

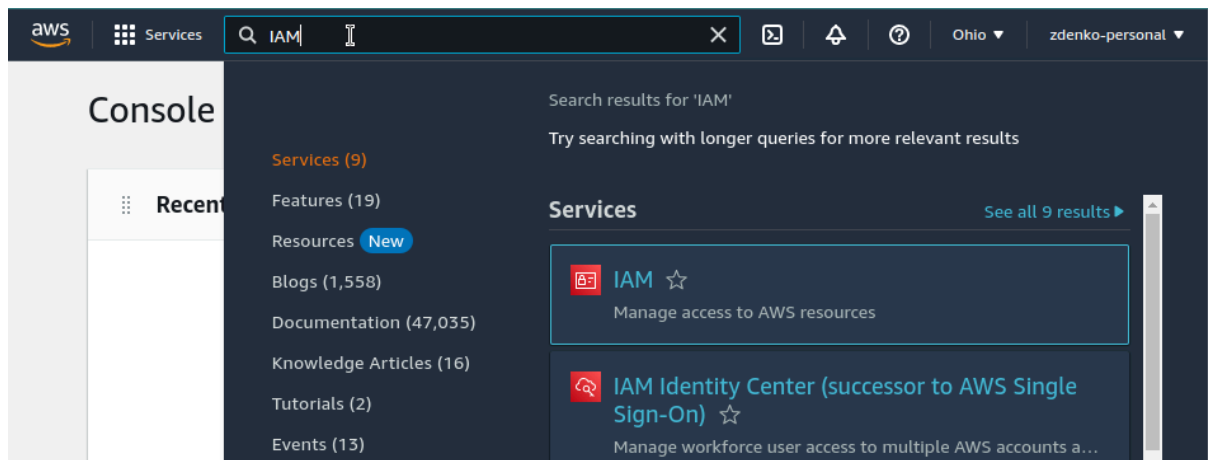
Usuarios IAM - Seguridad

Una vez creada la cuenta, lo ideal es crear un usuario Admin y no usar el correo con la que se dio de alta la cuenta de AWS, esto se va de la mano con la filosofía “less privilege”.

En esta parte, aumentaremos la seguridad de nuestra cuenta. Configuraremos la seguridad del usuario root como así también crearemos un usuario administrador para utilizar en las prácticas en vez del usuario root.

Primera parte MFA cuenta root

- 1) En esta primer mitad de la segunda parte nos encargaremos de hacer más seguro al usuario root, para esto tendremos que buscar el servicio IAM



- 2) Una vez ya en el servicio IAM tendremos que tener una ventana como la siguiente:

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings

Access reports

- Access analyzer
- Archive rules
- Analyzers
- Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Related consoles

- [IAM Identity Center](#) **New**
- [AWS Organizations](#)

IAM dashboard

Security recommendations **1**

Add MFA for root user
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.

Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.

IAM resources

User groups	Users	Roles	Policies
0	0	2	0

What's new

Updates for features in IAM

- Advanced Notice:** Amazon S3 will automatically enable S3 Block Public Access and disable all new buckets starting in April 2023. 5 months ago
- AWS IAM Identity Center** now supports session management capabilities for AWS Command Line Interface (CLI) and SDKs. 6 months ago
- AWS Lambda** announces support for Attribute-Based Access Control (ABAC) in AWS GovCloud. 6 months ago
- Amazon ElastiCache** simplifies password rotations with Secrets Manager. 6 months ago

[more](#)

- 3) Vamos a asegurarnos de configurar las 2 sugerencias de seguridad, que el usuario Root no tenga access keys activas y que tenga MFA activado para mayor seguridad. Así nos debería quedar (si comparamos con la img anterior no tengo recomendaciones):

