



Introduction

Welcome to Serverless Workshop! Based on my experiences of the [ServerlessConf](#) in Paris this year and my current interests, I have decided to host a Bits & Bites session in a workshop format on Serverless, more specifically: on the Amazon WebServices Serverless Platform, [AWS Lambda](#).

All information contained in this and subsequent tutorials is [caveat emptor](#)-ware, e.g. best effort at providing correct information, but you – the proud holder of the AWS account – will be ultimately responsible for handling your account and seeing to it that you do not exceed the limits of the [free tier](#) services.

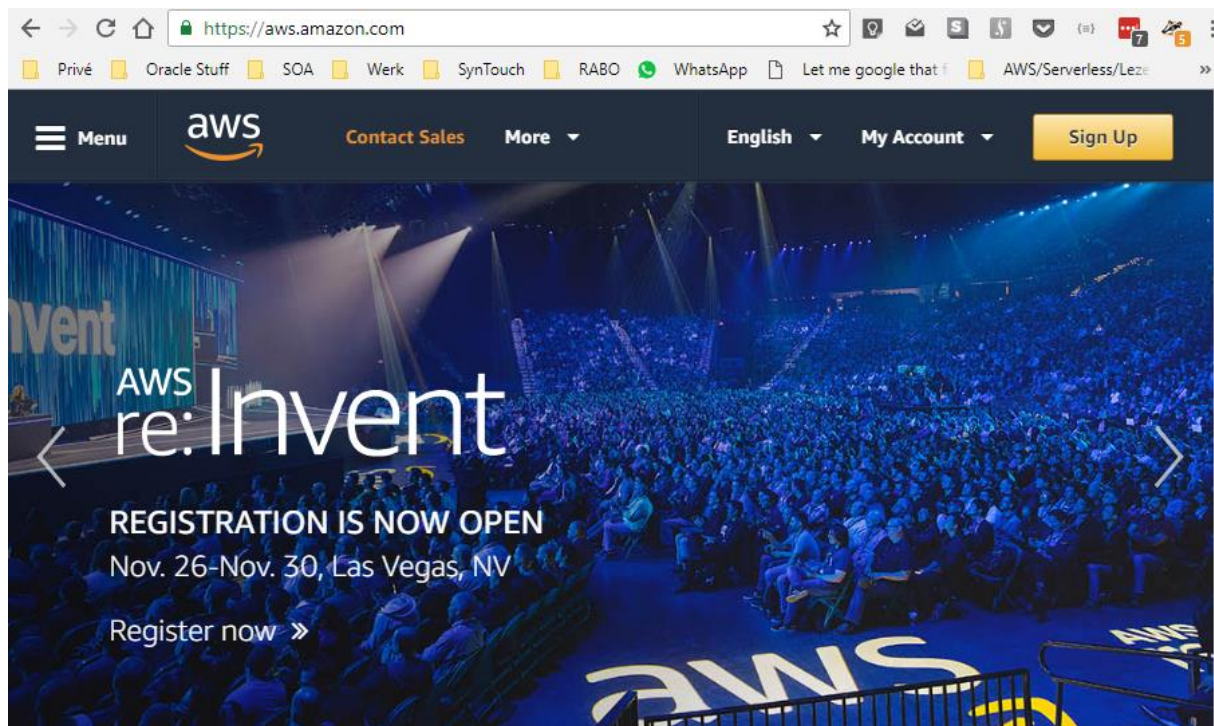
This first tutorial will guide you how to create and setup your account.

Registration

Registration for an AWS account is easy, but it does require you to supply your credit card details (you should not incur any charges from the activities in the workshop).

Creating your account

Navigate to the AWS site (aws.amazon.com), in the top right there is a “Sign Up” button:



In the next dialogue, provide the requested information; we'll setup your account to use the AWS account name in your sign-in URL. Register a [personal account](#) and provide personal details including your mobile phone number as you will receive a confirmation code for the completion of the registration:



Email address

Password

Confirm password

AWS account name ⓘ

[Continue](#)

[Sign in to an existing AWS account](#)

© 2018 Amazon Web Services, Inc. or its affiliates.
All rights reserved.
[Privacy Policy](#) | [Terms of Use](#)

Please select the account type and complete the fields below with your contact details.

Account type ⓘ

☐ Professional ☒ Personal

Full name

Phone number

Country/Region

Address

Apartment, suite, unit, building, floor, etc.

City

State / Province or region

Postal code

☒ Check here to indicate that you have read and agree to the terms of the [AWS Customer Agreement](#)

[Create Account and Continue](#)

Continue to provide your credit card details for billing:

Payment Information

Please type your payment information so we can verify your identity. We will not charge you unless your usage exceeds the AWS Free Tier Limits. Review [frequently asked questions](#) for more information.

Credit/Debit card number

Expiration date

Cardholder's name

Billing address

☒ Use my contact address

Hoofddorp Noord-Holland NL

☐ Use a new address

[Secure Submit](#)

You will be asked to confirm your telephone number and AWS will call you, instructing you to enter the code shown on the web page to confirm your identity.

