



Tecnológico de Monterrey

Desarrollo de Aplicaciones Avanzadas de Ciencias Computacionales

Gpo 201

Docentes

Benjamín Valdés Aguirre

Pedro Oscar Pérez Murueta

Manuel Casillas

Integrantes

Carlos Rodrigo Salguero Alcántara

A00833341

Diego Perdomo Salcedo

A01709150

Querétaro, Querétaro

1.0 Entendimiento del Negocio	4
1.1 Contexto	4
1.2 Objetivos de negocio y de minería de datos	4
1.2.1 Objetivo de Negocio	4
1.2.2 Criterios de éxito	4
1.2.3 Objetivo de minería de datos	4
1.2.4 Criterios de éxito	4
2.0 Entendimiento de los datos	5
2.1 Adquisición de datos	5
2.2 Caracterización del conjunto de datos	5
3.0 Preparación de los Datos	5
4.0 Modelado	6
4.1 Diseño de las Pruebas	6
4.2 Metricas	6
4.3 Modelo	8
4.4 Resultados	8
4.5 Validación de Objetivos	10
4.6 Hallazgos	11
4.6.1 Hallazgos extra solicitados por el Socio Formador	12
5.0 Deployment	13
5.1 Requisitos previos	13
5.1.1 Hardware	13
5.1.2 Software	13
5.1.3 Credenciales y accesos	14
5.2 Estructura del proyecto	14
5.3 Configuración del entorno (En la Raspberry PI, o en la PC)	14
5.3.1 Instalación de Docker Engine	14
5.3.2 Preparación de directorios	16
5.3.3 Archivo .env	16
5.3.4 Docker Compose	16
5.3.5 Descargar imagen de docker	17
5.4 AWS	17
5.4.1 Creación de cuenta	17
5.4.2 Creación de Bucket S3	17
5.4.3 Creación de cuenta para acceso externo a la Bucket	19
5.4.4 Creación de claves de acceso	24
5.4 Región	26
5.5 Ejecución (En la Raspberry PI, o en la PC)	28
5.5.1 Verificar estado del contenedor	28
5.5.2 Verificar logs del contenedor	28
5.5.3 Revisar AWS	28
5.5.4 Detener contenedor y modelo	28

1.0 Entendimiento del Negocio

1.1 Contexto

El CAETEC (Campo Agro Experimental del Tec de Monterrey) es un laboratorio para el aprendizaje de actividades agropecuarias. Este cuenta con un establo con corrales destinados a las vacas que habitan esta área. Las vacas tienen varias actividades que llegan a realizar a lo largo de un día, tales como; socializar con otras vacas, comer, dormir en sus camas, y ser ordeñadas automáticamente por una máquina. En esta área del CAETEC una de sus prioridades es mantener a las vacas en un nivel alto de comodidad. Esto es porque mientras más cómodas se encuentren tienden a rendir una mayor cantidad y calidad de leche.

El sector lácteo enfrenta desafíos continuos en la optimización de sus procesos de ordeño. La eficiencia en este proceso no solo afecta la productividad de la granja, sino también el bienestar animal y la calidad de la leche. Este proyecto de ciencia de datos busca abordar estas cuestiones mediante el análisis avanzado de datos y la implementación de técnicas de visión por computadora.

1.2 Objetivos de negocio y de minería de datos

1.2.1 Objetivo de Negocio

Identificar el número de vacas en cada fila en un periodo de tiempo determinado.

1.2.2 Criterios de éxito

- ❖ Determinar con alta precisión la cantidad de vacas en una imagen. Arturo o Ivo determinarán si la precisión es satisfactoria.

1.2.3 Objetivo de minería de datos

Determinar la cantidad de vacas en cada imagen en cualquier condición.

