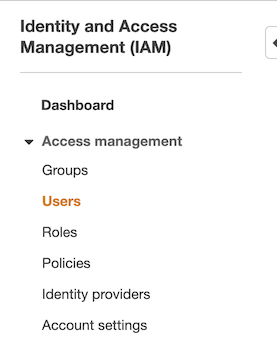
While you may want to download [AWS CLI](https://aws.amazon.com/cli/) for certain management aspects, note that Kotless doesn’t need it.

**Step 2 - Create IAM credentials**

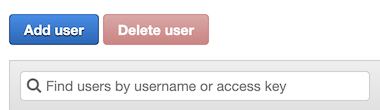
This is required by Kotless (Terraform actually) to deploy your functions. To do this, go to the [AWS Management Console](https://aws.amazon.com/console/) (make sure you’re logged in) and search for IAM



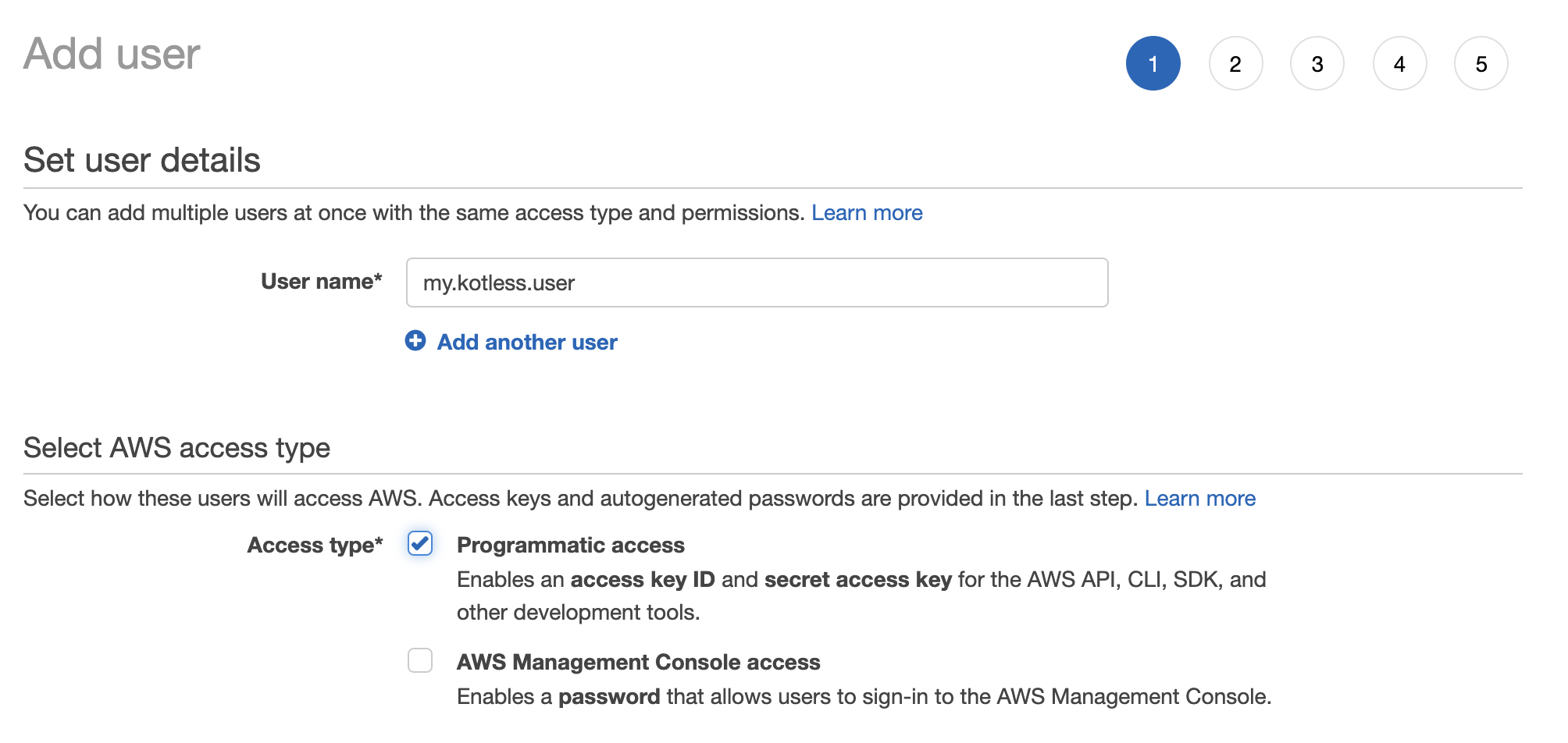
Once in the IAM section, proceed to create a new user account by clicking on Users