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles

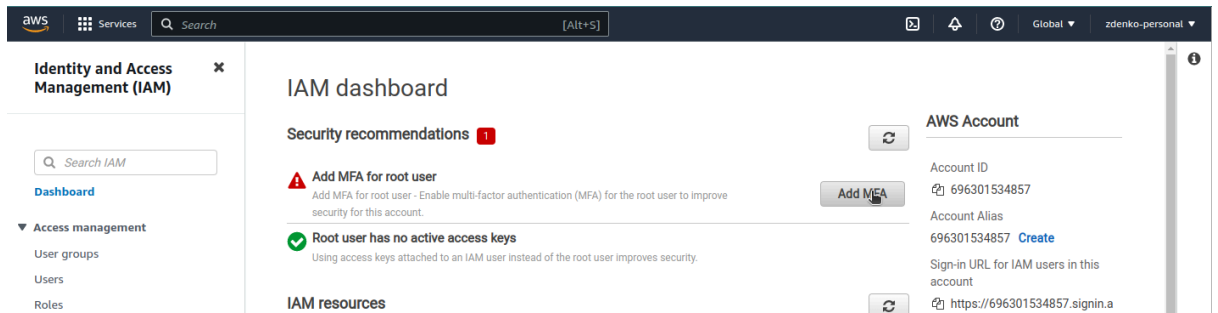
IAM Dashboard

Security recommendations **0**

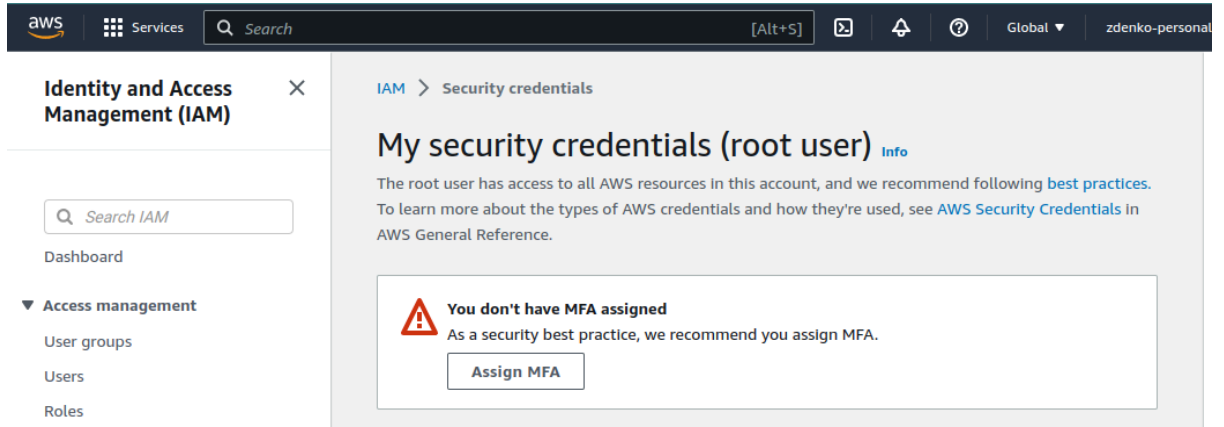
Root user has MFA
Having multi-factor authentication (MFA) for the root user improves security for this account.

Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.

- 4) En el caso de cuentas nuevas el usuario Root por defecto no tiene llaves de acceso pero sí tendremos que configurar el MFA, haremos click en el botón “Add MFA”



5) Una vez en la sección de seguridad, agregaremos un dispositivo de MFA:



6) Para esto, pueden usar alguna aplicación o dispositivo físico, en nuestro caso usaremos Google Authenticator:

aws

Services

Search

[Alt+S]

Global

zdenko

≡

IAM

>

Security credentials

>

Assign MFA device

Step 1 of 2

Select MFA device

Specify MFA device name

Device name

Enter a meaningful name to identify this device.

Dispositivo-MFA


Maximum 128 characters. Use alphanumeric and '+', '.', '@', '-', '_' characters.

Select MFA device

Info

Select an MFA device to use, in addition to your username and password, whenever you need to authenticate.


☒



Authenticator app

Authenticate using a code generated by an app installed on your mobile device or computer.


☐



Security Key

Authenticate using a code generated by touching a YubiKey or other supported FIDO security key.

