

# Configuring Access Credentials

In this lesson, you will learn how to set up access credentials for AWS CLI.

## WE'LL COVER THE FOLLOWING ^

- SAM configuration
  - AWS regions
- Verification

## SAM configuration #

AWS SAM CLI reuses the credential configurations from AWS command-line tools. If you already have credentials set up for AWS CLI, skip this section.

To deploy software to the AWS cloud, you will need an access key ID and a secret key ID associated with your user account. If you do not have these already, here is how you can generate a set of keys:

1. Sign in to the AWS Web Console at <https://aws.amazon.com/>.
2. Select the Identity and Access Management (IAM) service.
3. In the left-hand IAM menu, select *Users*.
4. Click on the *Add User* button.
5. On the next screen, enter a name for the user account then, in the 'Select AWS access type' section, select *Programmatic access*.

The screenshot shows the AWS IAM Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information 'Gojko Adzic - Claudia Test'. The main heading is 'Add user' with a progress indicator showing four steps, with the first step '1' being active. Below the heading is the section 'Set user details'. A text input field for 'User name\*' contains the text 'testuser'. Below this field is a link '+ Add another user'. The next section is 'Select AWS access type', with a subtext explaining that access keys and passwords are provided in the last step. There are two radio button options: 'Programmatic access' (selected) and 'AWS Management Console access'. The 'Programmatic access' option is described as enabling an 'access key ID' and 'secret access key' for the AWS API, CLI, SDK, and other development tools. The 'AWS Management Console access' option is described as enabling a 'password' for sign-in to the console. At the bottom of the form, there is a '\* Required' label, a 'Cancel' button, and a 'Next: Permissions' button. The footer contains a 'Feedback' link, 'English (US)' language selection, and copyright information for Amazon Web Services, Inc. (2008-2019).

aws Services Resource Groups Gojko Adzic - Claudia Test Global Support

## Add user

1 2 3 4

### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\* testuser

+ Add another user

### Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\* ☒ **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

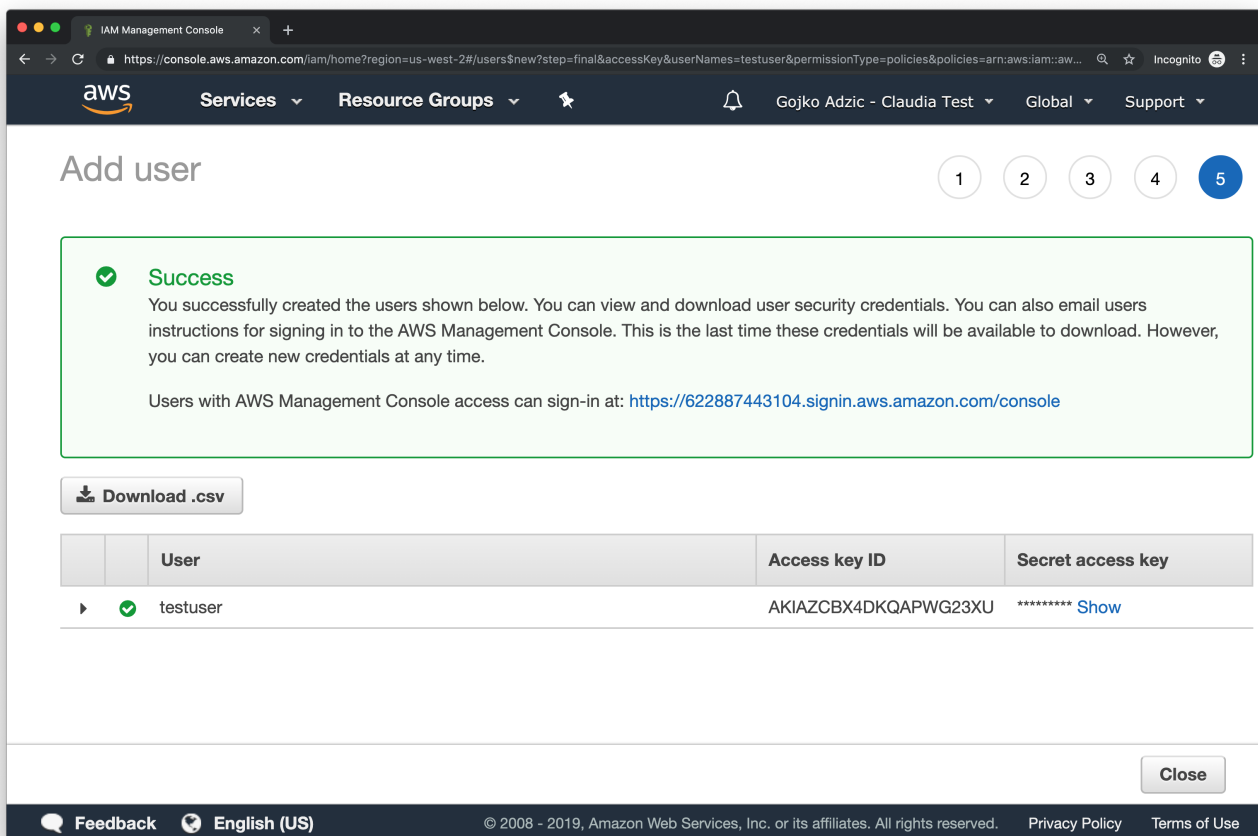
☐ **AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required Cancel Next: Permissions