and then the **Add User** button

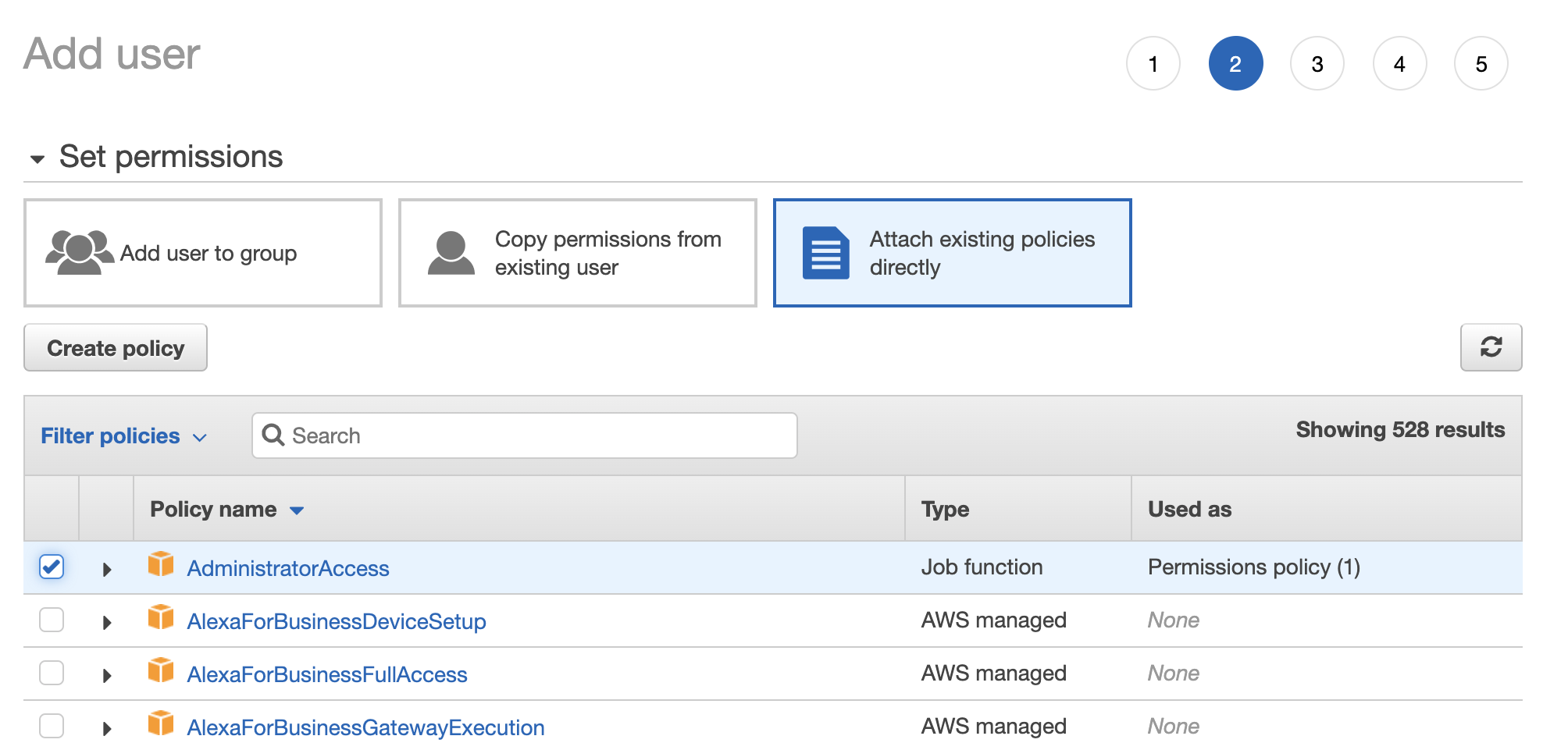


This takes you through a series of steps to provide information for the new user. It’s important to remember here the name we give the user, which is the same one which will be defined on our local machine for Kotless to use. For this example, I’ve named it **my.kotless.user**.