☐



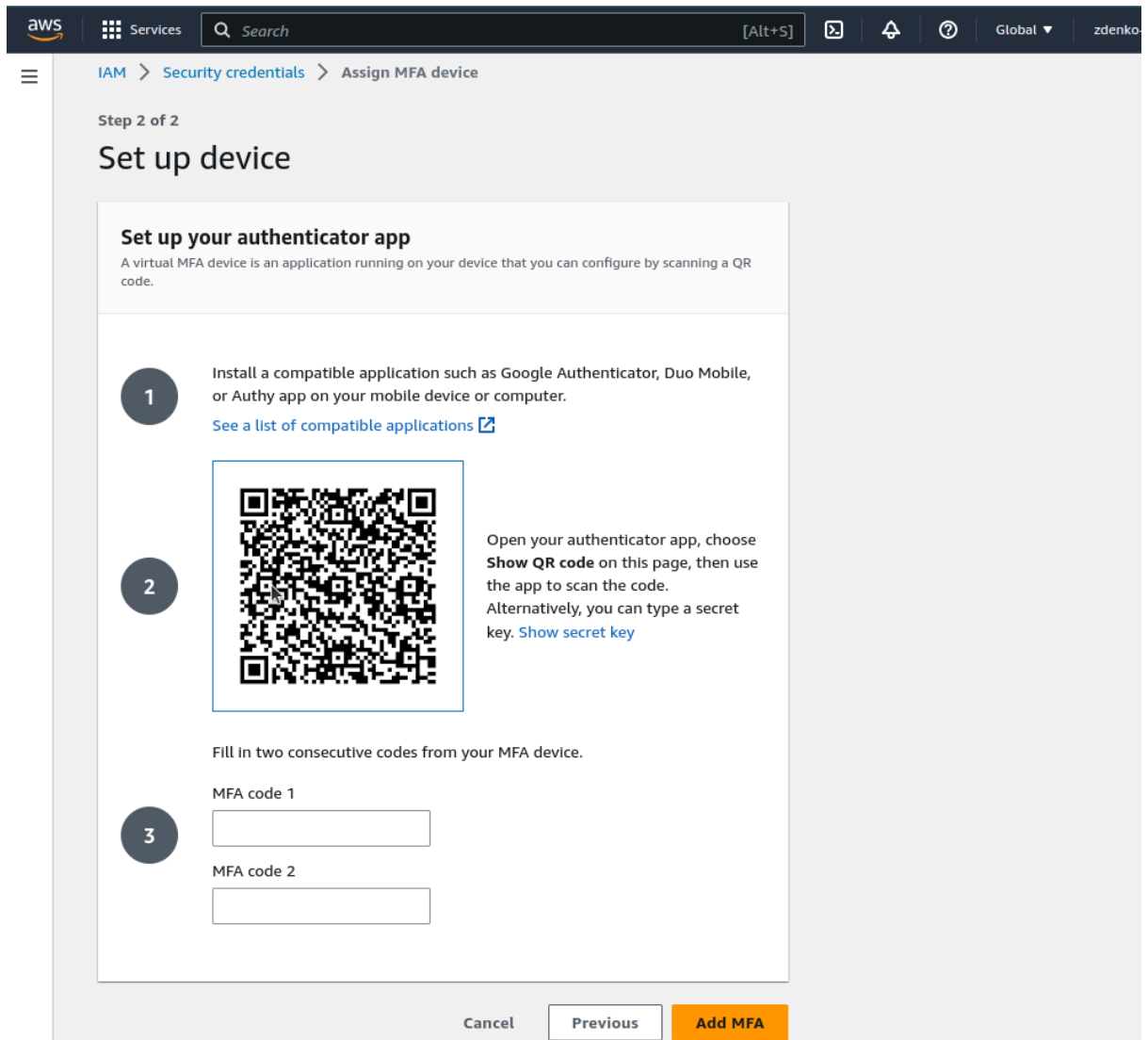
Hardware TOTP token

Authenticate using a code displayed on a hardware Time-based one-time password (TOTP) token.

Cancel

Next

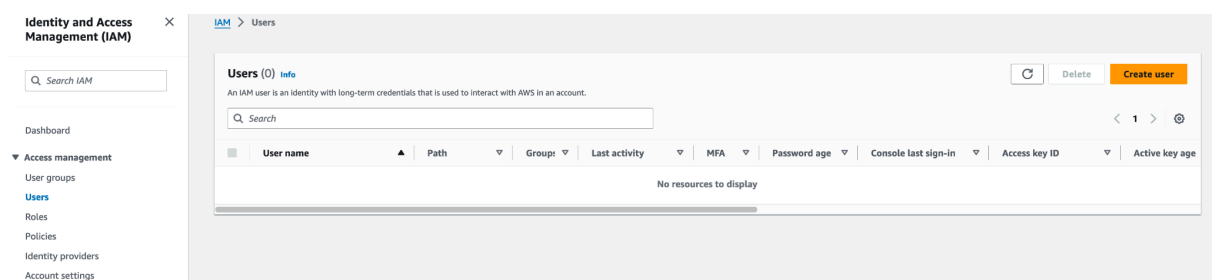
- 7) Seguimos los pasos que nos dice la guía de configuración para agregar nuestro código



Y una vez terminado nuestro usuario root ya quedo seguro (MFA activado y sin llaves de acceso programático, solo por Consola)

Segunda parte creado usuario Admin

- 8) Ahora, crearemos un usuario Admin que será el usuario que usaremos para las prácticas, en nuestro caso crearemos un usuario para todos los servicios pero pueden crear un usuario con permisos más acotados e ir añadiendo permisos a medida que los vayan necesitando en las prácticas. Para crear este usuario, iremos a la sección de Users (menú a la izquierda) dentro del servicio IAM:



- 9) Y ahora le daremos al botón **Create Users** y nos dará la ventana de la imagen; es importante destacar que queremos marcar la opción de que el usuario tenga acceso a la Consola de AWS y que sea un usuario IAM, ya que necesitaremos acceso programático, le configuramos una contraseña y que la deba de cambiar; click en **Next**

