**AWS Pre-requisite for Using AWS Lambda with Node and React**

1. Who is this guide for?

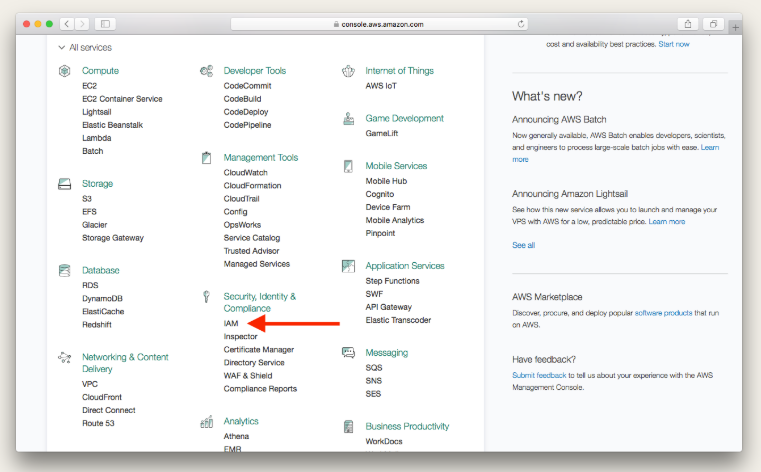
This guide is meant for full-stack developers or developers that would like to build full stack serverless applications. By providing a step-by-step guide for both the frontend and the backend. This guide is meant to serve as a resource for learning about how to build and deploy serverless applications, as opposed to laying out the best possible way of doing so.

1. Create an AWS Account

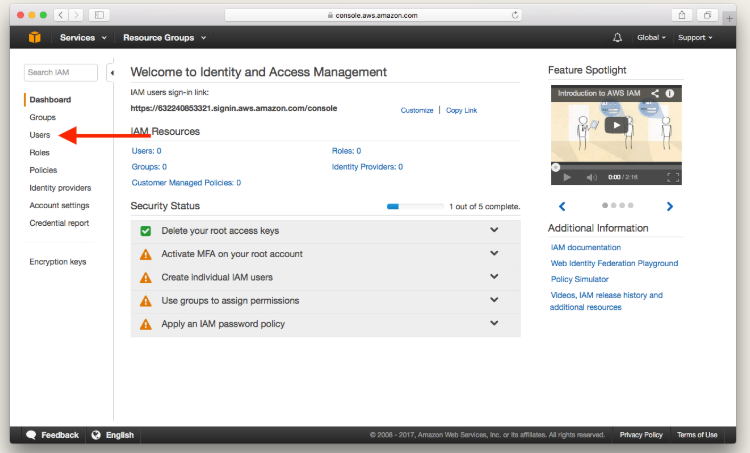
* Go to the following URL and create your amazon account-https://aws.amazon.com/console/

1. Create an IAM User

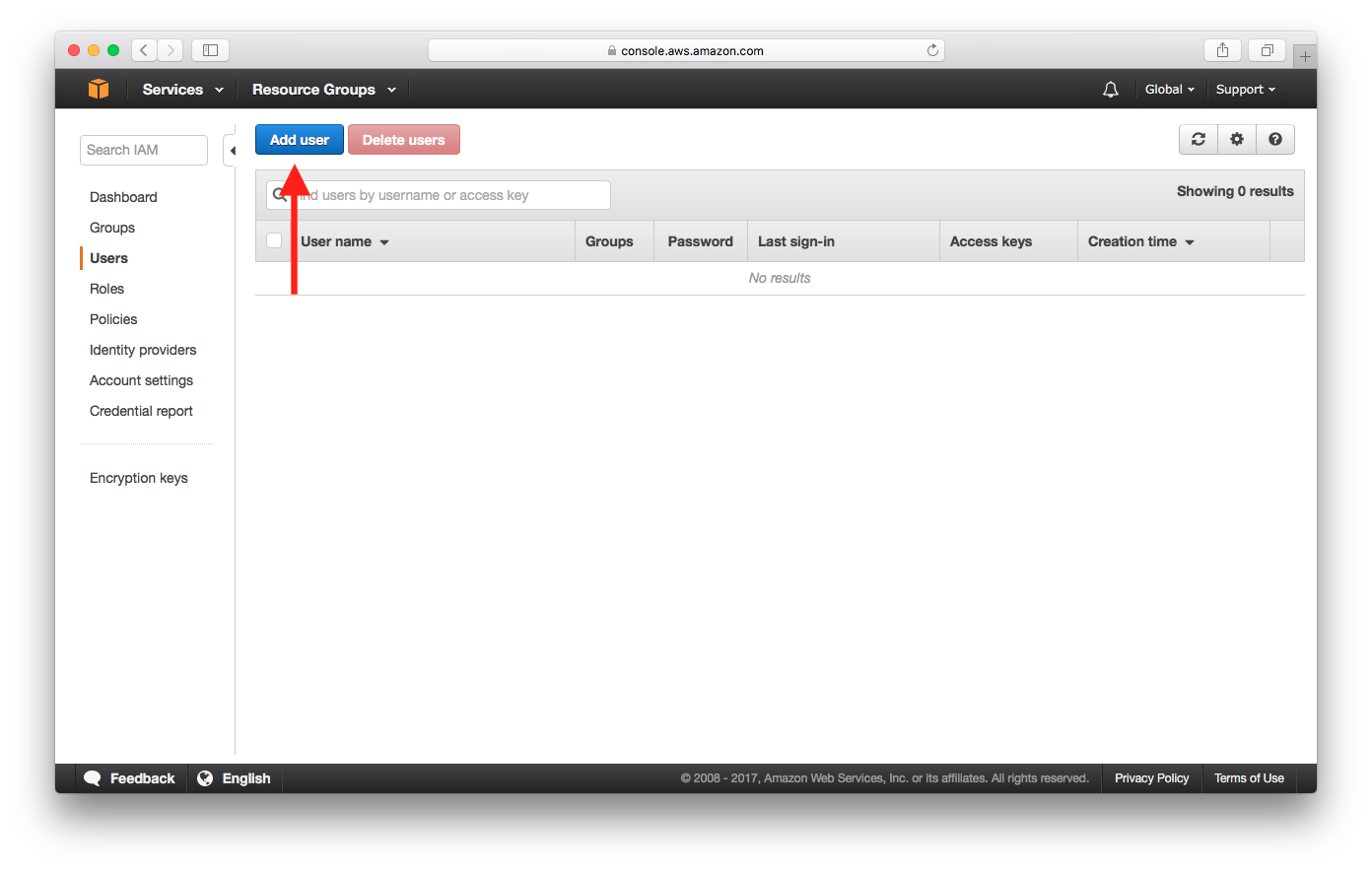
* First, log in to your [AWS Console](https://console.aws.amazon.com/) and select IAM from the list of services.



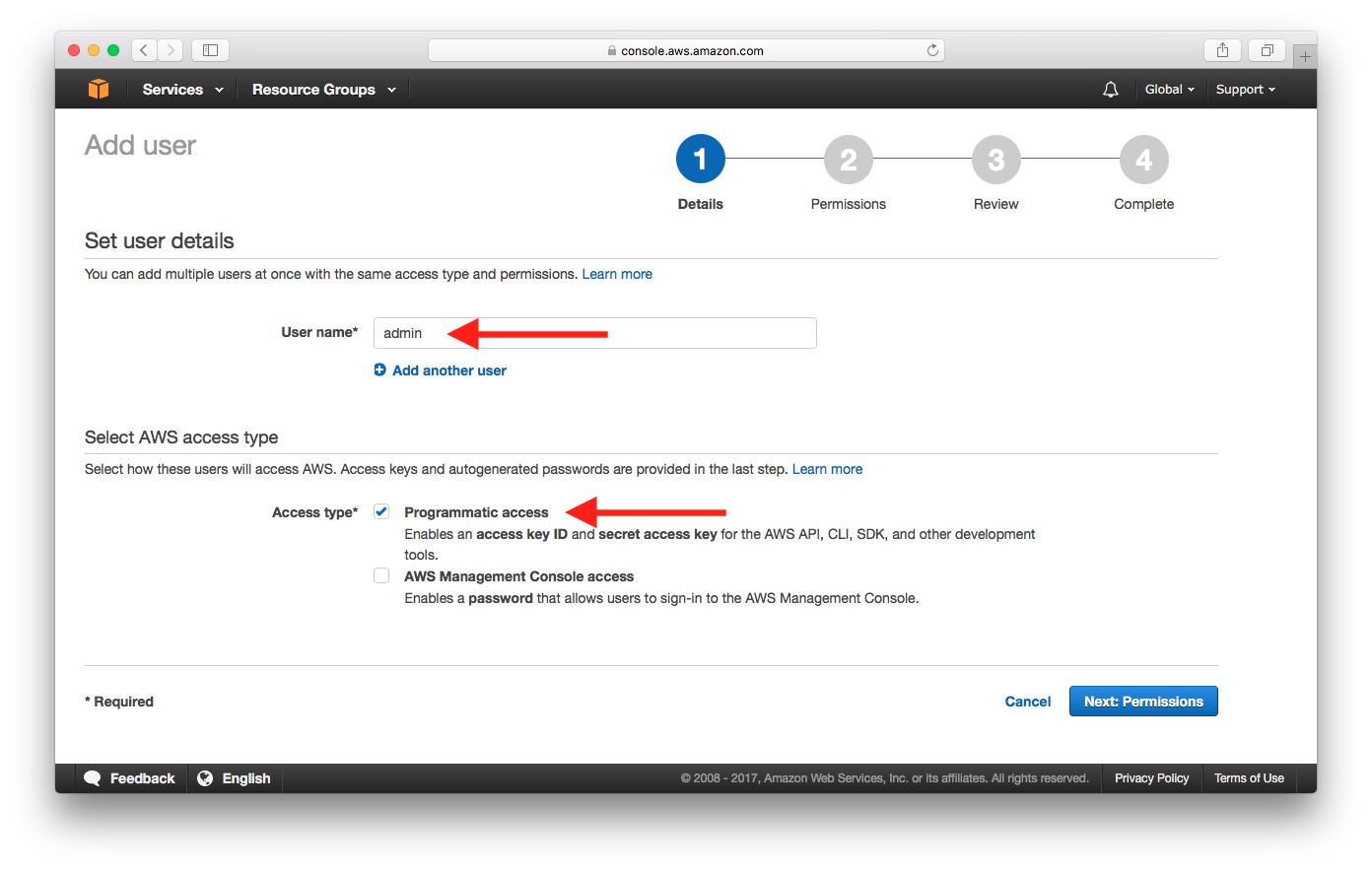
* Select **Users**.



