### Installing Node.js

Serverless is a [Node.js](https://nodejs.org/) CLI tool so the first thing you need to do is to install Node.js on your machine.

Go to the official [Node.js website](https://nodejs.org/), download and follow the [installation instructions](https://nodejs.org/en/download/) to install Node.js on your local machine.

**Note:** Serverless runs on Node v4 or higher.

You can verify that Node.js is installed successfully by running node --version in your terminal. You should see the corresponding Node version number printed out.

### Installing the Serverless Framework

Next, install the Serverless Framework via [npm](https://npmjs.org/) which was already installed when you installed Node.js.

Open up a terminal and type npm install -g serverless to install Serverless.

npm install -g serverless

Once the installation process is done you can verify that Serverless is installed successfully by running the following command in your terminal:

serverless

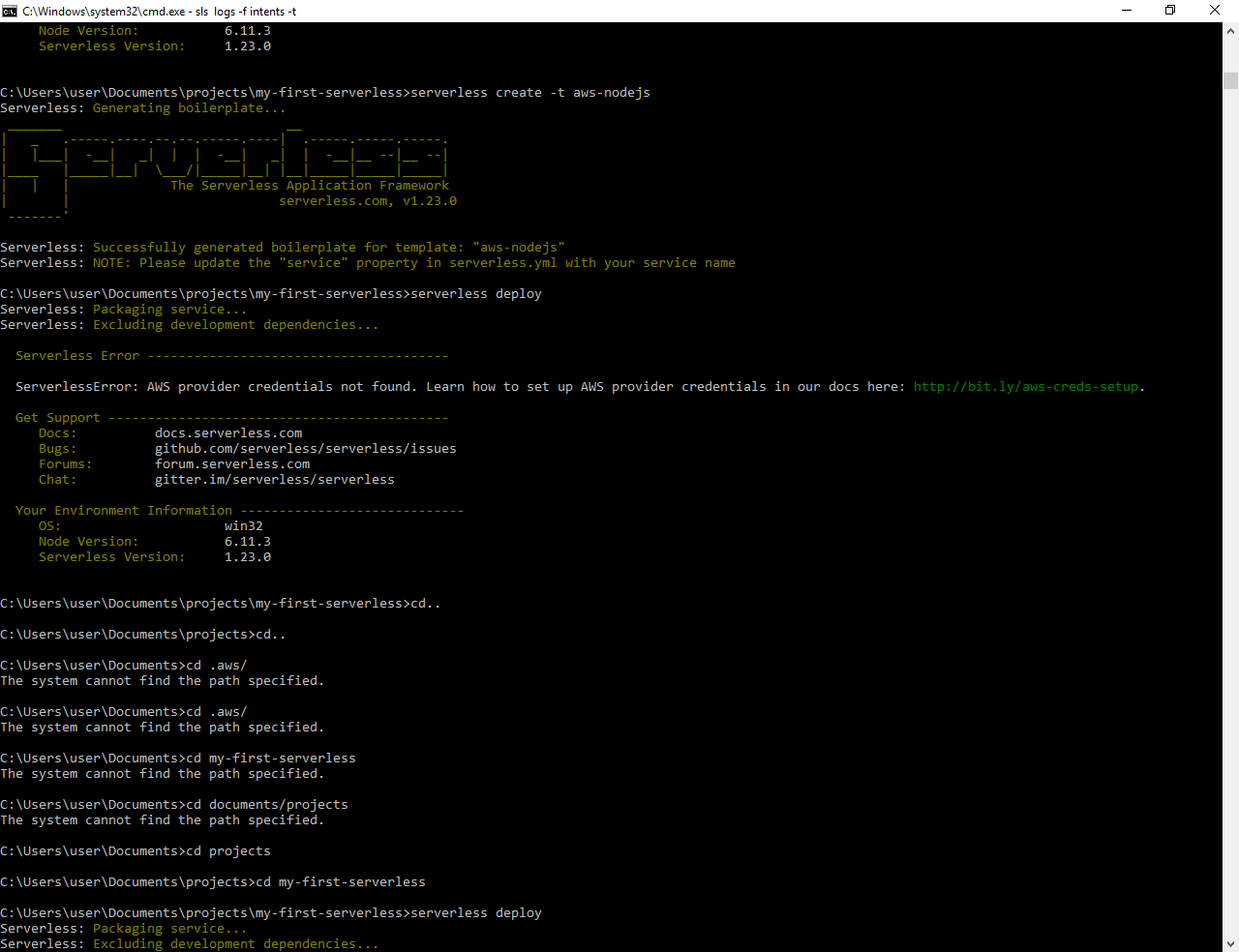
To see which version of serverless you have installed run:

serverless --version

**Create a new service**

Create a new service using the Node.js template.