The screenshot shows the AWS IAM 'Create user' console. The left sidebar indicates the current step is 'Specify user details'. The main content area is titled 'Specify user details' and contains the following sections:

- User details**
 - User name**: A text input field containing 'adminjg'. Below it, a note states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, -, @, _ (hyphen)'.
 - Provide user access to the AWS Management Console - optional**: A checkbox that is checked. Below it, a note states: 'If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.'
 - Are you providing console access to a person?**: A section with two radio button options:
 - Specify a user in Identity Center - Recommended**: Unselected. Below it, a note states: 'We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.'
 - I want to create an IAM user**: Selected. Below it, a note states: 'We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.'
- Console password**
 - Autogenerated password**: Unselected. Below it, a note states: 'You can view the password after you create the user.'
 - Custom password**: Selected. Below it, a text input field is present with a masked password '*****'. Below the field, two requirements are listed:
 - 'Must be at least 8 characters long'
 - 'Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols (!@#\$%^&*()_+-=;{}|'`~<>[]\,./:~)'
 - Show password**: An unchecked checkbox.
- Users must create a new password at next sign-in - Recommended**: A checked checkbox. Below it, a note states: 'Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.'

At the bottom right of the console, there are 'Cancel' and 'Next' buttons. A blue information box at the bottom of the main content area states: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)'.

- 10) El siguiente paso será asignar permisos, como solo usaremos un usuario no será necesario crear un grupo y asignarle los permisos al grupo sino que directamente asignaremos los permisos al usuario, en este caso usaremos una Policy que sea de Administrador y que nos brinde acceso a todos los servicios. Para esto, seleccionaremos la opción **“Attach policies directly”** y en el buscador de policies, buscaremos una llamada **“AdministratorAccess”**, tendrá que darnos la siguiente policy

aws Services Search [Alt+S] Global zdenko-personal

IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Set permissions

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 functions. [Learn more](#)

Permissions options

- ☐ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- ☐ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- ☒ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1090)

Choose one or more policies to attach to your new user.

admin 36 matches

<input type="checkbox"/>	Policy name	Type	Attached en...
<input type="checkbox"/>	AdministratorAccess	AWS managed - job f...	0