1.2.4 Criterios de éxito

- ❖ Un modelo para condiciones diurnas con un 80% de precisión.
- ❖ Un modelo para condiciones nocturnas con un 50% de precisión

2.0 Entendimiento de los datos

2.1 Adquisición de datos

Se llevó a cabo utilizando una webcam Logitech instalada en el techo del corral, capturando imágenes desde una vista superior. Este enfoque permitió obtener datos visuales en ciclos continuos tanto de día como de noche. El conjunto de datos fue compartido y descargado mediante la plataforma OneDrive, garantizando un acceso centralizado y eficiente para el equipo de análisis.

Considerando que los datos fueron otorgados por el CAETEC no tuvimos problemas en adquirirlos.

2.2 Caracterización del conjunto de datos

Originalment el dataset contiene un aproximado de 8110 imágenes con resolución de 1920x1080, clasificados según el número de vacas presentes. La principal característica del conjunto es su variación extrema de iluminación, con imágenes muy oscuras o completamente negras durante la noche y alta luminosidad con posible sobreexposición en el día.

3.0 Preparación de los Datos

El etiquetado del dataset se hizo con segmentación, utilizando Bounding - Boxes para marcar la ubicación exacta de cada vaca en la imagen por medio de coordenadas y de esta forma contar la cantidad de vacas en la imagen.

Para asegurar la consistencia, se establecieron criterios específicos que definen cuándo se debe considerar la presencia de una vaca. Se determinó que una vaca debe ser contabilizada cuando al menos el 20% de su cuerpo es visible en la imagen. Por otro lado, se establecieron criterios claros para elementos que no deben ser considerados como vacas:

Criterios de Exclusión:

- Partes aisladas del animal (cola, pata, oreja)
- Personal de trabajo

- Desechos animales
- Aves en el entorno
- Sombras (de vacas o personas)
- Elementos del entorno (botas, escaleras, puertas)

4.0 Modelado

4.1 Diseño de las Pruebas

Para evaluar la calidad del modelo, dividimos el dataset en dos conjuntos principales: entrenamiento y validación. Esta división nos permite entrenar el modelo y validar su precisión de manera independiente. La distribución se realizó con aproximadamente 81% para entrenamiento (7,283 imágenes) y 19% para validación (1,658 imágenes), manteniendo una proporción que asegura suficientes datos para ambas fases.

La división del dataset se realizó manteniendo una estructura organizada:

- ❖ Entrenamiento: 7,283 imágenes
 - 6,469 imágenes con vacas
 - 1,218 imágenes de fondo
- ❖ Validación: 1,658 imágenes
 - 1,618 imágenes con vacas
 - 144 imágenes de fondo

Esta distribución asegura que tanto el conjunto de entrenamiento como el de validación contengan una mezcla representativa de imágenes con y sin vacas, permitiendo una evaluación robusta del rendimiento del modelo.

4.2 Metricas

El diseño de pruebas utiliza cuatro métricas fundamentales para validar el cumplimiento de nuestro objetivo de minería de datos.

Objetivo de Minería de datos:

- ❖ Determinar la cantidad de vacas en cada imagen en cualquier condición.
 - Un modelo para condiciones diurnas con un 80% de precisión.

- Un modelo para condiciones nocturnas con un 50% de precisión.

Métricas:

❖ **Precisión (P):**

- Mide el porcentaje de detecciones correctas entre todas las detecciones realizadas
- Nos ayuda a validar la exactitud de las predicciones del modelo
- Directamente relacionada con nuestros criterios de éxito de 80% para condiciones diurnas y 50% para nocturnas
- Un valor alto indica que el modelo tiene pocos falsos positivos

❖ **Recall (R):**

- Indica la proporción de vacas reales que el modelo logra detectar
- Crucial para asegurar que no se están omitiendo vacas en el conteo
- Complementa la precisión para una evaluación más completa
- Un valor alto significa que el modelo raramente pasa por alto vacas presentes

❖ **mAP50 (Mean Average Precision con IoU=50%):**

- Evalúa la precisión del modelo considerando un umbral de solapamiento del 50%
- Métrica estándar para evaluar la calidad de las detecciones
- Proporciona una visión general del rendimiento del modelo
- Útil para comparar con los umbrales de precisión establecidos en nuestros criterios