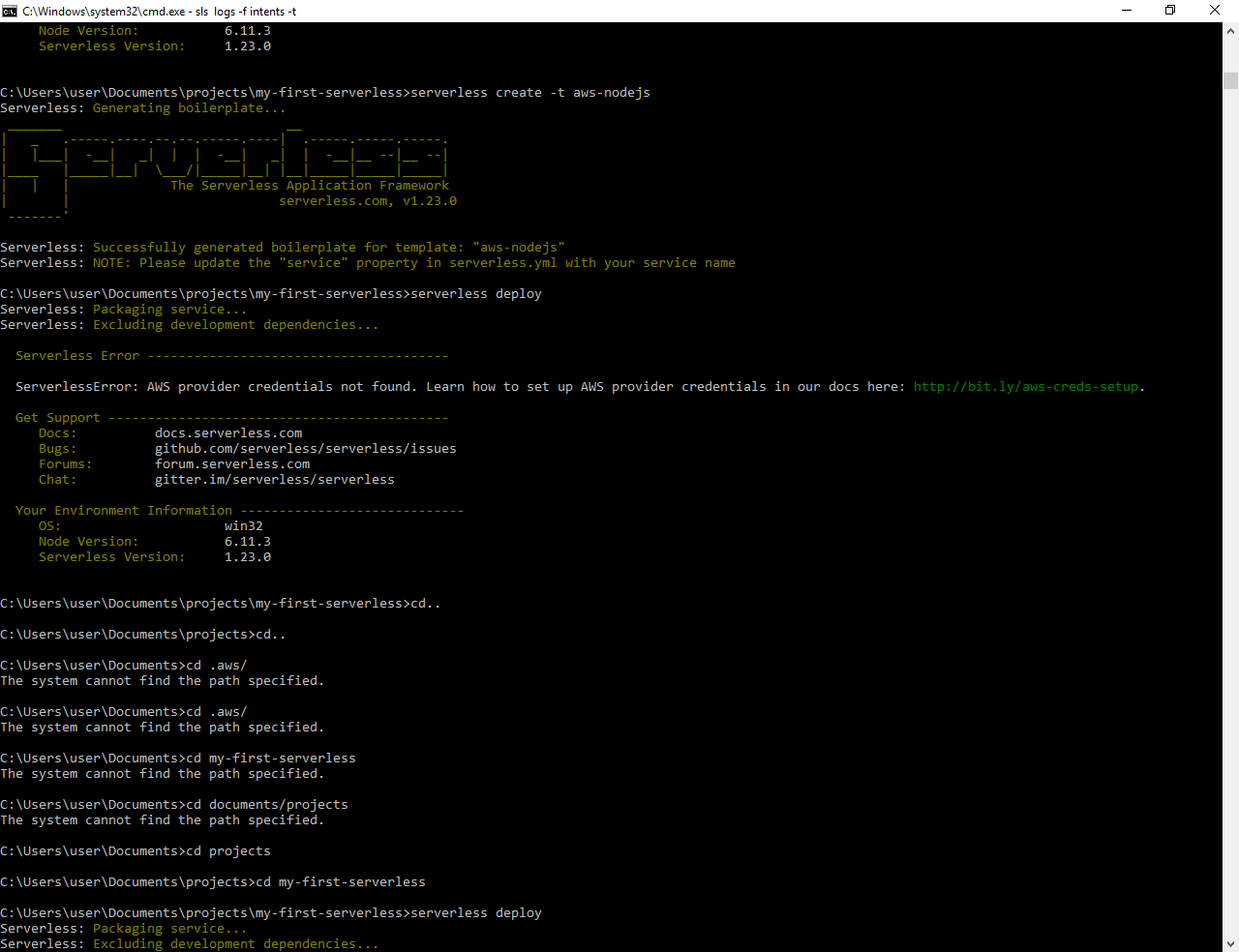
serverless create --template aws-nodejs



**Deploy the Service**

Use this when you have made changes to your Functions, Events or Resources in serverless.yml

serverless deploy



**AWS - Credentials**

The Serverless Framework needs access to your cloud provider's account so that it can create and manage resources on your behalf.

**Creating AWS Access Keys**

To let the Serverless Framework access your AWS account, we're going to **create an IAM User with Admin access**, which can configure the services in your AWS account. This IAM User will have its own set of AWS Access Keys.

Create or login to your Amazon Web Services Account and go to the Identity & Access Management (IAM) page.

Click on **Users** and then **Add user**. Enter a name in the first field to remind you this User is the Framework, like serverless-admin. Enable **Programmatic access** by clicking the checkbox. Click **Next** to go through to the Permissions page. Click on **Attach existing policies directly**. Search for and select **AdministratorAccess** then click **Next: Review**. Check everything looks good and click **Create user**. Later, you can create different IAM Users for different apps and different stages of those apps. That is, if you don't use separate AWS accounts for stages/apps, which is most common.

View and copy the **API Key** & **Secret** to a temporary place. You'll need it in the next step.

### Using AWS Access Keys

Setup with serverless config credentials command

Serverless provides a convenient way to configure AWS profiles with the help of the serverless config credentials command.

Here's an example how you can configure the default AWS profile:

serverless config credentials --provider aws --key AKIAIOSFODNN7EXAMPLE --secret wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

### Now you can deploy the service :

serverless deploy