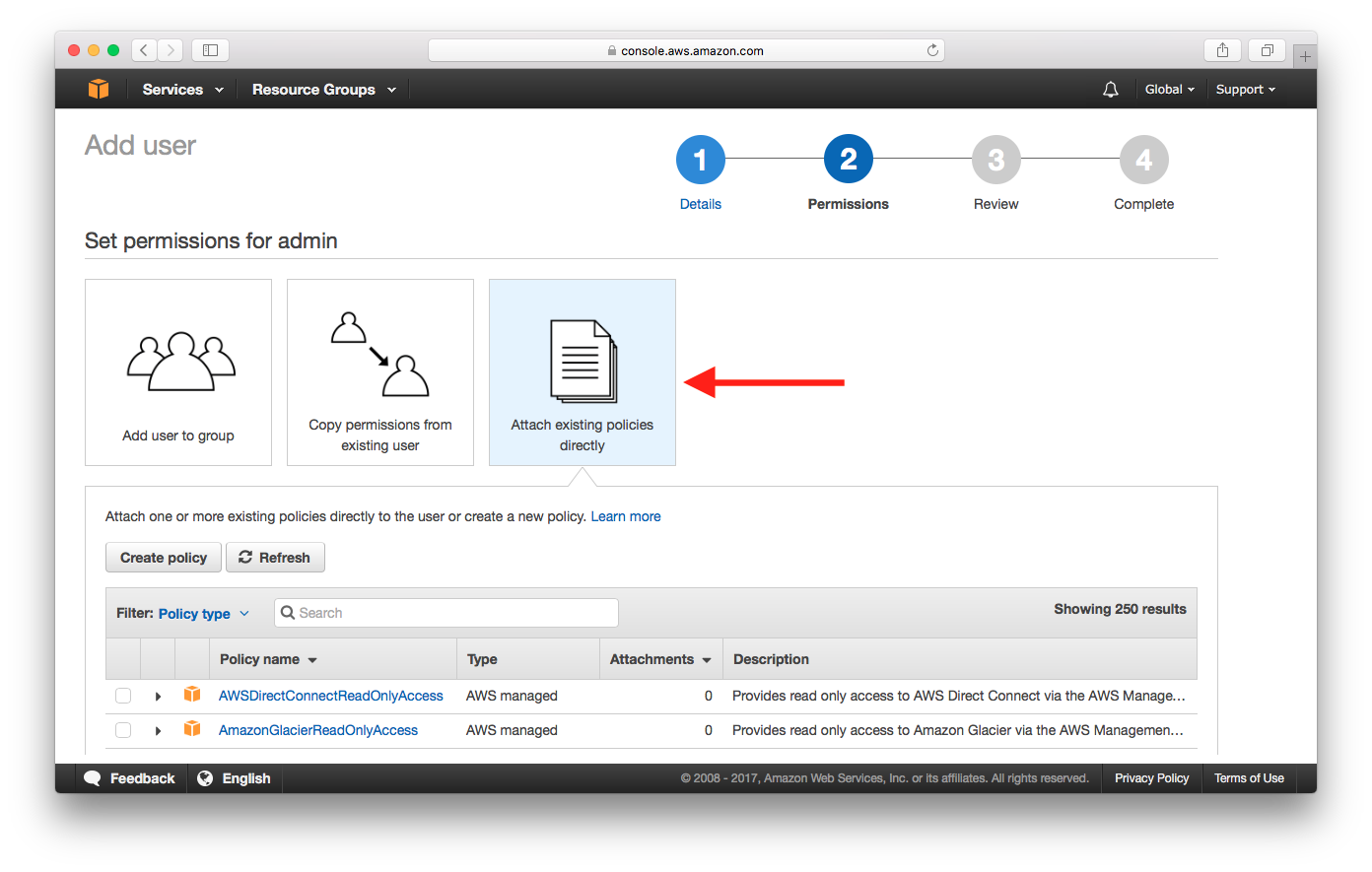
* Select **Add User**.



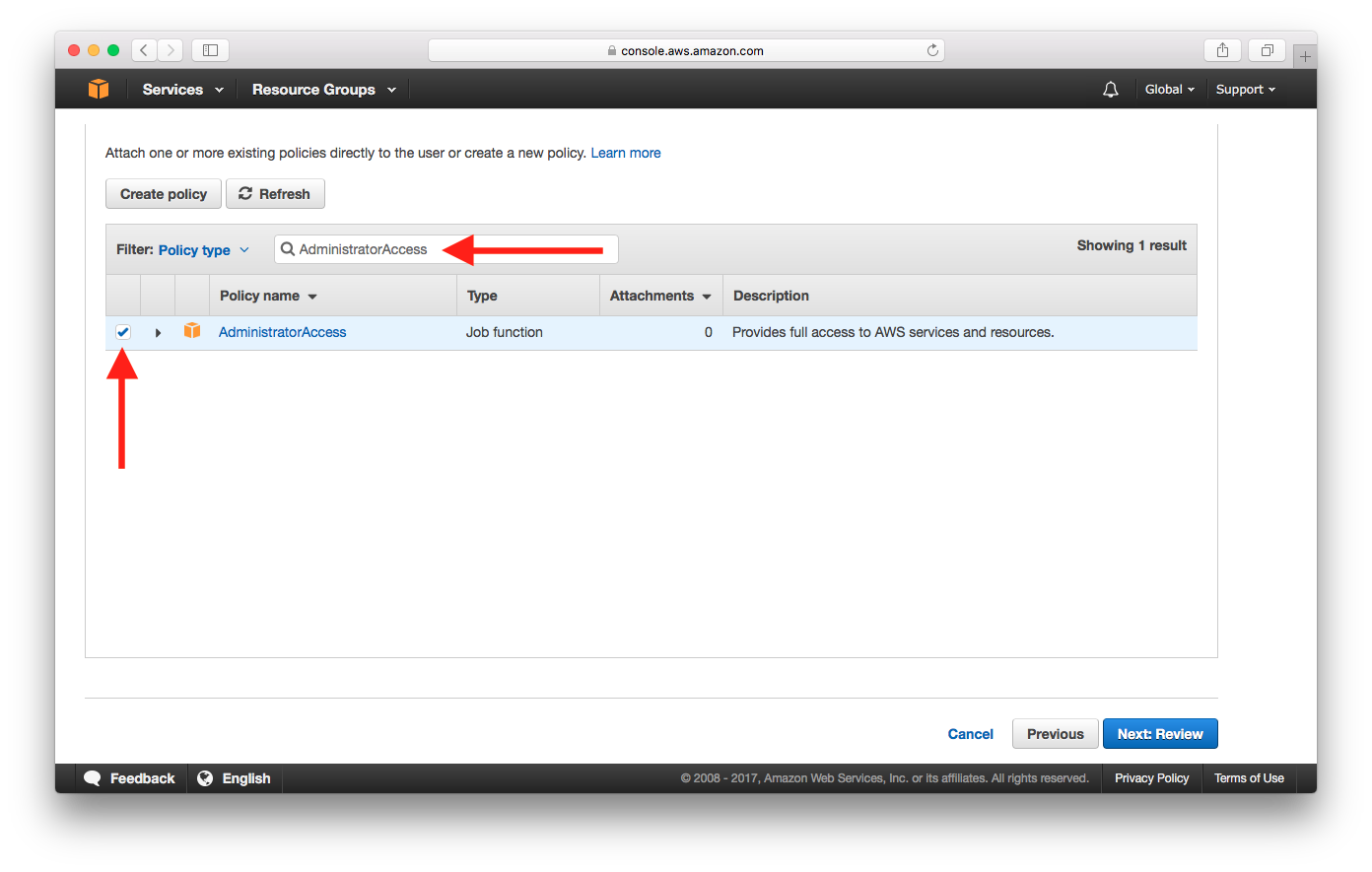
* Enter a **User name** and check **Programmatic access**, then select **Next: Permissions**.
* This account will be used by our [AWS CLI](https://aws.amazon.com/cli/) and [Serverless Framework](https://serverless.com/). They’ll be connecting to the AWS API directly and will not be using the Management Console.



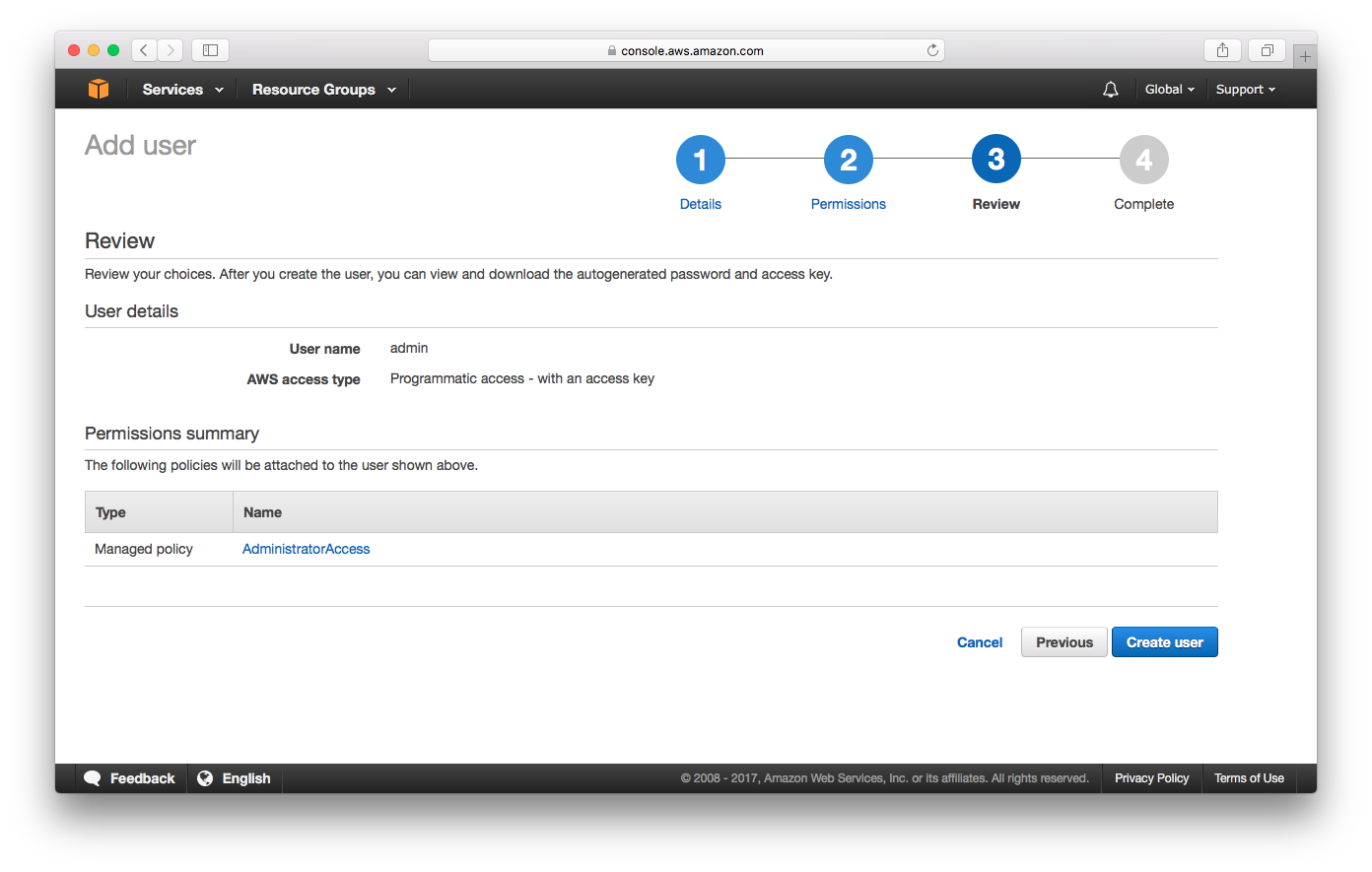
* Select **Attach existing policies directly**.