Choose the **Basic Support** plan that is marked Free:



Select a Support Plan

AWS offers a selection of support plans to meet your needs. Choose the support plan that best aligns with your AWS usage. [Learn more](#)

Basic Plan	Developer Plan	Business Plan
Free	From \$29/month	From \$100/month
<ul style="list-style-type: none"> Included with all accounts 24/7 self-service access to forums and resources Best practice checks to help improve security and performance Access to health status and notifications 	<ul style="list-style-type: none"> For early adoption, testing and development Email access to AWS Support during business hours 1 primary contact can open an unlimited number of support cases 12-hour response time for nonproduction systems 	<ul style="list-style-type: none"> For production workloads & business-critical dependencies 24/7 chat, phone, and email access to AWS Support Unlimited contacts can open an unlimited number of support cases 1-hour response time for production systems

Need Enterprise level support?
Contact your account manager for additional information on running business and mission critical workloads on AWS (starting at \$15,000/month). [Learn more](#)

This is the final step:

Welcome to Amazon Web Services

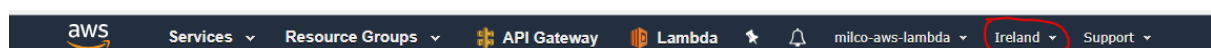
Thank you for creating an Amazon Web Services Account. We are activating your account, which should only take a few minutes. You will receive an email when this is complete.

Wait for the email to arrive

You have created the **ROOT** user, which means that this user has unlimited administrative access to the account, including billing information. As in UNIX, this is not a good user for day-to-day operations, so as soon as we're able to login we will need to create a restricted user and optionally secure access to the root account.

(Information) Regions

As soon as we login, you should take notice of the region you're working in – this will be shown on the top of the page. Although AWS has multiple regions in EMEA, not all services are already available in all of these regions. Whenever creating resources, these will be created in your current region. As we want to use the Cloud9 development environment, make sure that your region is set to Ireland (this is currently the only EMEA region that supports this service).





Account Setup

As soon as you have received the email that registration is completed, log into your root account:



Root user sign in

Email: milco.numan@protonmail.com

Password [Forgot password?](#)

Sign in

[Sign in to a different account](#)

[Create a new AWS account](#)

When logging in for the first time, you should be automatically directed to the IAM (Identity and Access Management) Module, where a number of items that require your attention will be marked by exclamation mark icons:

Welcome to Identity and Access Management

IAM users sign-in link:

<https://612457436284.signin.aws.amazon.com/console>

[Customize](#)

IAM Resources

Users: 0

Groups: 0

Customer Managed Policies: 0

Roles: 2

Identity Providers: 0

Security Status

1 out of 5 complete.

<input checked="" type="checkbox"/>	Delete your root access keys	▼
<input type="checkbox"/>	Activate MFA on your root account	▼
<input type="checkbox"/>	Create individual IAM users	▼
<input type="checkbox"/>	Use groups to assign permissions	▼
<input type="checkbox"/>	Apply an IAM password policy	▼

(Optional) Creating custom account sign-in link

Setup your account sign-in link to be more descriptive than the default <https://your-account-id-here.signin.aws.amazon.com/console> by overwriting the numeric account id with a more descriptive account alias (choose your preferred alias):

Create Account Alias ✕

Account Alias

Cancel Yes, Create


(Optional) Protecting your account using MFA

MFA stands for Multi Factor Authentication and is the mechanism to require multiple proofs of identity before granting access. Usually, the identification tokens consist of something you know (password) and something you have (authentication token generator) or are (biometric characteristics like finger print reader). It is always a good practice to protect your account with MFA!



Manage MFA device

If your virtual MFA application supports scanning QR codes, scan the following QR code with your smartphone's camera.



[Show secret key for manual configuration](#)
 After the application is configured, enter two consecutive authentication codes in the boxes below and choose **Activate virtual MFA**.

Authentication code 1

Authentication code 2

Cancel

Previous

Activate virtual MFA

Set Multi-Factor Authentication on your root account; for this to work, you will need to have an app like **Google Authenticator** or **Authy** running on your smartphone. AWS also supports miscellaneous hardware devices.

Scan the QR-code using your virtual MFA and follow instructions to require MFA for your account.

Creating a user and group

Next up, creating a *non-root* access user for day-to-day operations (administrator and development). This user will be granted elevated privileges (administrator rights for all services) that in a corporate scenario you typically would not have, but this makes development and administration easier for the workshop. Use the Manager Users button to start:

Security Status

2 out of 5 complete.

<input checked="" type="checkbox"/>	Delete your root access keys	▼
<input checked="" type="checkbox"/>	Activate MFA on your root account	▼
<input type="checkbox"/>	Create individual IAM users	▲
<p>Create IAM users and give them only the permissions they need. Do not use your AWS root account for day-to-day interaction with AWS, because the root account provides unrestricted access to your AWS resources. Learn More</p> <div>Manage Users</div>		
<input type="checkbox"/>	Use groups to assign permissions	▼
<input type="checkbox"/>	Apply an IAM password policy	▼

Create an admin user of your choice, enable for console access, set your password manually:

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*

admin

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.