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Create a user with programmatic access rights for SAM and AWS command line tools.

- Click the *Next* button to assign permissions, then select *Attach existing policies directly*.
- In the list of policies, find the **PowerUserAccess** and **IAMFullAccess** policies and tick the checkboxes next to them.
- You can skip the remaining wizard steps. The final page will show the access key ID and show a link to reveal the secret key as shown in the figure below. Reveal the secret key and copy both keys somewhere.



In the final step, reveal and copy the access key and secret key.

Once you have the access keys, you may run the following command to save the keys to your local machine:

```
aws configure
```

When the AWS utility asks you about the keys, paste what you copied in the previous step. You will also likely be asked to enter a default region and a default output format.

For the default output format, enter `json`, or just press Enter to keep it unset.

For the region, use `us-east-1` or check whether your IT administrators have a preference. Because AWS adds new regions and services frequently, for a full list of available options, it is best to check out the [AWS regions and endpoints](#) documentation page.

For the sake of this course, please enter your access key ID, secret access key and `us-east-1` in the fields `AWS_ACCESS_KEY_ID` and

`AWS_SECRET_ACCESS_KEY` , and `AWS_REGION` respectively.

Environment Variables		^
Key:	Value:	
LANG	C.UTF-8	
LC_ALL	C.UTF-8	
AWS_ACCESS_KEY_ID	Not Specified...	
AWS_SECRET_ACCE...	Not Specified...	
BUCKET_NAME	Not Specified...	
AWS_REGION	Not Specified...	
● Terminal		↺ ^

## AWS regions

AWS has data centers all over the world. The region setting tells the command line tools which data center to use. Region selection is useful to ensure that user data is hosted in a specific country for compliance reasons and to speed up data transfers by using the closest available access point. The original AWS data center in North Virginia is `us-east-1` . Generally, new services launch in that region first, so it is a safe setting for experiments.

## Verification #

Check that your credentials are correctly configured by running the following command line:

```
aws sts get-caller-identity
```

If this command prints a result similar to the following, everything works correctly in the terminal provided above:

```
$ aws sts get-caller-identity
{
  "UserId": "111111111111",
  "Account": "222222222222",
  "Arn": "arn:aws:iam:1111111111:root"
}
```

If you get an error, check out the section [\*Configuring the AWS CLI\*](#) from the AWS CLI user guide for troubleshooting information.

Now that you are done with the account configurations, you'll see how to run the AWS services with restricted user accounts.