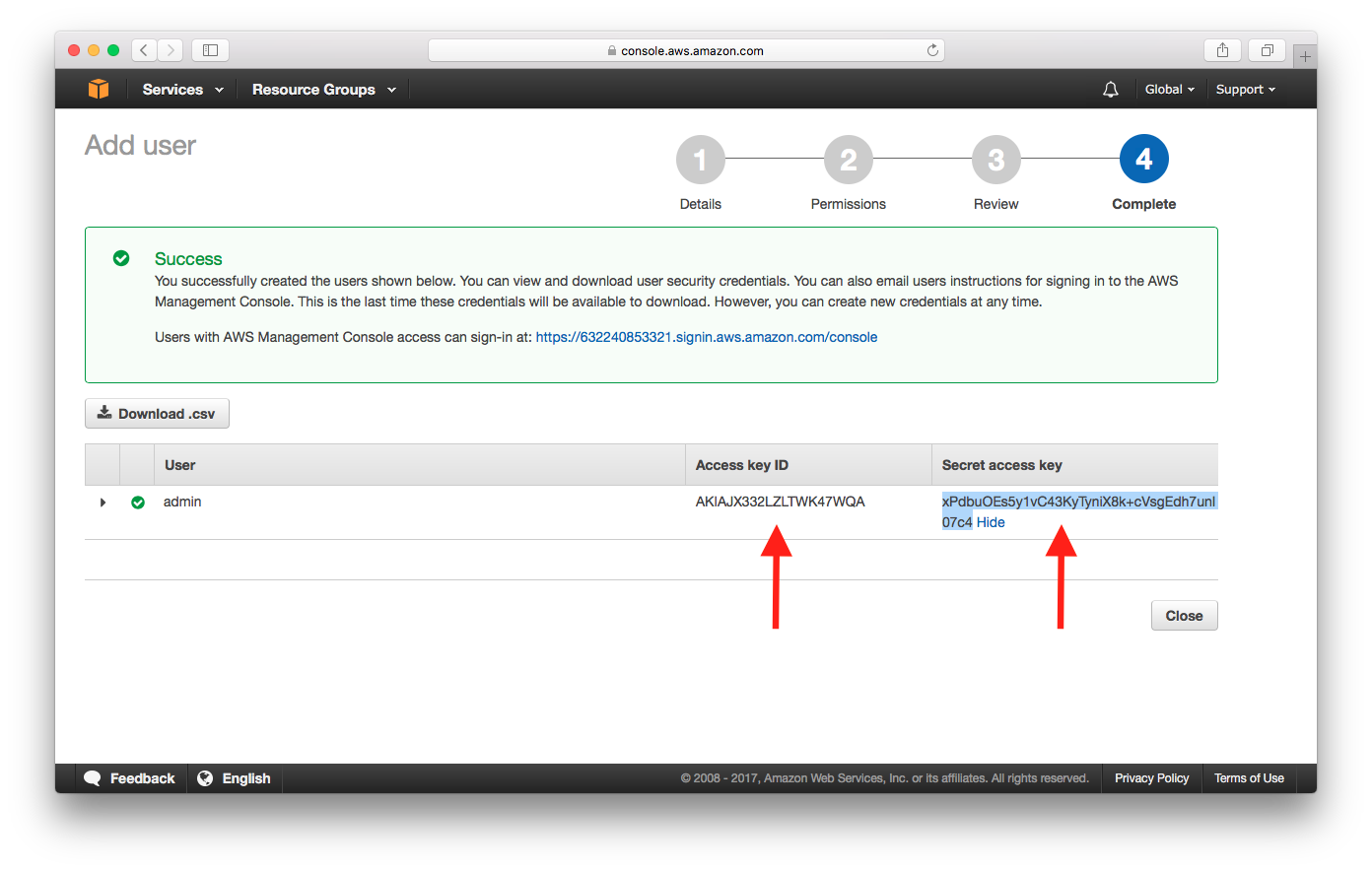
* Search for **AdministratorAccess** and select the policy, then select **Next: Review**.



* Select **Create user**.

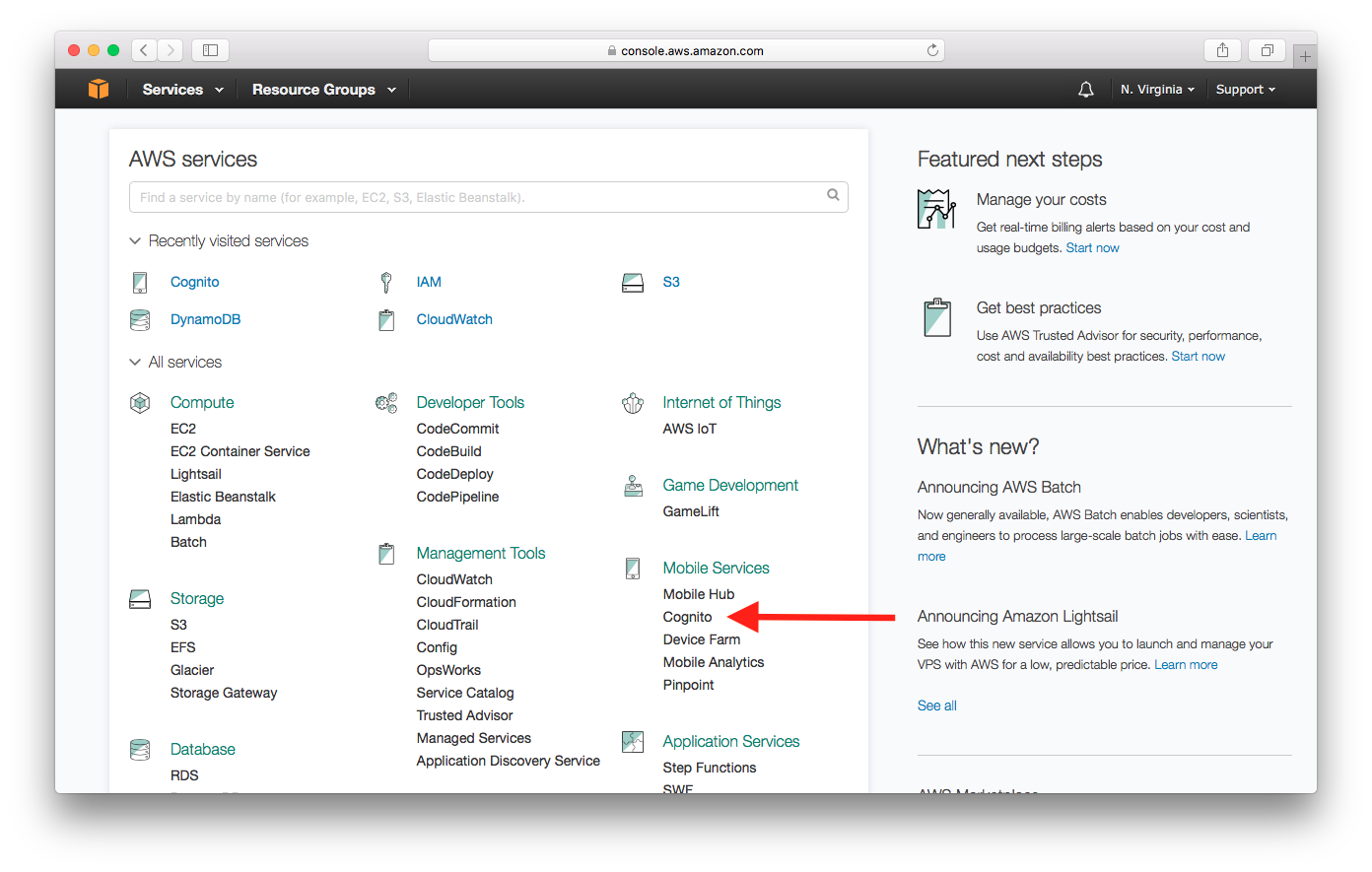


* Select **Show** to reveal **Secret access key**.
* Take a note of the **Access key ID** and **Secret access key**. We will be needing this later.