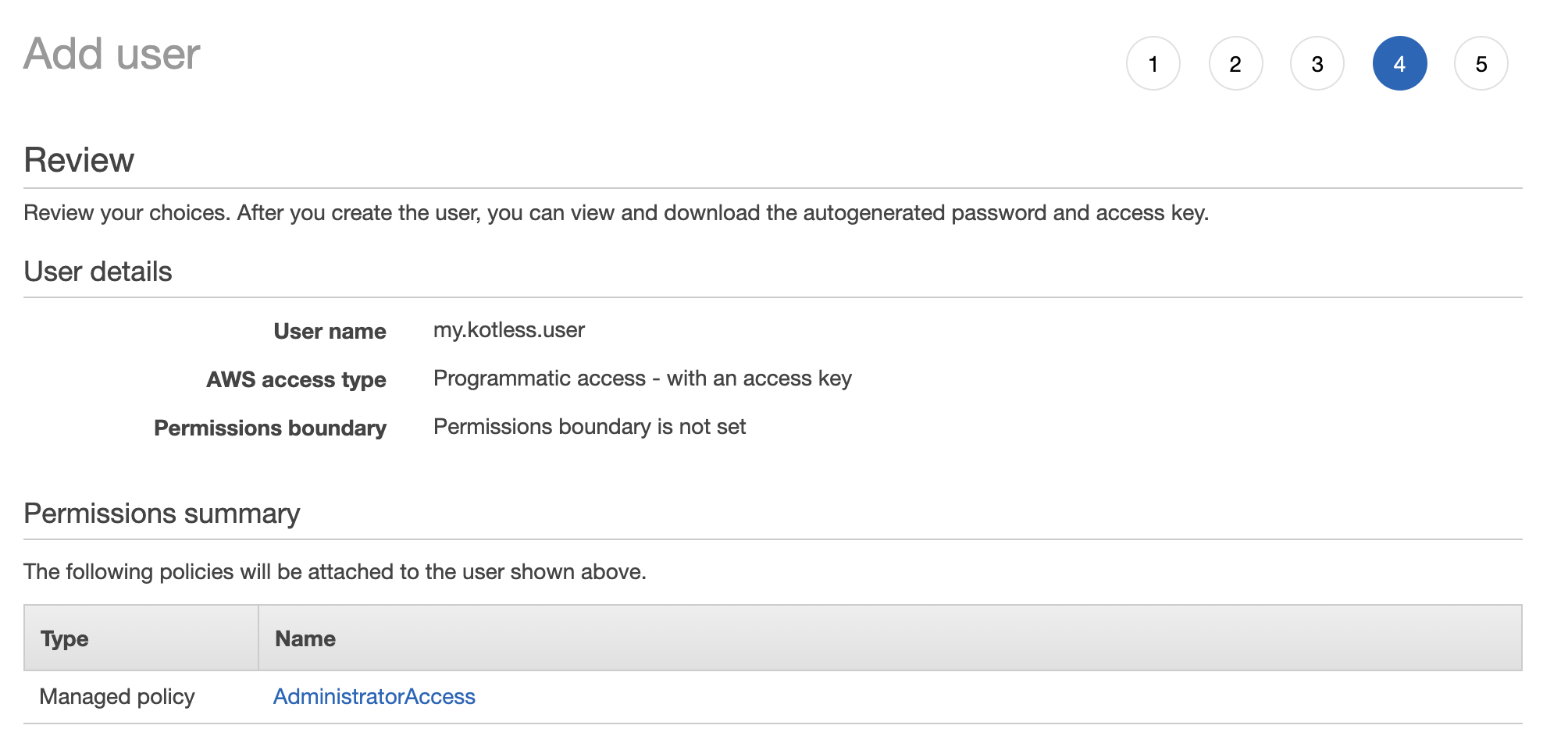


Make sure **Programmatic access** is ticked.