AdministratorAccess

Provides full access to AWS services and resources.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "*",  
7       "Resource": "*"   
8     }  
9   ]  
10 }
```

11) Seleccionamos esta policy y le damos a **Next**, ahora podremos revisar que esté todo como esperamos:

The screenshot shows the AWS IAM console's 'Create user' wizard, specifically the 'Review and create' step. The left sidebar lists four steps: 'Specify user details', 'Set permissions', 'Review and create' (which is the active step), and 'Retrieve password'. The main content area is titled 'Review and create' and includes a sub-header 'Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.'

The 'User details' section shows the 'User name' as 'zdenko-personal' and the 'Console password type' as 'Custom password'. The 'Require password reset' option is set to 'No'.

The 'Permissions summary' section shows a table with one entry: 'AdministratorAccess' (Name), 'AWS managed - Job function' (Type), and 'Permissions policy' (Used as). The table has a pagination control showing '1' of 1 items.

The 'Tags - optional' section explains that tags are key-value pairs for resource identification and provides a button to 'Add new tag'. It notes that up to 50 tags can be added.

At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create user' (highlighted in orange).

12) **Create User**. Al crear el usuario, nos dará una opción de descargar un archivo .csv, recomendamos descargarlo para tener un backup de nuestras credenciales. De ahora en más, este es el usuario que utilizaremos

The screenshot shows the AWS IAM console after a user has been successfully created. A green banner at the top states 'User created successfully' and provides a link to 'View user'. Below this, the 'Retrieve password' step is active, showing the 'Console sign-in details' for the user 'adminJG'.

The 'Console sign-in details' section includes the 'Console sign-in URL' (https://083911806224.signin.aws.amazon.com/console), the 'User name' (adminJG), and the 'Console password' (masked with dots and a 'Show' button). There is also a link to 'Email sign-in instructions'.

At the bottom right, there are three buttons: 'Cancel', 'Download .csv file' (highlighted in orange), and 'Return to users list'.

Recomendamos añadirle un MFA de la misma forma que lo hicimos para el usuario root.

- 13) Una vez creado el usuario, en Users click sobre el nombre del usuario de IAM para configurar un Access Key, para poder acceder a los servicios de AWS de forma programática. Para esto, moverse a la pestaña “**Security credentials**”, bajar hasta encontrar la sección de **Access Key** le daremos al botón de **Create access key**:

The screenshot shows the AWS IAM console interface. The top navigation bar includes the AWS logo, a search bar, and the user's profile (zdenko-personal). The left sidebar displays the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, Access reports, and Related consoles. The main content area is titled 'Security credentials' and contains three sections: 'Console sign-in', 'Multi-factor authentication (MFA) (0)', and 'Access keys (0)'. The 'Access keys (0)' section is highlighted, showing a 'Create access key' button. Below this, a message states 'No access keys' and advises using short-term credentials. At the bottom, there is a section for 'SSH public keys for AWS CodeCommit (0)'.

Console sign-in [Manage console access](#)

Console sign-in link
<https://696301534857.signin.aws.amazon.com/console>

Console password
Updated 1 minute ago (2023-05-26 23:00 GMT-05:00)

Last console sign-in
Never

Multi-factor authentication (MFA) (0)
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. [Learn more](#)

[Remove](#) [Resync](#) [Assign MFA device](#)

Device type	Identifier	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment		

[Assign MFA device](#)

Access keys (0)
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

[Create access key](#)

No access keys
As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

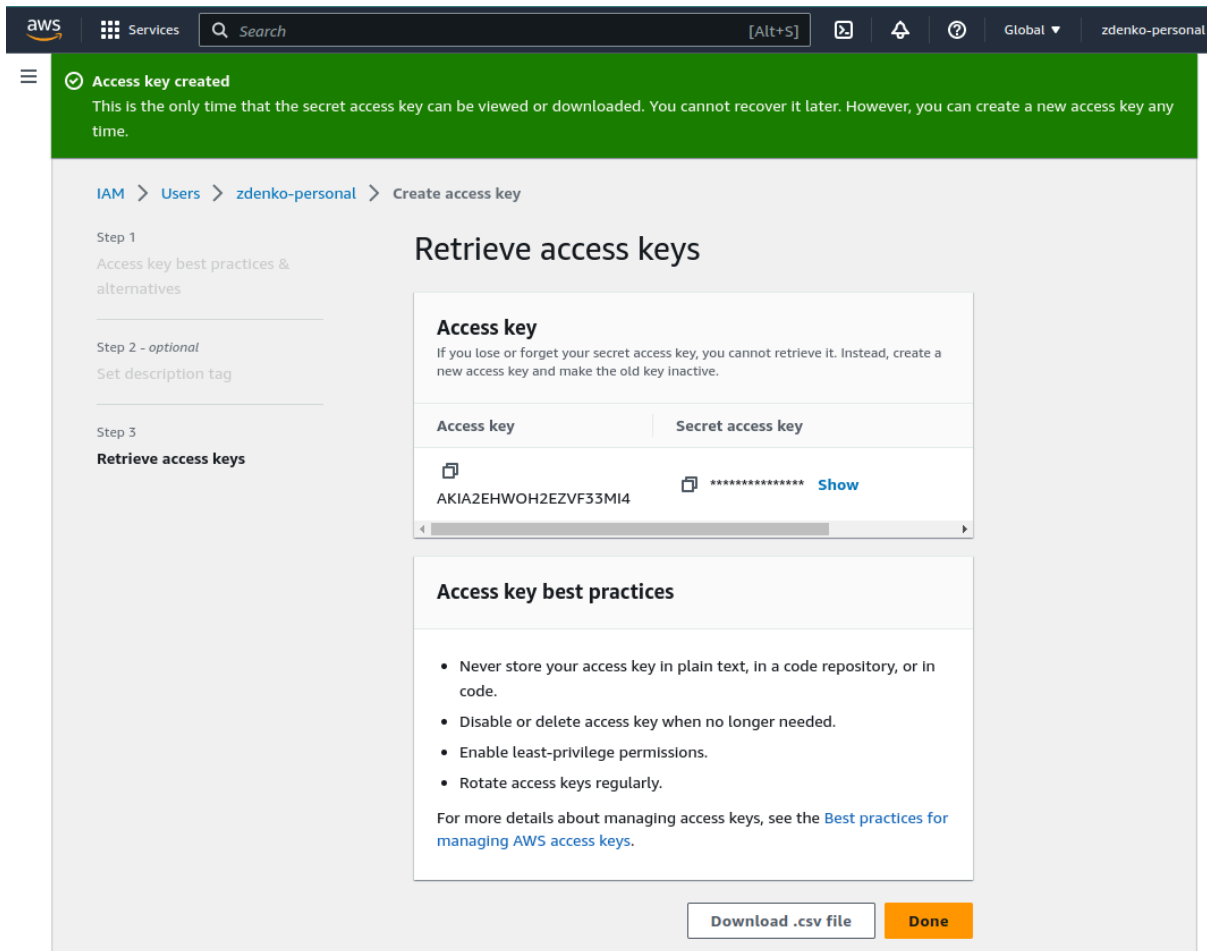
[Create access key](#)

SSH public keys for AWS CodeCommit (0)

- 14) Elegiremos la opción Other, luego **Next**

The screenshot shows the AWS IAM console interface. The top navigation bar includes the AWS logo, 'Services', a search bar, and user information 'Global' and 'zdenko-personal'. The left sidebar contains a menu icon and a list of steps: 'Step 1: Access key best practices & alternatives', 'Step 2 - optional: Set description tag', and 'Step 3: Retrieve access keys'. The main content area is titled 'Access key best practices & alternatives' and includes a warning: 'Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.' Below this, there are six radio button options for use cases: 'Command Line Interface (CLI)', 'Local code', 'Application running on an AWS compute service', 'Third-party service', 'Application running outside AWS', and 'Other' (which is selected). An information box at the bottom states: 'It's okay to use an access key for this use case, but follow the best practices:' followed by a list of four best practices: 'Never store your access key in plain text, in a code repository, or in code.', 'Disable or delete access keys when no longer needed.', 'Enable least-privilege permissions.', and 'Rotate access keys regularly.'

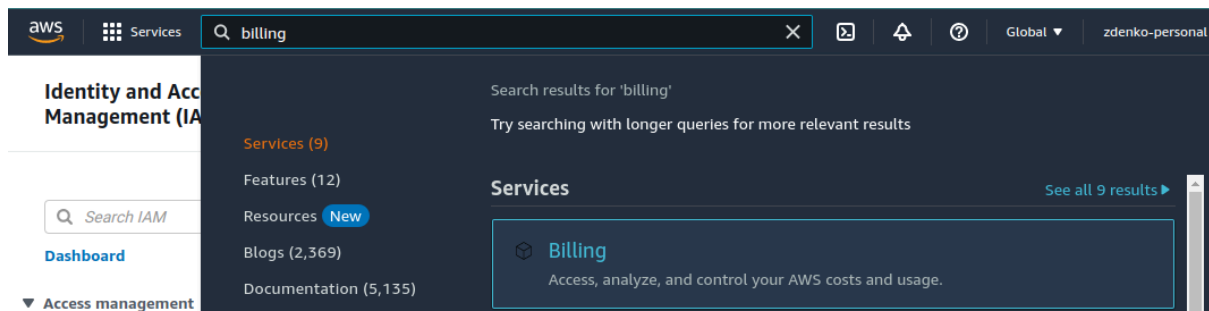
15) En la siguiente ventana no le agregaremos un tag sino que directamente crearemos el access key, click en **Create access key**, en la siguiente ventana nos mostrará el access key y el secret key con la opción de descargar el .csv de nuevo (recomendamos guardar una copia de este archivo). Es importante guardar estas credenciales ya que son las que utilizaremos en herramientas como Terraform por ej. En caso de perder estas credenciales podremos eliminar estas Access Keys y crear otras.



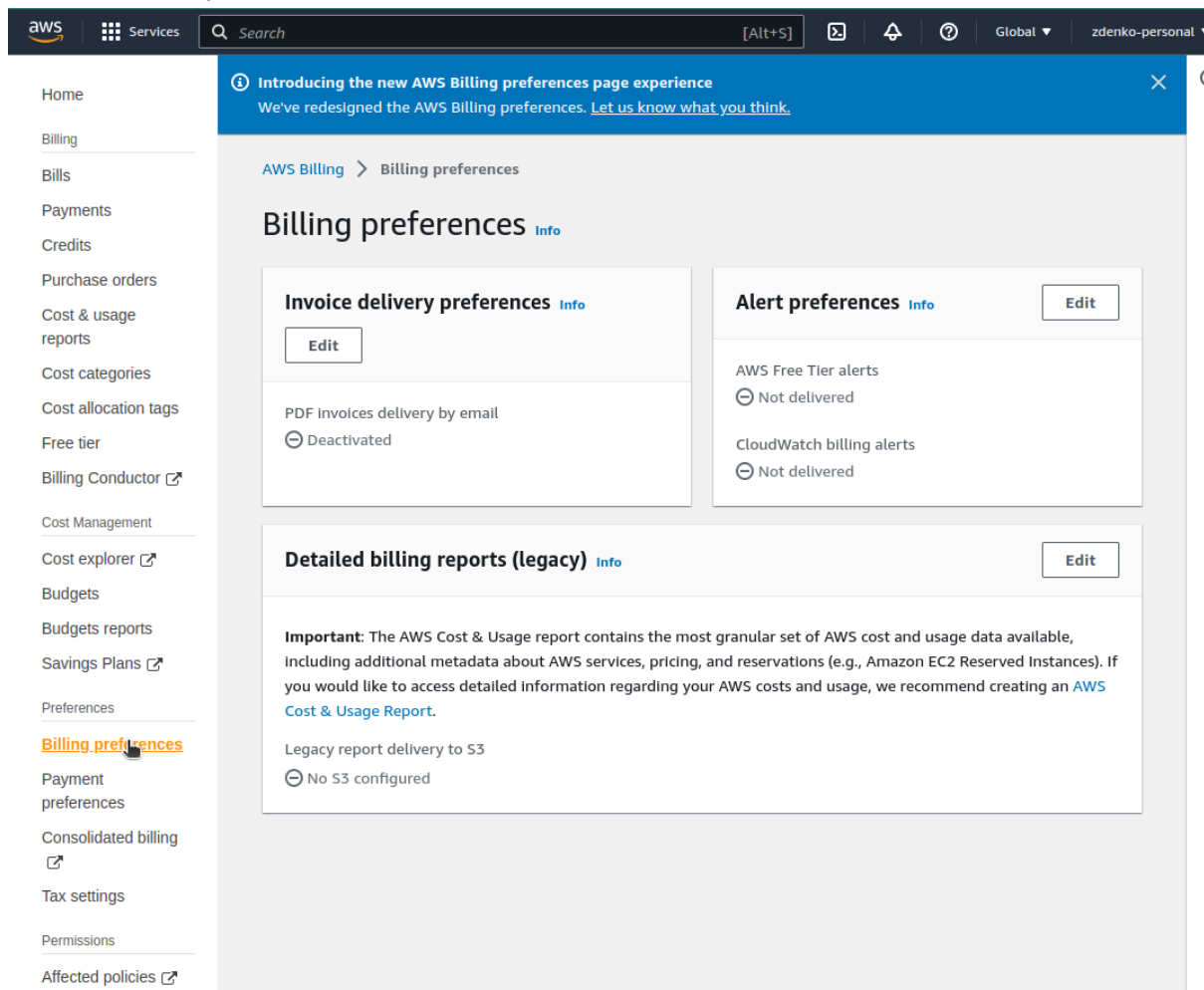