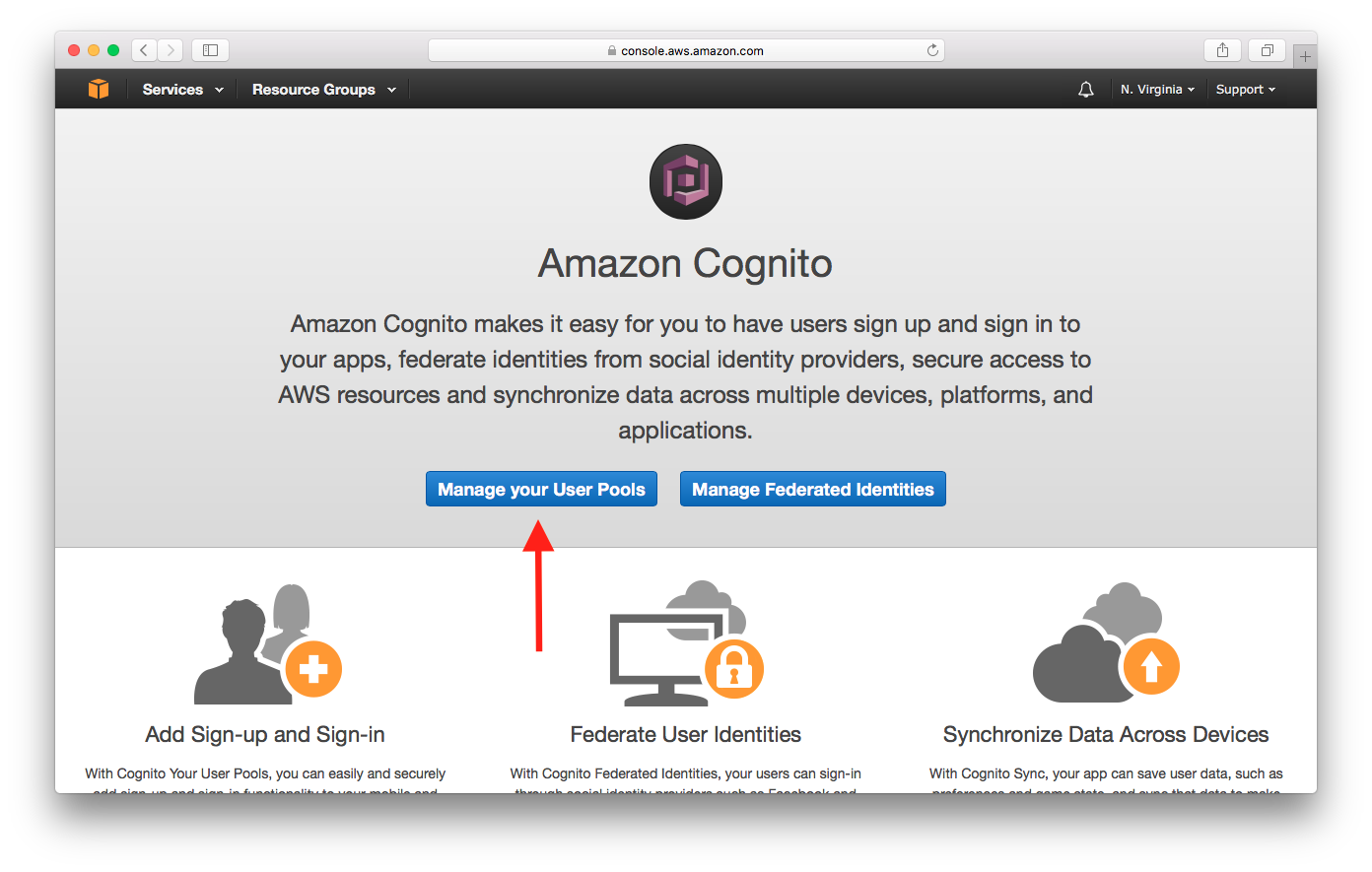


1. Create Cognito Pool

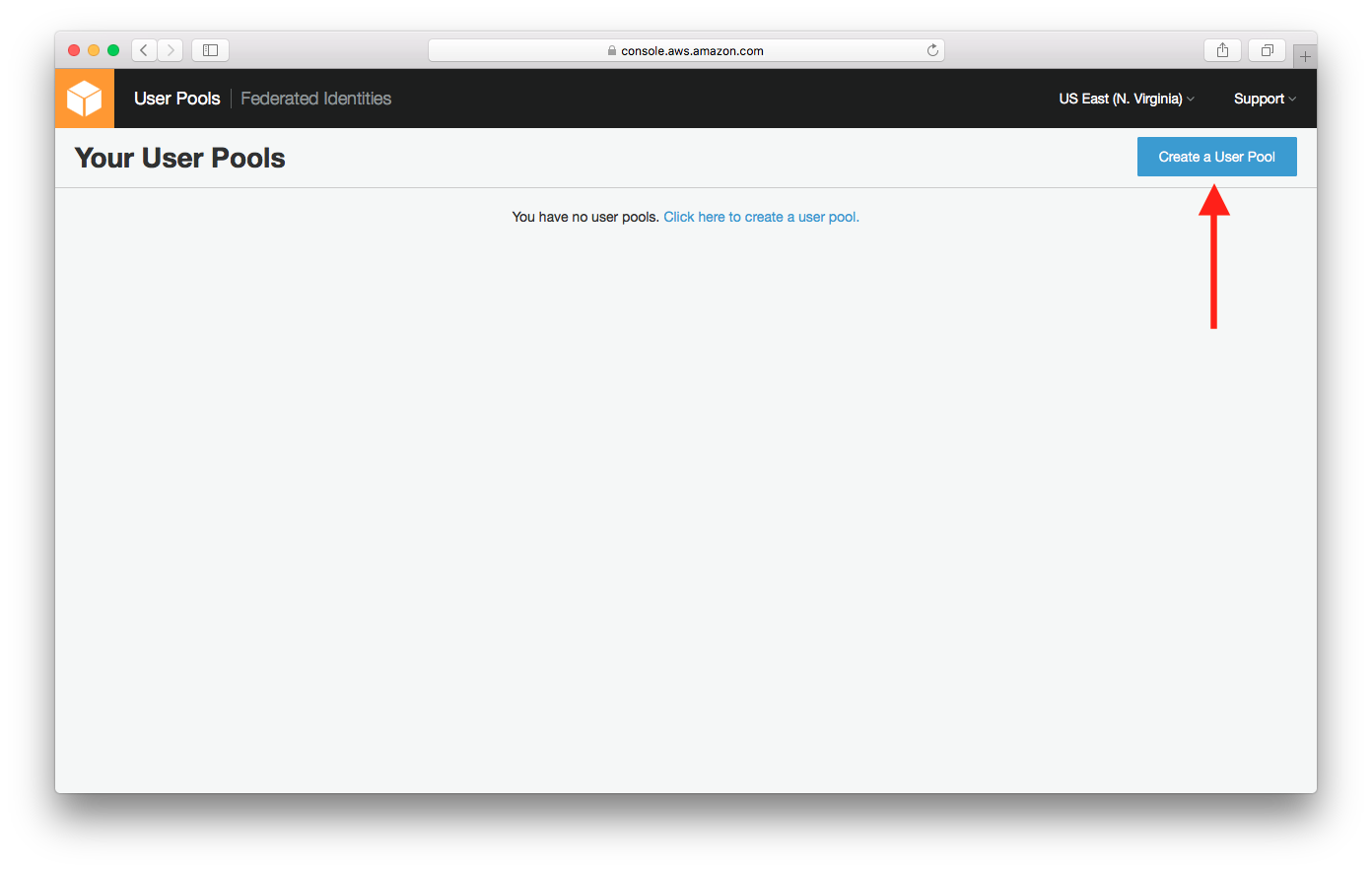
* From the following url: <https://console.aws.amazon.com/console/home?region=us-east-1> , select **Cognito** from the list of services.



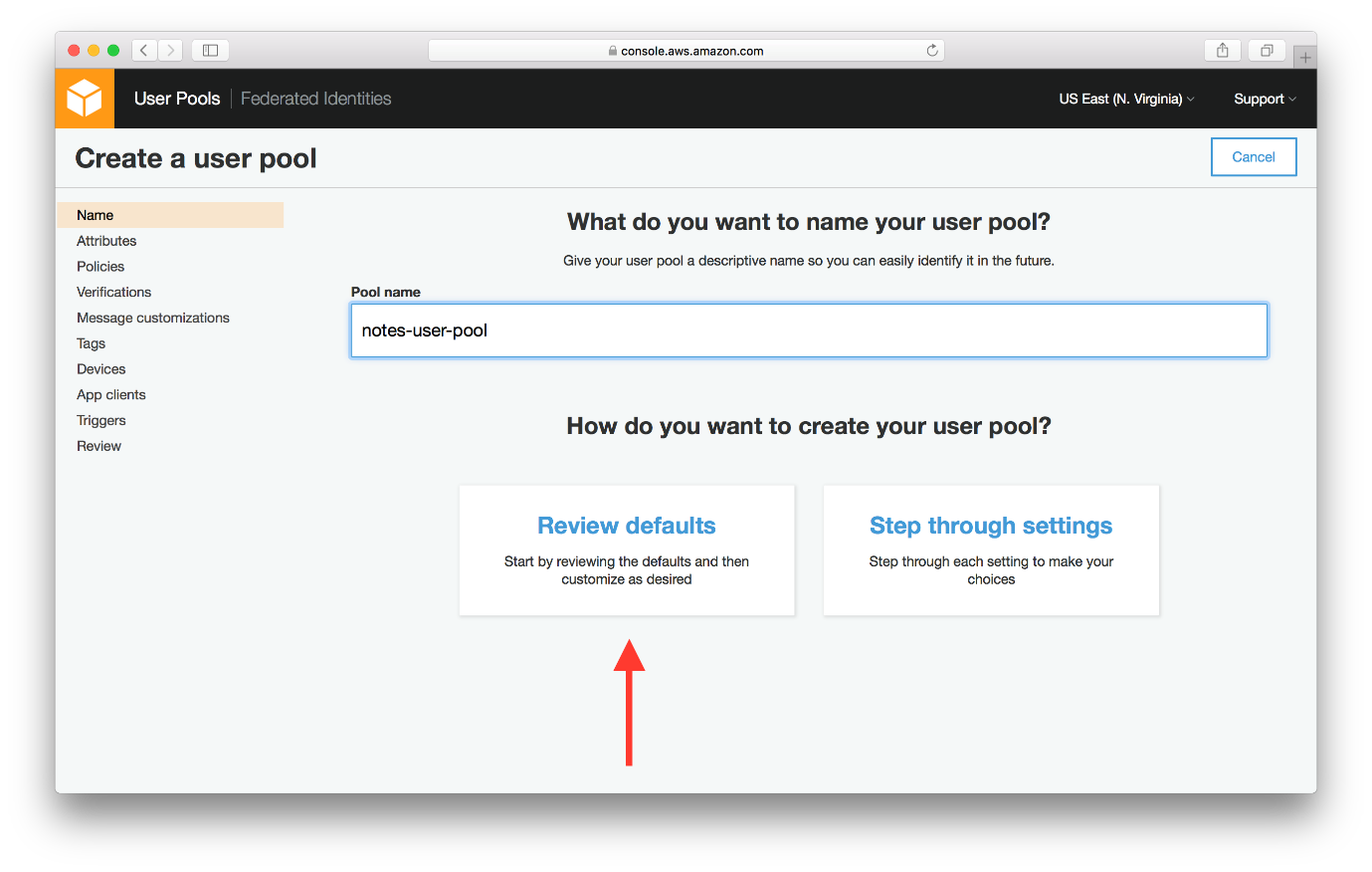
* Select **Manage your User Pools**.