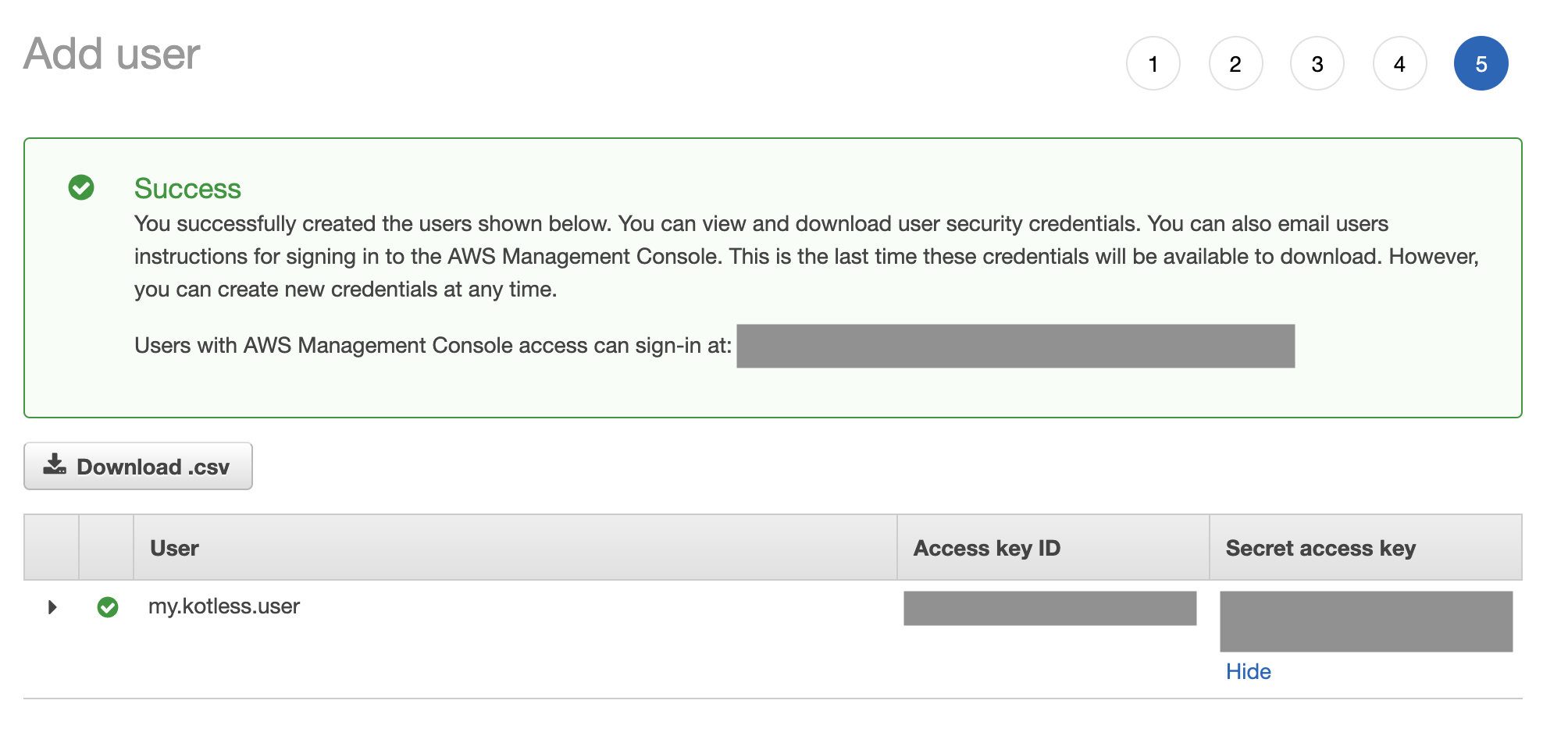
In the next step we’re going to define permissions. Obviously this needs to be fine-tuned based on what’s needed. For now we’re going to give full Admin



The next step we’ll skip (as it’s to define tags), leading us to the final step which is to review and create the user.

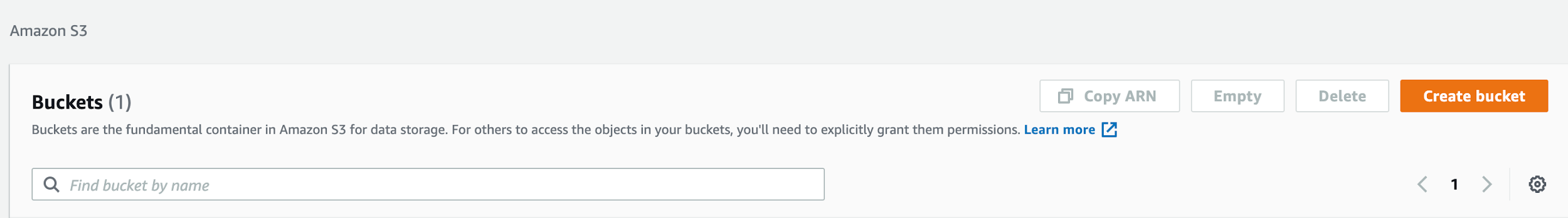


Once done, you’ll be prompted with the user along with two values: **Access key ID** and **Secret Access** key.