- 16) Y listo, así tendremos nuestro usuario Administrador listo para utilizar ya sea desde la Consola de AWS o de forma programática desde nuestra terminal u otras aplicaciones.

Presupuestos y Alertas

- 1) En esta última parte, explicaremos un poco la sección gratuita de AWS (además de proveer recursos para una lectura más en detalle) además de configurar alertas para que en caso de llegar al límite del uso gratuito o en caso de tener algún costo, que nos notifiquen para poder así apagar el servicio. El free tier de AWS se compone de 3 tipos: Un periodo de 12 meses de free tier para algunos servicios, otros servicios que son gratuitos para siempre y pruebas gratuitas cortas (para algunos servicios específicos). Siempre antes de crear cualquier tipo de infraestructura o de utilizar cualquier servicios, que revisemos los costos asociados con este, para más información del free tier visitar la siguiente pagina:
<https://aws.amazon.com/free/free-tier-faqs/>
- 2) Primero configuraremos las alertas por uso del free tier que nos notifican al llegar al 85% de uso de free tier de los servicios, para esto buscaremos el servicio de billing:



- 3) Una vez dentro de billing, iremos a **Billing Preferences** en la izquierda, casi abajo de todo y aca activaremos las alertas de AWS Free tier:



Alert preferences [Info](#)

[Edit](#)

AWS Free Tier alerts

⊖ Not delivered

CloudWatch billing alerts

⊖ Not delivered

Alert preferences [Info](#)

☒ Receive AWS Free Tier alerts

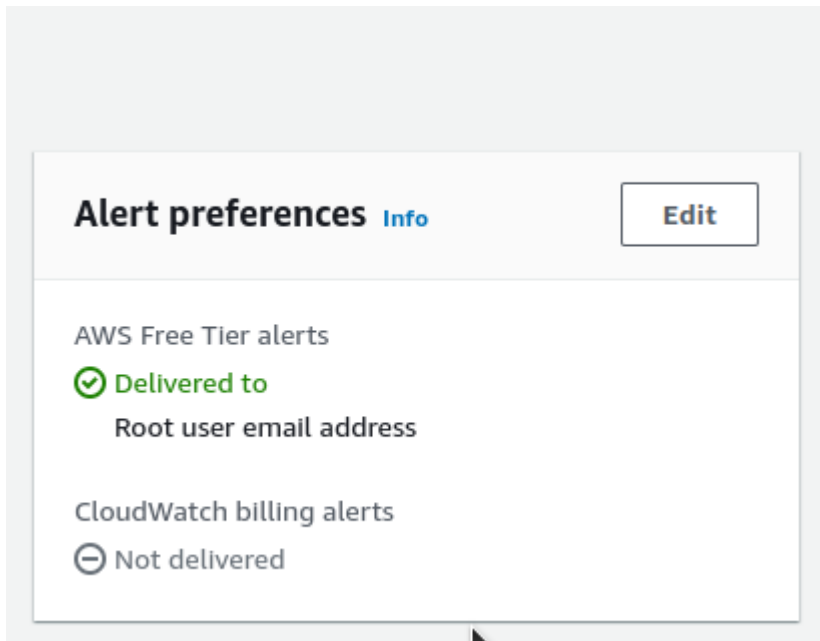
Your AWS Free Tier usage alerts will be delivered to this account's root user email address if this is activated. You can add an additional recipient for these email alerts.

Additional email address to receive alerts - *optional*

☐ Receive CloudWatch billing alerts

Once enabled, this preference cannot be disabled.

[Update](#)[Cancel](#)



Por default, estas alertas se activan al 85% del límite del free tier.

- 4) Pero no solo queremos configurar las alertas para el free tier, que pasa si sin querer utilizamos un servicio fuera del free tier? Bueno, para ello configuraremos un presupuesto que se limite a 0 dólares gastados y que nos alerte.
- 5) Para esto, dentro del mismo servicio de Billing, iremos a la sección de **Budgets**

The screenshot shows the AWS Billing console with the 'Budgets' section selected in the left-hand navigation menu. The main content area is titled 'AWS Budgets' and features a large heading 'Set custom budgets that alert you when you exceed your budgeted thresholds'. Below this, a sub-header reads 'Start tracking your AWS costs and usage', followed by a description: 'Once you have a budget created, AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.' A prominent orange button labeled 'Create a budget' is displayed. Further down, a 'How it works' section contains a four-step flowchart: 1. 'AWS Budgets' (Improve planning and cost), 2. 'Create a budget' (Customize how you want to), 3. 'Get alerted' (Receive alerts when you), and 4. 'Respond with actions' (Define and trigger cost). The 'Create a budget' step is highlighted with a yellow background in the original image.