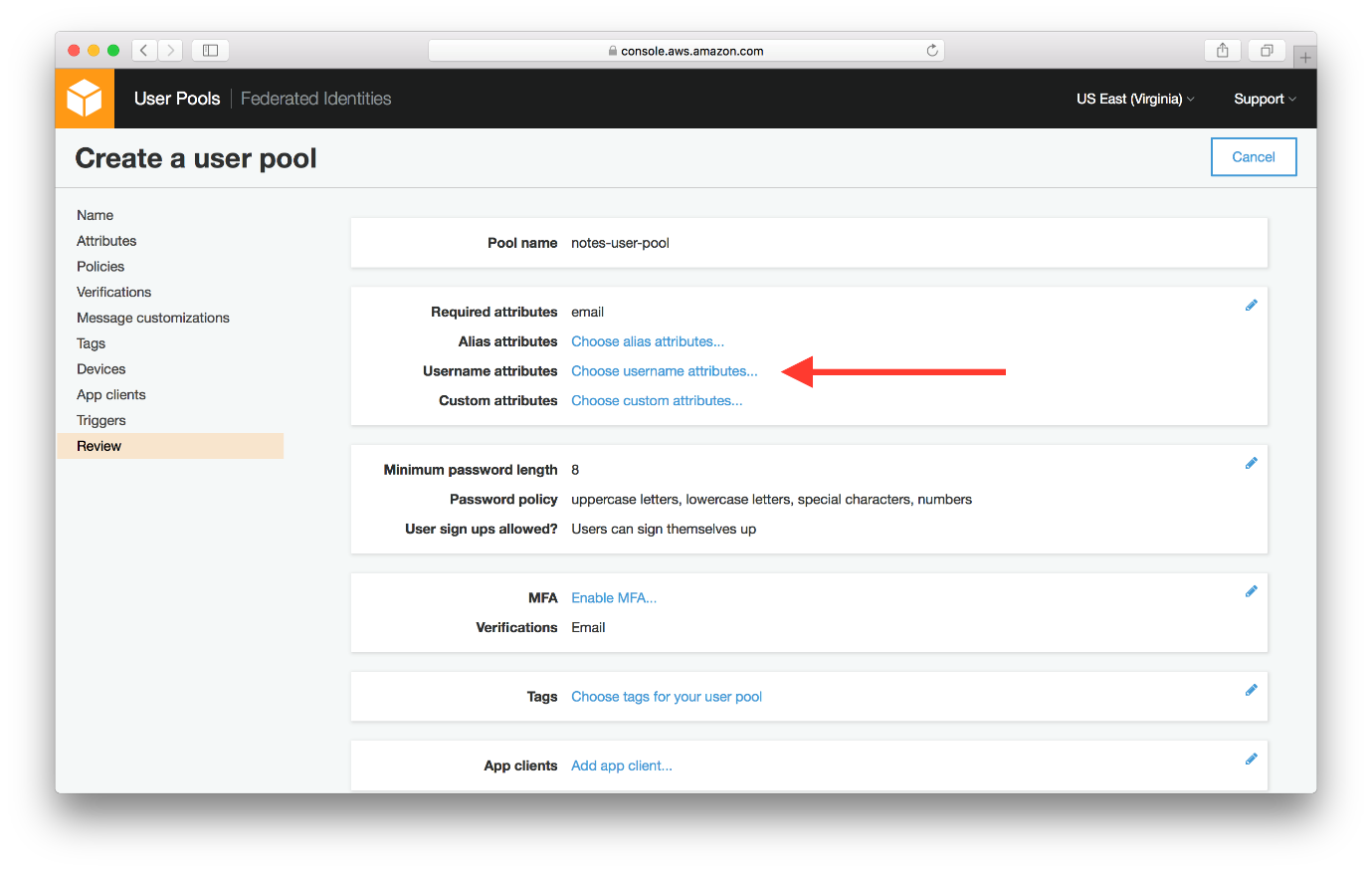
* Select **Create a User Pool**.



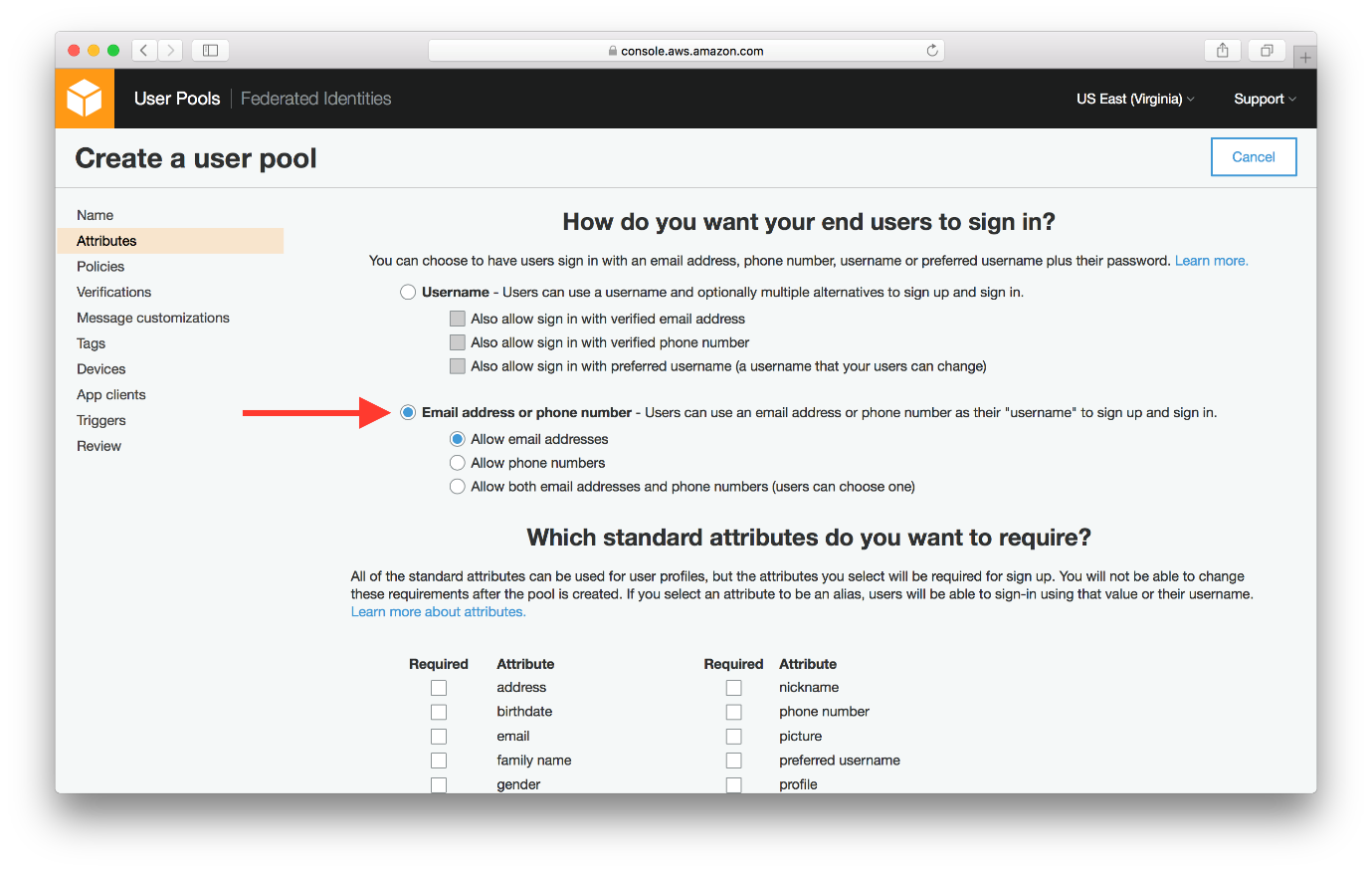
* Enter **Pool name** and select **Review defaults**.



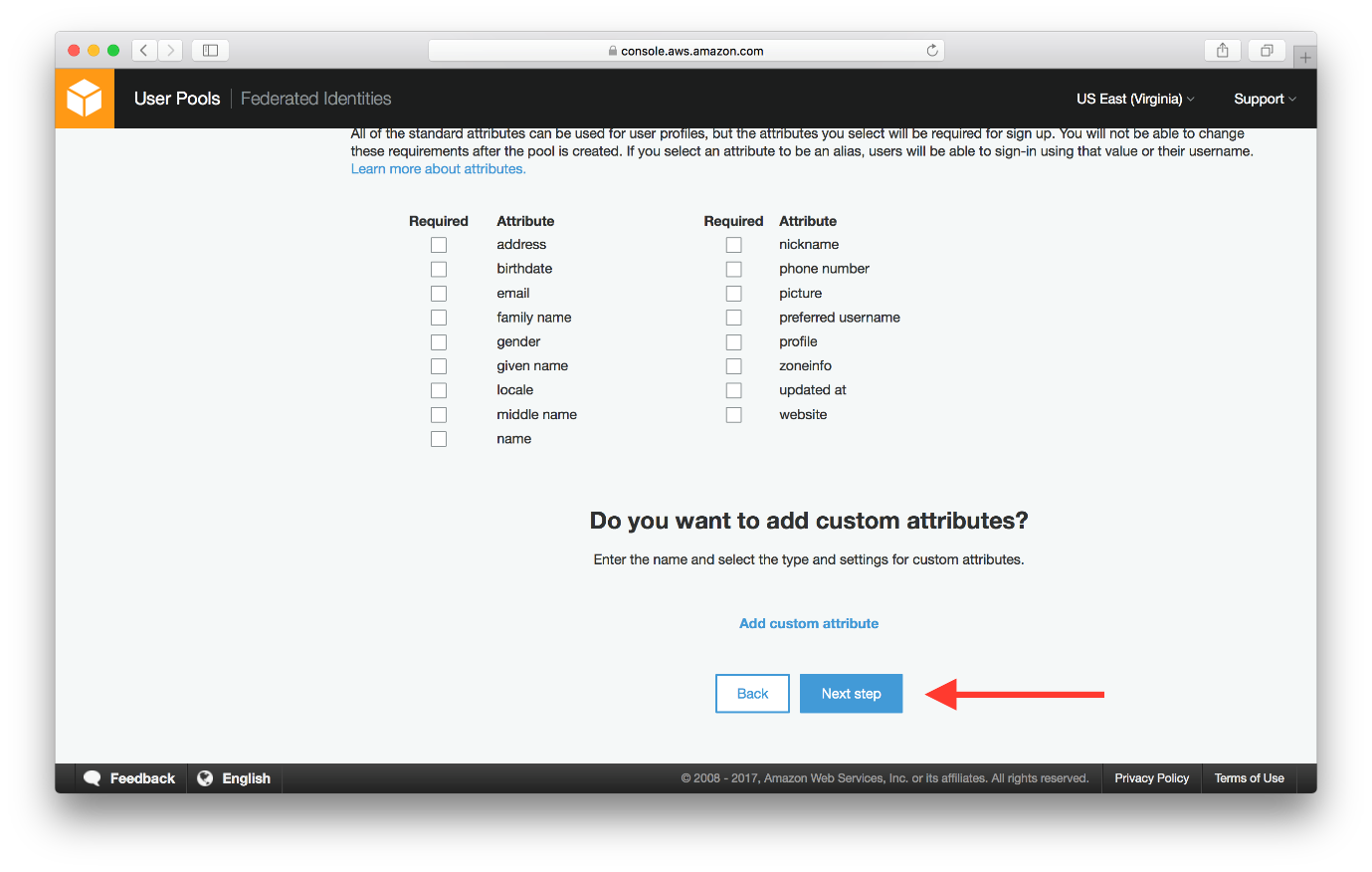
* Select **Choose username attributes…**.