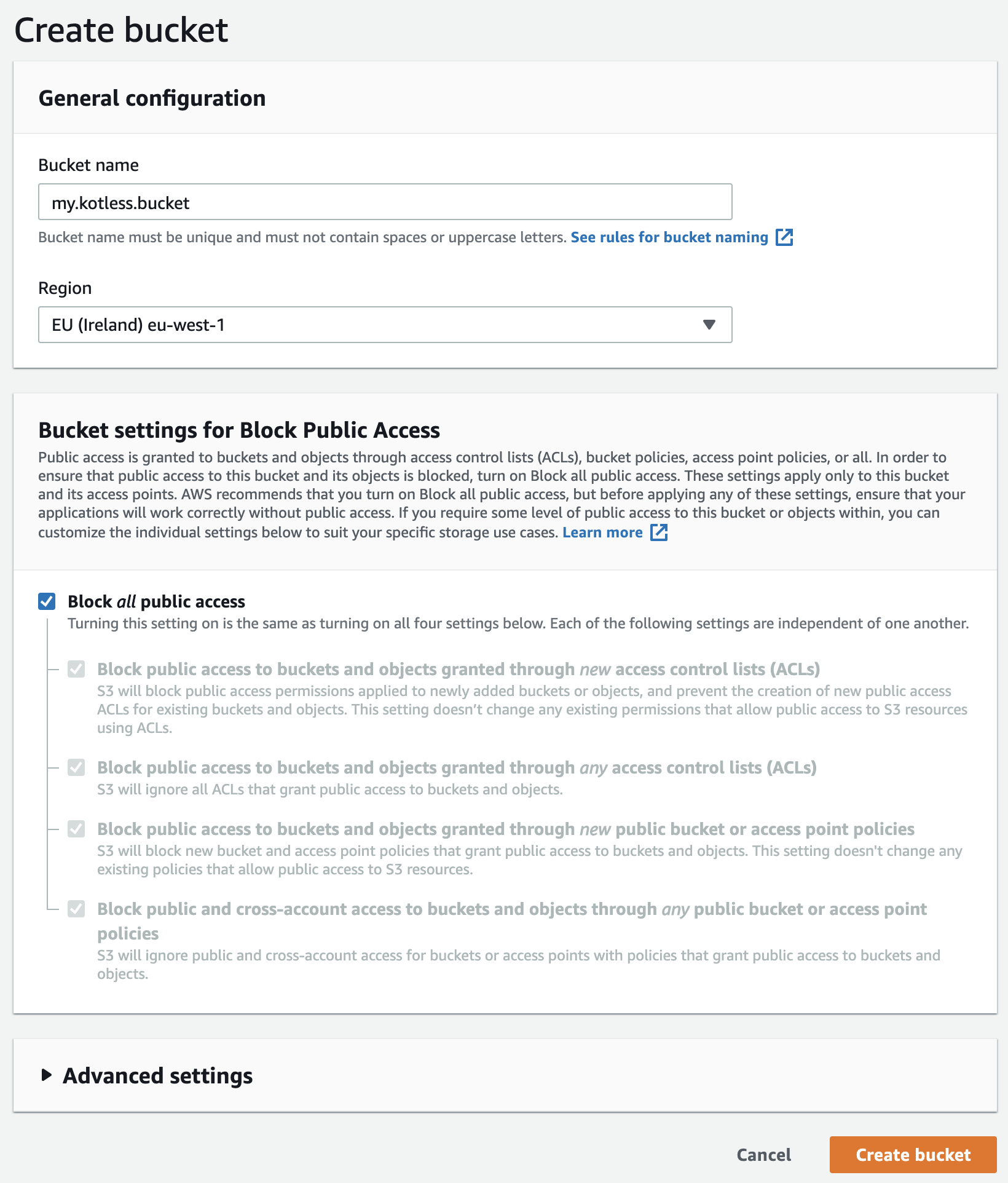


**Step 3 - Create an S3 bucket**

Go back to the AWS Management Console and search for S3.



Click on **Create bucket** providing a name and region. Again, keep track of these two values as we’ll use them later.



Leave all other options as default.

**Step 4 - Store the credentials locally**

Kotless is going to need access to the credentials created, and we need to somehow provide these. These are stored in the user directory (on macOS/Linux this would be ~/.aws and on Windows in the home directory).

Create a file name **~/.aws/credentials** and type in the following contents

[profile my.kotless.user]

aws\_access\_key\_id={the\_access\_key\_id}

aws\_secret\_access\_key={the\_secret\_access\_key}

Notice how the profile name matches the name of the IAM credential we created earlier.

And that’s it. We’re now ready to write our function and deploy with Kotless.

**Important** - When you set up your AWS account, the system itself asks you to follow a series of good practices, such as removing root access, setting up MFA, defining groups with restricted permissions, etc. It’s important to go back and do this at some point. I’m avoiding it in here cause I know HOW EXCITED YOU ARE TO SEE THIS WORK! So let’s move on.