Console password*

☐ Autogenerated password
 ☒ Custom password

Show password

Require password reset

☐ User must create a new password at next sign-in
Users automatically get the `IAMUserChangePassword` policy to allow them to change their own password.

* Required

Cancel

Next: Permissions



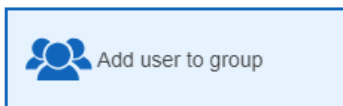
(Console access is for interactive users, programmatic access generates a new set of key and secret for command line access using the CLI and APIs).

On the next screen, Permissions, add the user to a new group you will create by using the “Create Group” button:

Add user

1

▼ Set permissions



Copy permissions from existing user



Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job fun

Add user to group



<input type="text" value="Search"/>	
Group ▼	Attached policies

Create an admin group, selecting the AdministratorAccess policy from the list:



Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name

Admin

A group named "Admin" already exists




Create policy

Refresh

Filter policies ▾

Q Administr

Showing 9 results

	Policy name ▾	Type	Used as
<input checked="" type="checkbox"/>	 AdministratorAccess	Job function	Permissions policy (1)
<input type="checkbox"/>	 AmazonAPIGatewa...	AWS managed	Permissions policy (1)
<input type="checkbox"/>	 AWS...	AWS managed	...

On returning, make sure the newly created Admin group is selected for the user, Review and Create.

(Optional) Password policies

In a real life scenario, password policies may also be applied, requiring you to generate “strong” passwords by requiring minimum length and certain other characteristics. If you feel like it, you can apply such a policy but as I am in the habit of generating strong passwords using LastPass, I will not apply a password policy on top of that:

▼ Password Policy

A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, go to [Managing Passwords](#) in Using IAM.

Currently, this AWS account does not have a password policy. Specify a password policy below.

Minimum password length:

6

☐ Require at least one uppercase letter ⓘ☐ Require at least one lowercase letter ⓘ☐ Require at least one number ⓘ☐ Require at least one non-alphanumeric character ⓘ☒ Allow users to change their own password ⓘ☐ Enable password expiration ⓘ

Password expiration period (in days):

☐ Prevent password reuse ⓘ

Number of passwords to remember:

☐ Password expiration requires administrator reset ⓘ

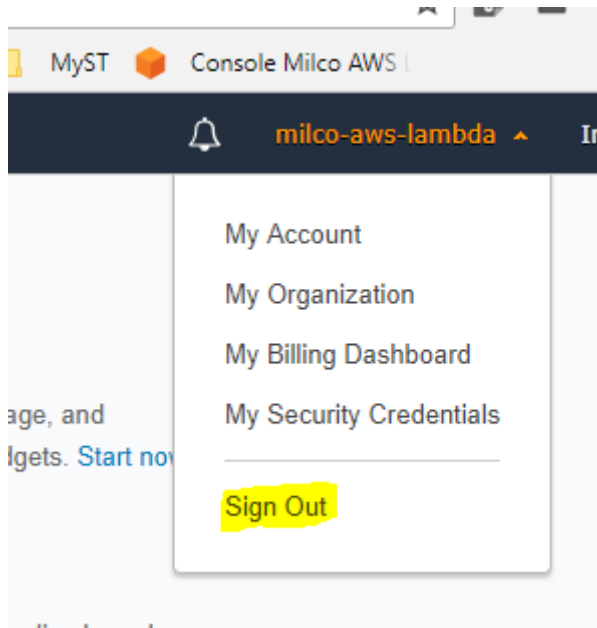
Apply password policy

Delete password policy



Test your user

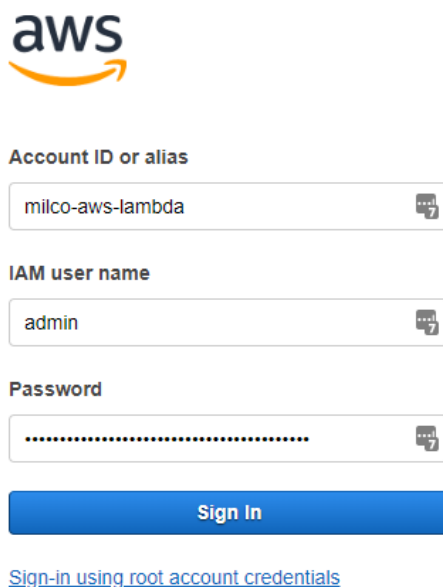
You're still working using your root account credentials, so it is time to logout and log back in again with your newly created admin user:



Login at your new account alias URL:



Now provide your admin user credentials to login (so **not your root account!**):



The image shows the AWS Sign In page. At the top is the AWS logo. Below it are three input fields: 'Account ID or alias' with the value 'milco-aws-lambda', 'IAM user name' with the value 'admin', and 'Password' which is masked with dots. Each input field has a small icon on the right. Below the fields is a blue 'Sign In' button. At the bottom, there is a link that says 'Sign-in using root account credentials'.

Next up: Cloud9