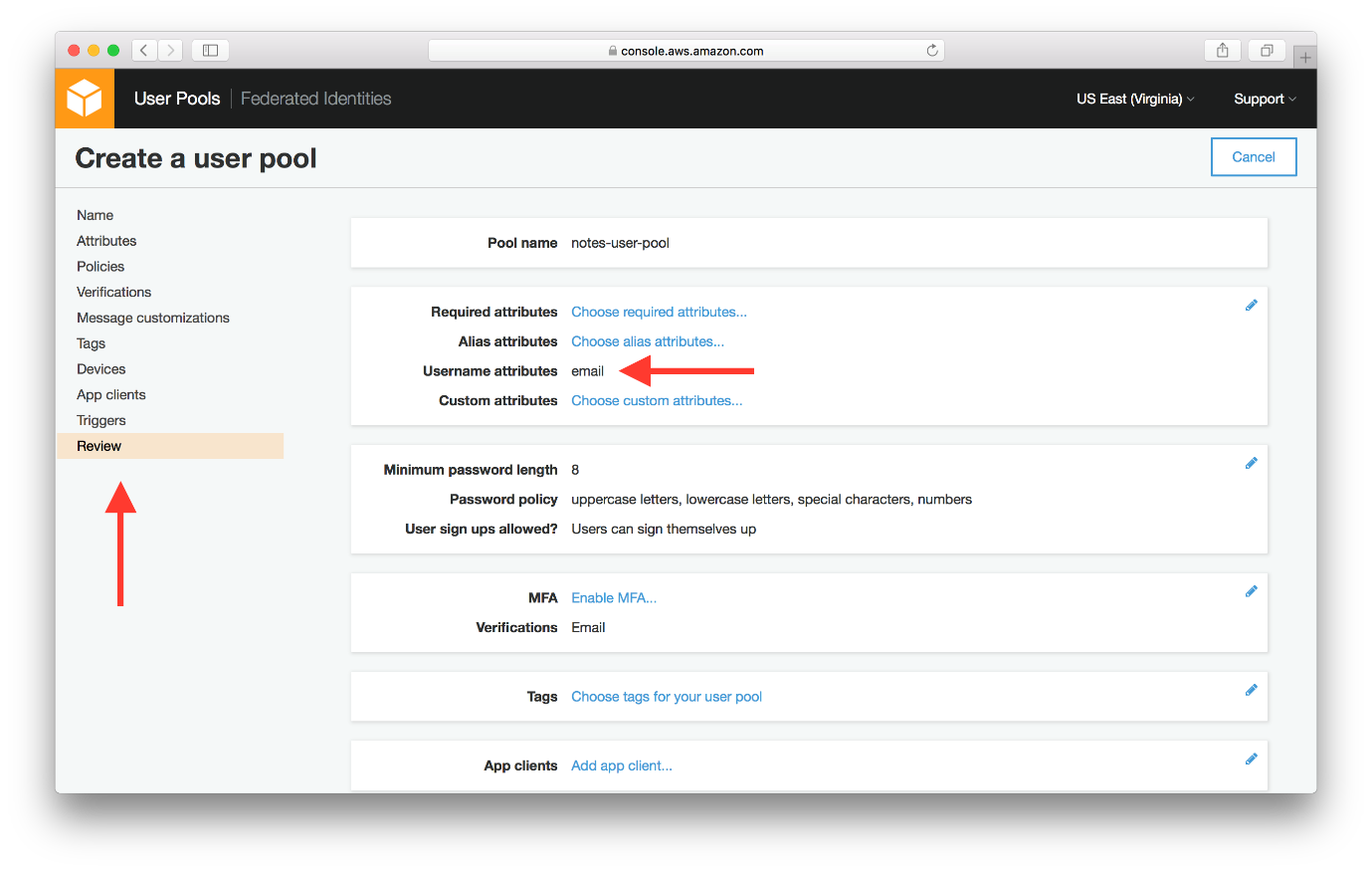
* And select **Email address or phone numbers** and **Allow email addresses**. This is telling Cognito User Pool that we want our users to be able to sign up and login with their email as their username.



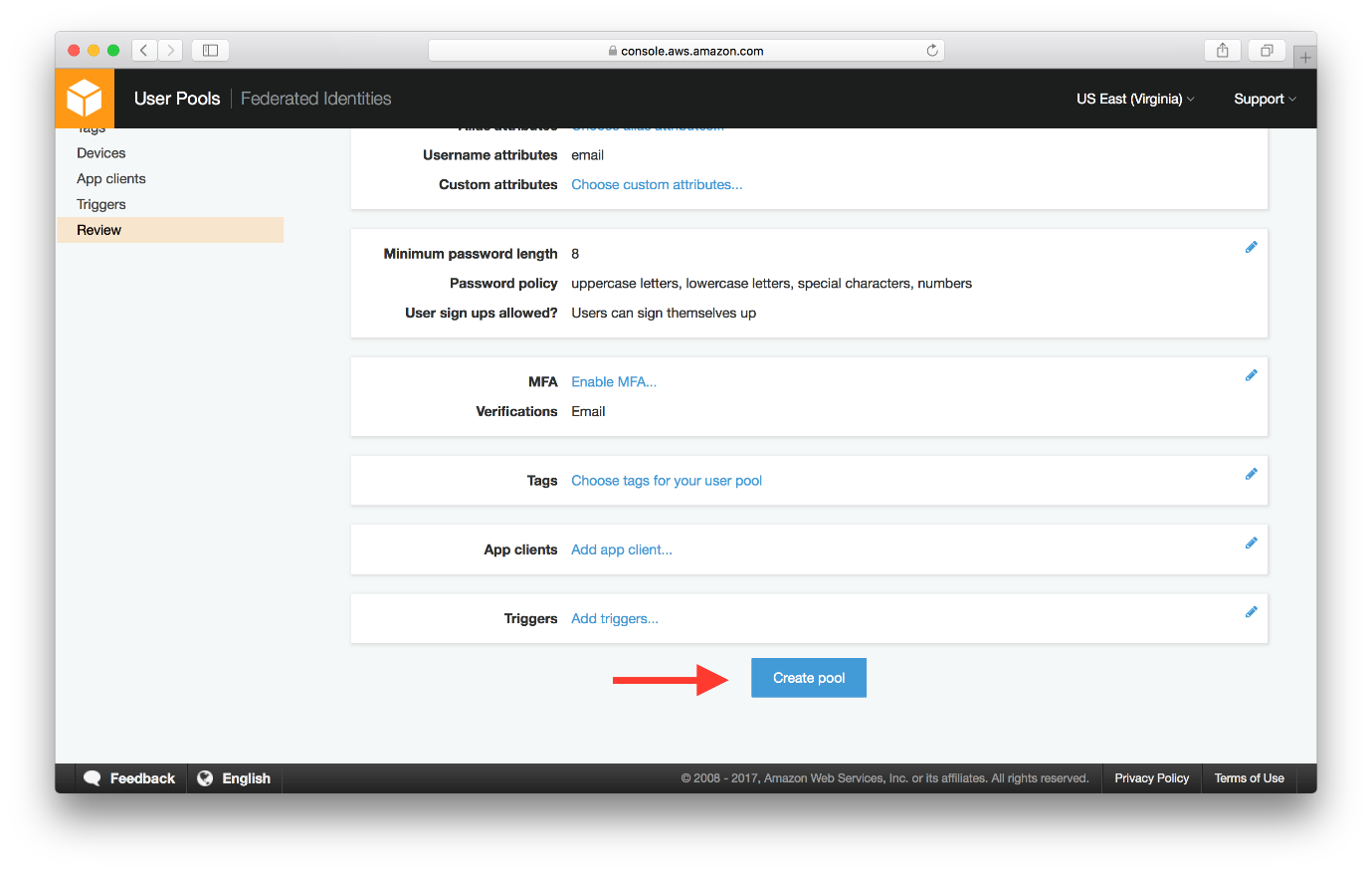
* Scroll down and select **Next step**.



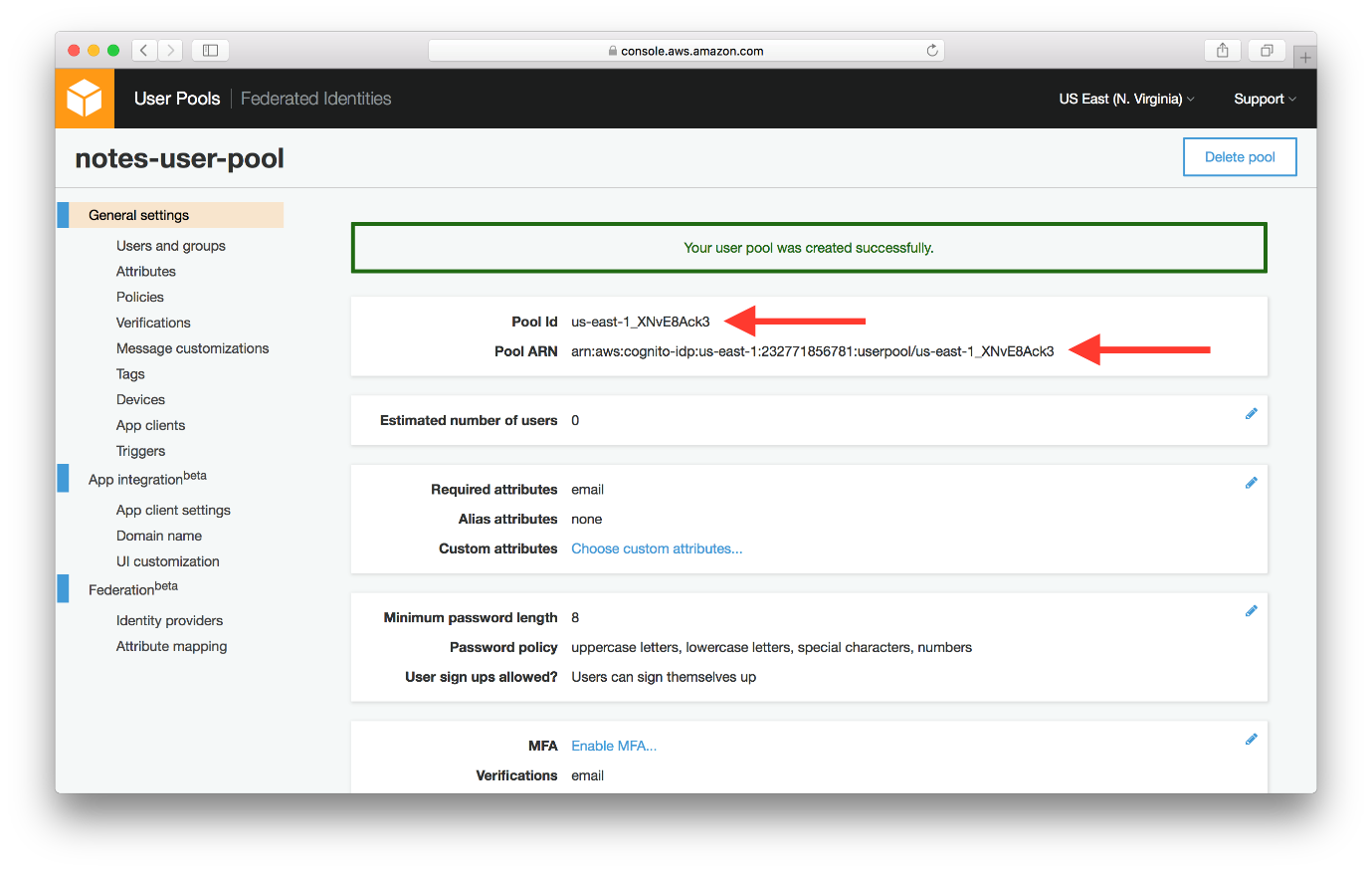
* Hit **Review** in the side panel and make sure that the **Username attributes** is set to **email**.