Una vez aquí damos en **Create a budget**

- 6) Sin cambiar nada ya estará por default la opción de utilizar un template y utilizar el **Zero Spend Budget** template, este presupuesto nos indicará una vez que los costos pasen los 0.01 centavos de dólar

aws Services Search [Alt+S] Global zdenko-personal

Home Billing Bills Payments Credits Purchase orders Cost & usage reports Cost categories Cost allocation tags Free tier Billing Conductor Cost Management Cost explorer **Budgets** Budgets reports Savings Plans Preferences Billing preferences Payment preferences Consolidated billing Tax settings Permissions Affected policies

AWS Billing > Budgets > Create budget

Choose budget type [Info](#)

Budget setup

☒ **Use a template (simplified)**
Use the recommended configurations. You can change some configuration options after the budget is created.

☐ **Customize (advanced)**
Customize a budget to set parameters specific to your use case. You can customize the time period, the start month, and specific accounts.

Templates - new

Choose a template that best matches your use case.

☒ **Zero spend budget**
Create a budget that notifies you once your spending exceeds \$0.01 which is above the AWS Free Tier limits.

☐ **Monthly cost budget**
Create a monthly budget that notifies you if you exceed, or are forecasted to exceed, the budget amount.

☐ **Daily Savings Plans coverage budget**
Create a coverage budget for your Savings Plans that notifies you when you fall below the defined target.

☐ **Daily reservation utilization budget**
Create a utilization budget for your reservations that notifies you when you fall below the defined target.

Zero spend budget - Template

Budget name
Provide a descriptive name for this budget.

Names must be between 1-100 characters.

Email recipients
Specify the email recipients you want to notify when the threshold has exceeded.

- 7) Una vez creado el presupuesto nuestra cuenta ya estará lista para utilizarse, es importante no olvidarse de agregar el mail al que queremos que se envíen las notificaciones en el paso anterior.

aws Services Search [Alt+S] Global zdenko-personal

Home Billing Bills Payments Credits Purchase orders Cost & usage reports Cost categories Cost allocation tags Free tier Billing Conductor Cost Management

✓ Your budget **Zero spend budget** has been created successfully. After creating a budget, it can take up to 24 hours to populate all of your spend data. [Submit feedback](#)

AWS Billing > Budgets > Overview

Overview [Info](#)

Budgets (1) [Info](#) [Download CSV](#) [Actions](#) [Create budget](#)

[Show all budgets](#) < 1 > ⚙️

<input type="checkbox"/>	Name	Thresholds	Budget	Amo...	Forec...
<input type="checkbox"/>	Zero spend budget	OK	\$1.00	\$0.00	-