* Now hit **Create pool** at the bottom of the page.

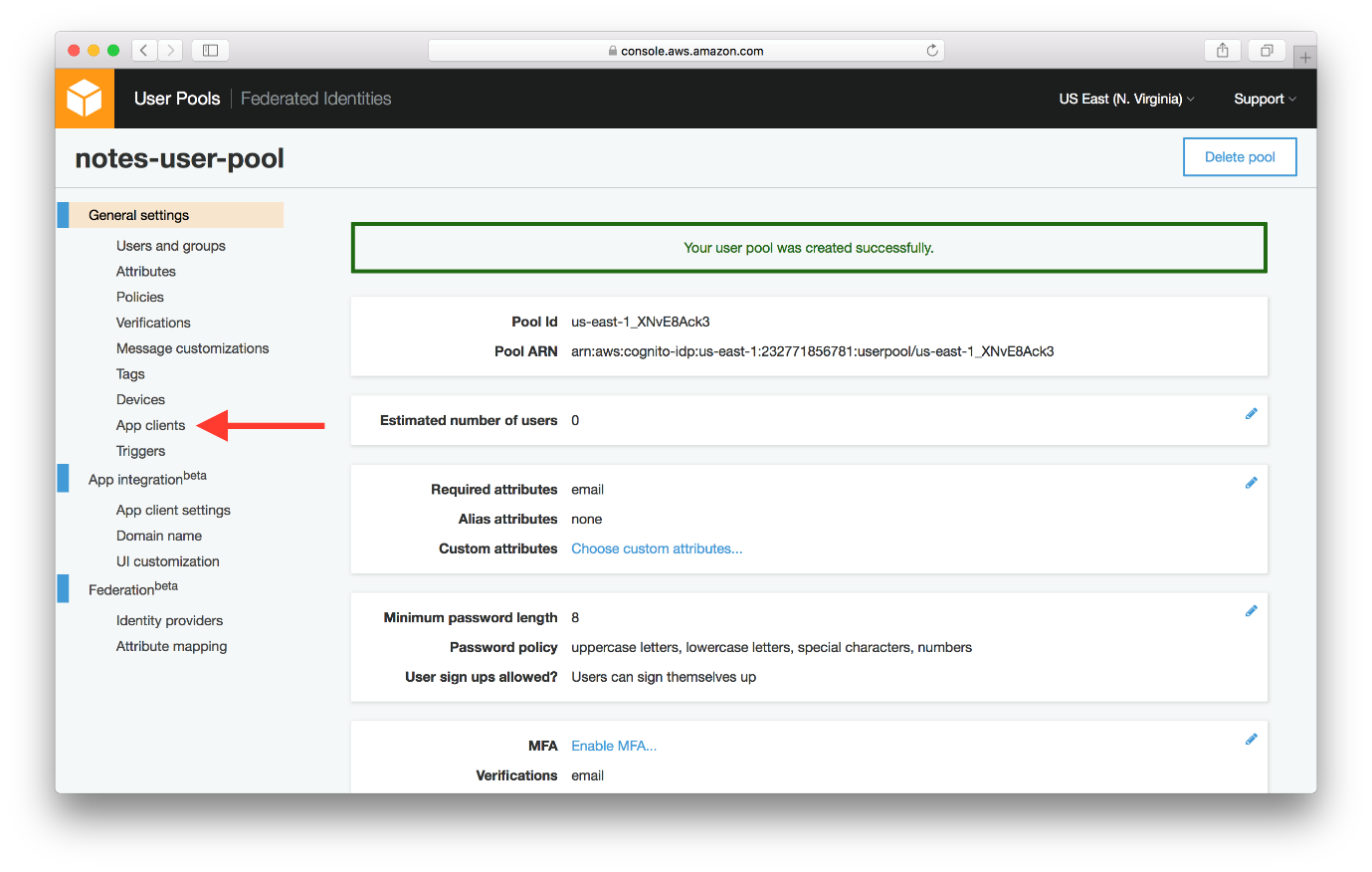


* Your User Pool has been created. Take a note of the **Pool Id** and **Pool ARN** which will be required later. Also, note the region that your User Pool is created in

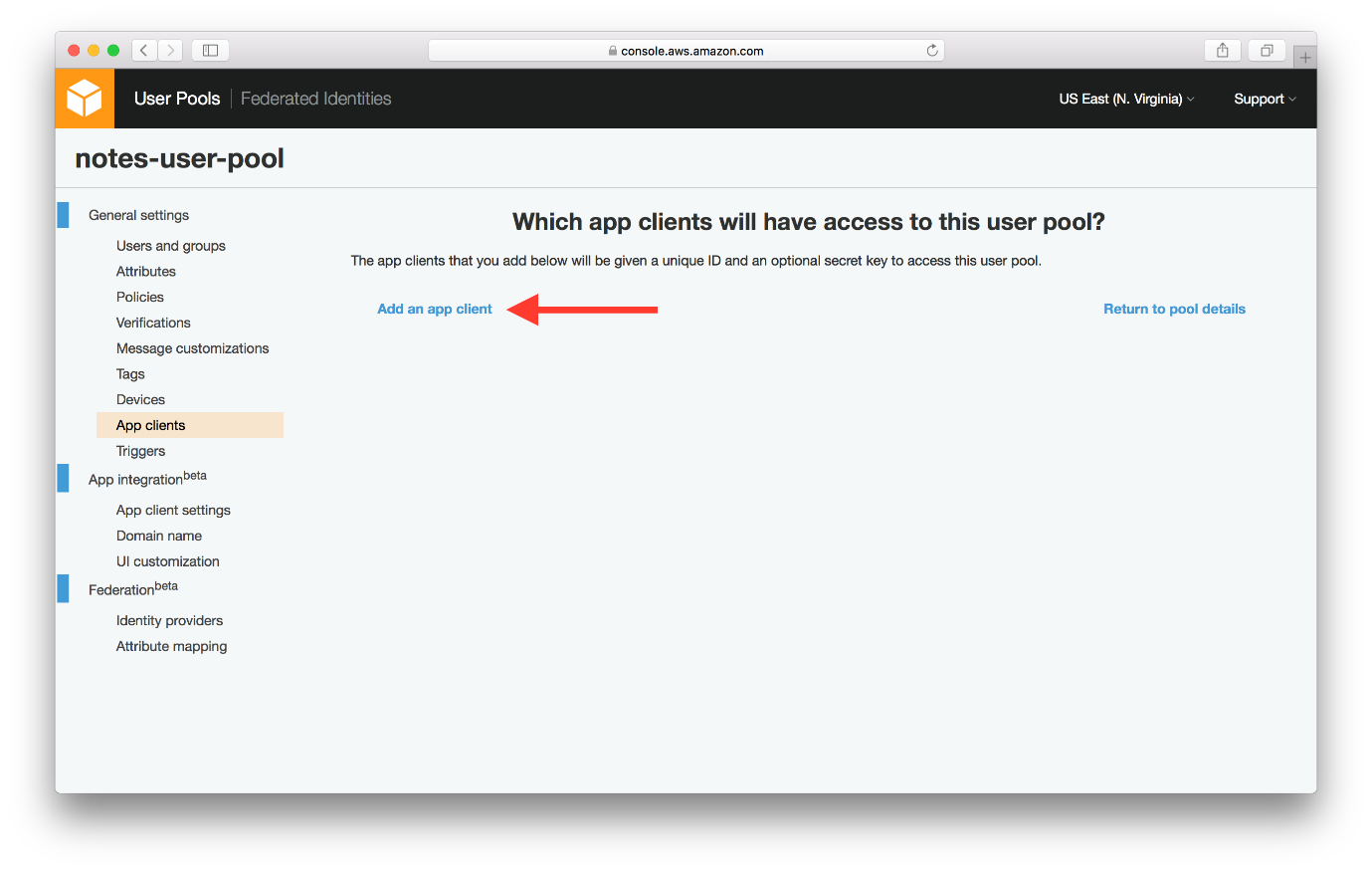


1. Create App Client

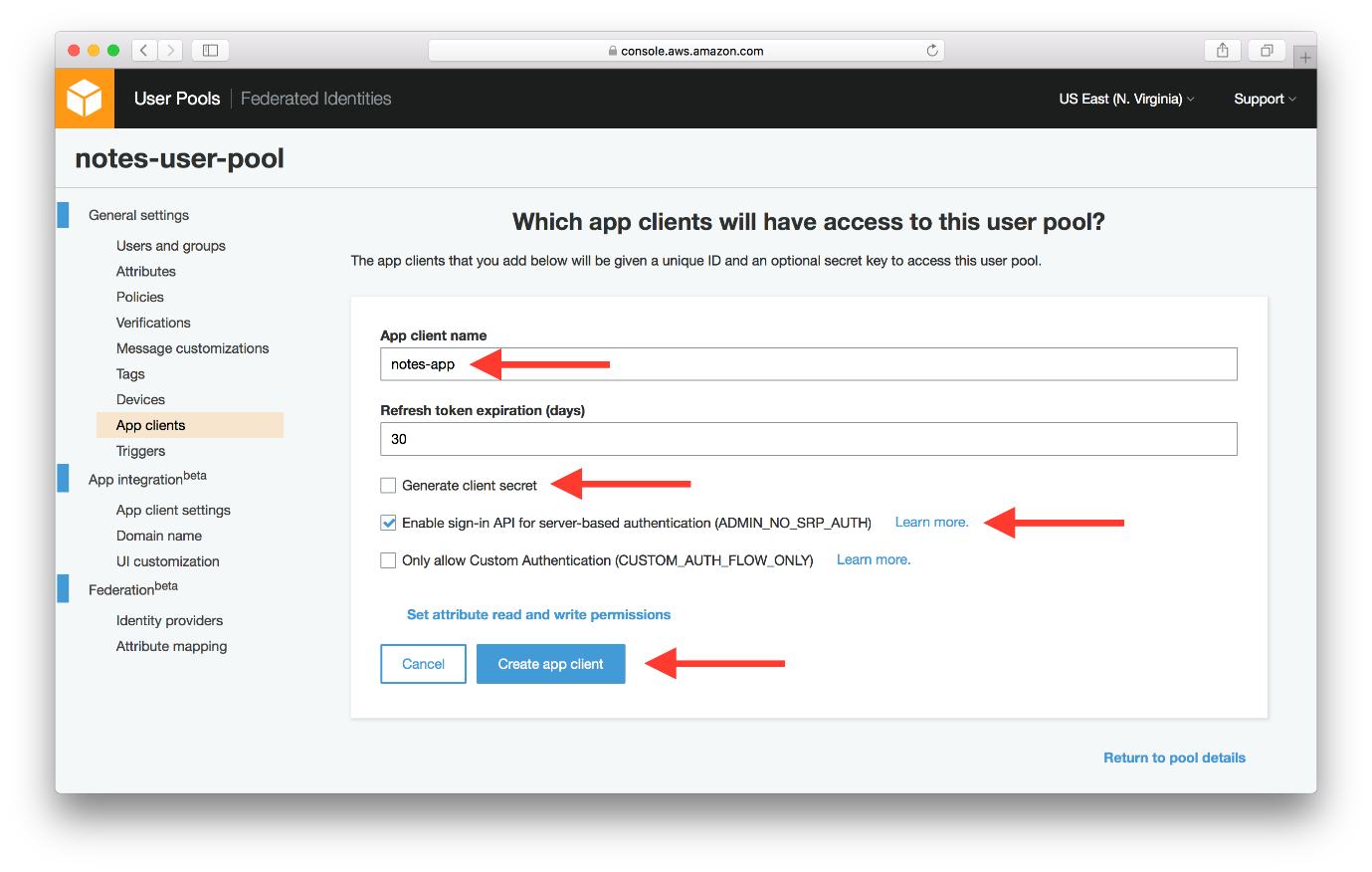
* Select **App clients** from the left panel.



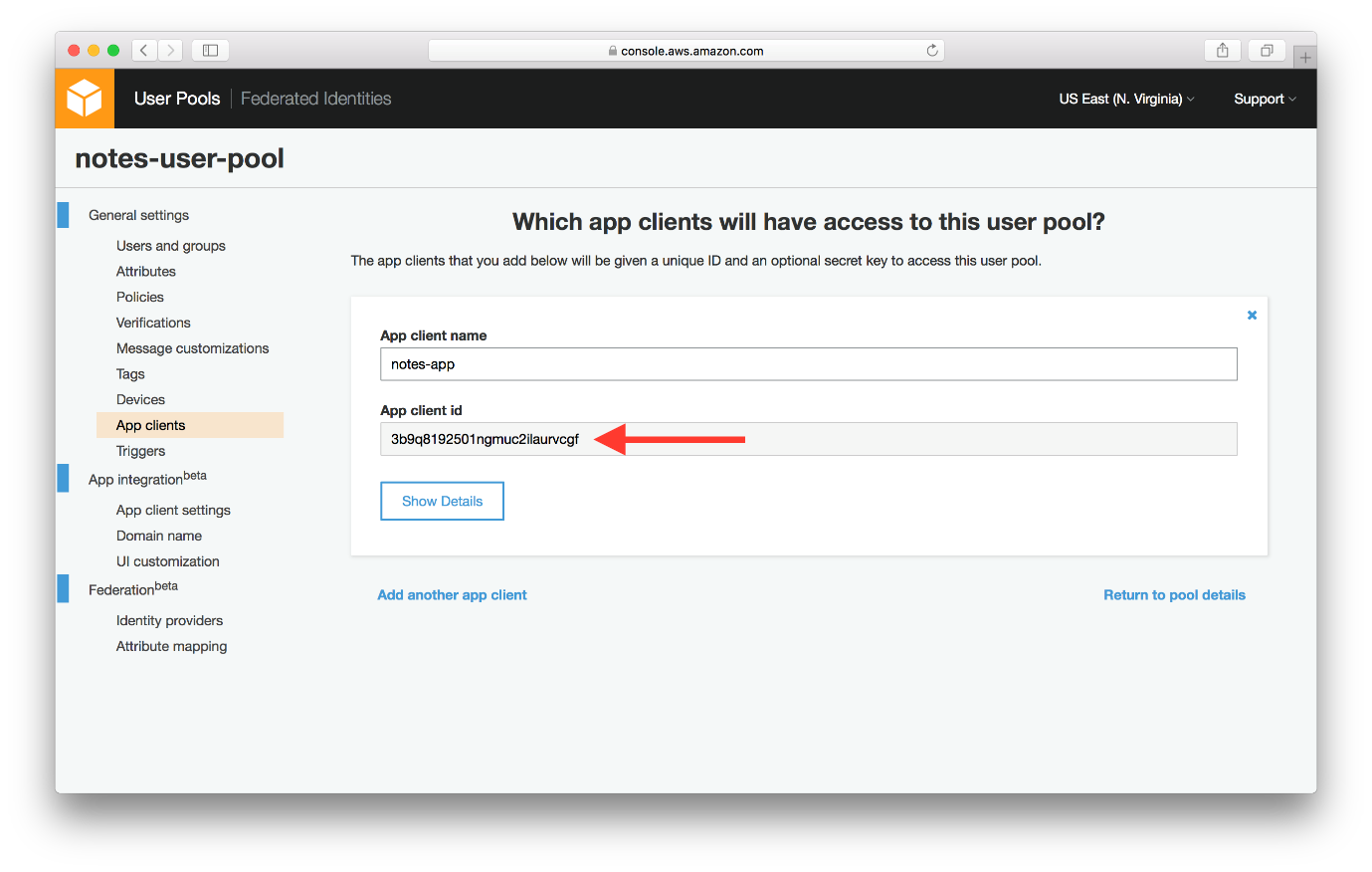
* Select **Add an app client**.



* Enter **App client name**, un-select **Generate client secret**, select **Enable sign-in API for server-based authentication**, then select **Create app client**.
* **Generate client secret**: user pool apps with a client secret are not supported by the JavaScript SDK. We need to un-select the option.
* **Enable sign-in API for server-based authentication**: required by AWS CLI when managing the pool users via command line interface. We will be creating a test user through the command line interface in the next chapter.

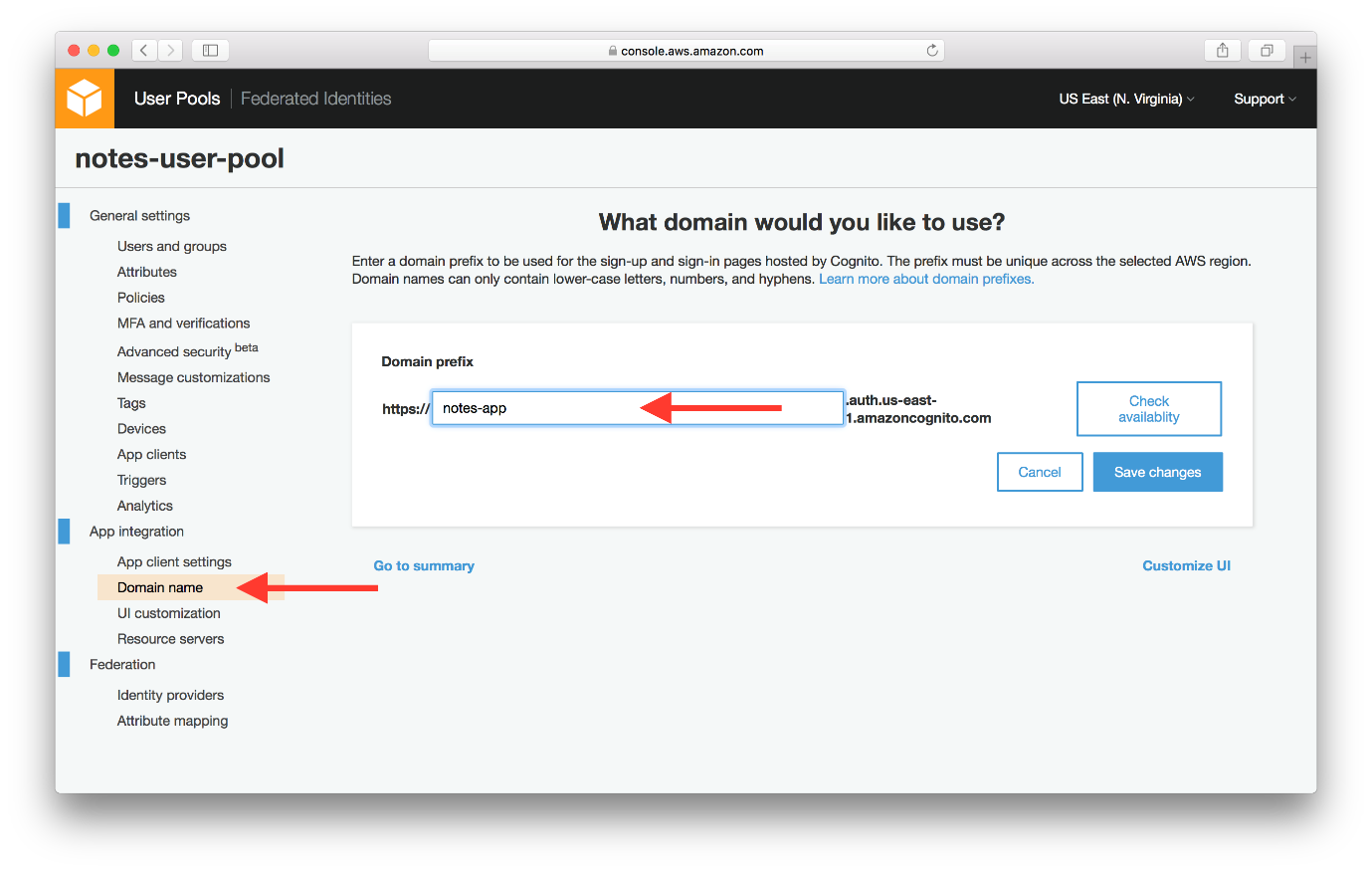


* Your app client has been created. Take note of the **App client id** which will be required in the later chapters.



1. Create Domain Name:

* Finally, select **Domain name** from the left panel. Enter your unique domain name and select **Save changes**. In our case we are using notes-app.



* You can use your own domain by clicking on “Your Own Domain” and then enter domain name and select aws managed certificate.
* Now our Cognito User Pool is ready. It will maintain a user directory for our notes app. It will also be used to authenticate access to our API. Next let’s set up a test user within the pool.