❖ **mAP50-95:**

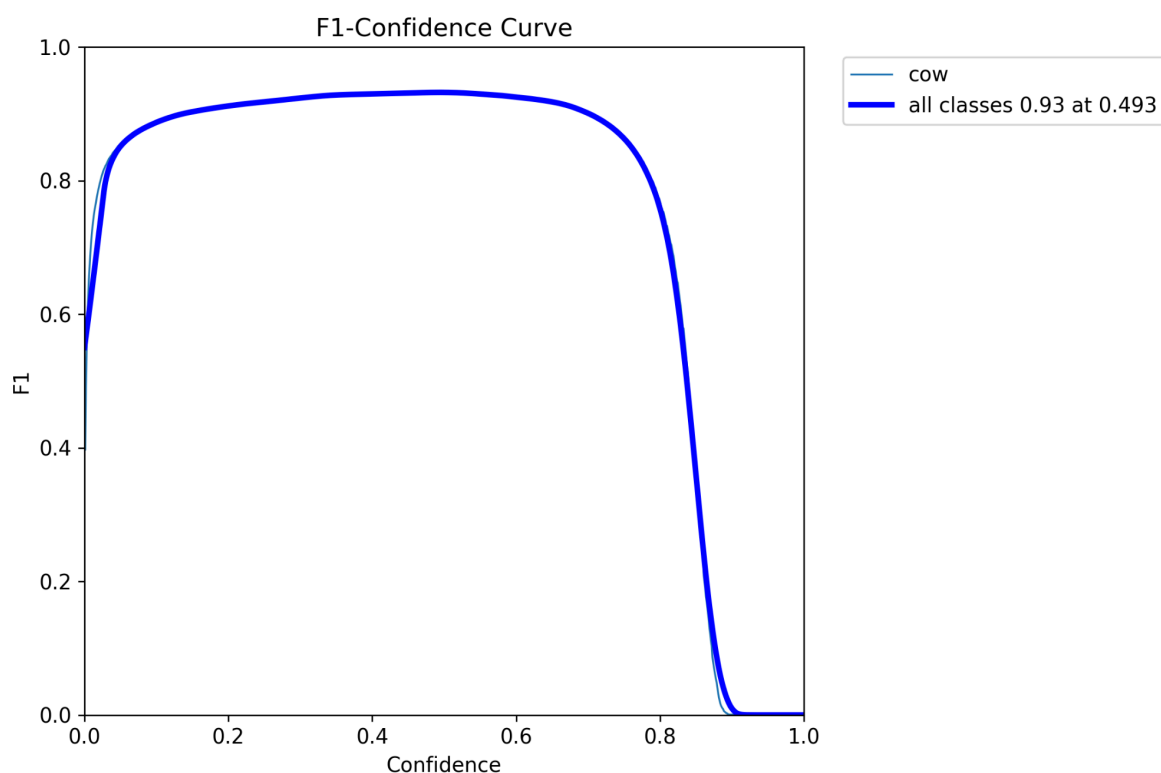
- Promedio de mAP en diferentes umbrales de IoU (de 50% a 95%)
- Evalúa la precisión de la localización de las detecciones
- Indica qué tan preciso es el modelo en diferentes niveles de exigencia
- Ayuda a validar la robustez general del modelo

Utilizando estas métricas podremos validar el objetivo de minería de datos adecuadamente. A pesar de que nuestro objetivo inicial era 50% de precisión con un modelo nocturno y 80% con un modelo diurno, con esto podremos cumplir con ambos a la vez, ya que el dataset utilizado contiene imágenes con ambos tipos de iluminación y se realizará un modelo generalizado.

4.3 Modelo

Para nuestro modelo se seleccionó **YOLO versión 8x**, que representa una variante más potente y sofisticada de la familia de detectores YOLO. YOLOv8x utiliza una arquitectura de red neuronal convolucional profunda con aproximadamente 68.2 M de parámetros, diseñada para maximizar la precisión en tareas de detección de objetos, ofreciendo un rendimiento en términos de exactitud y confiabilidad.

4.4 Resultados

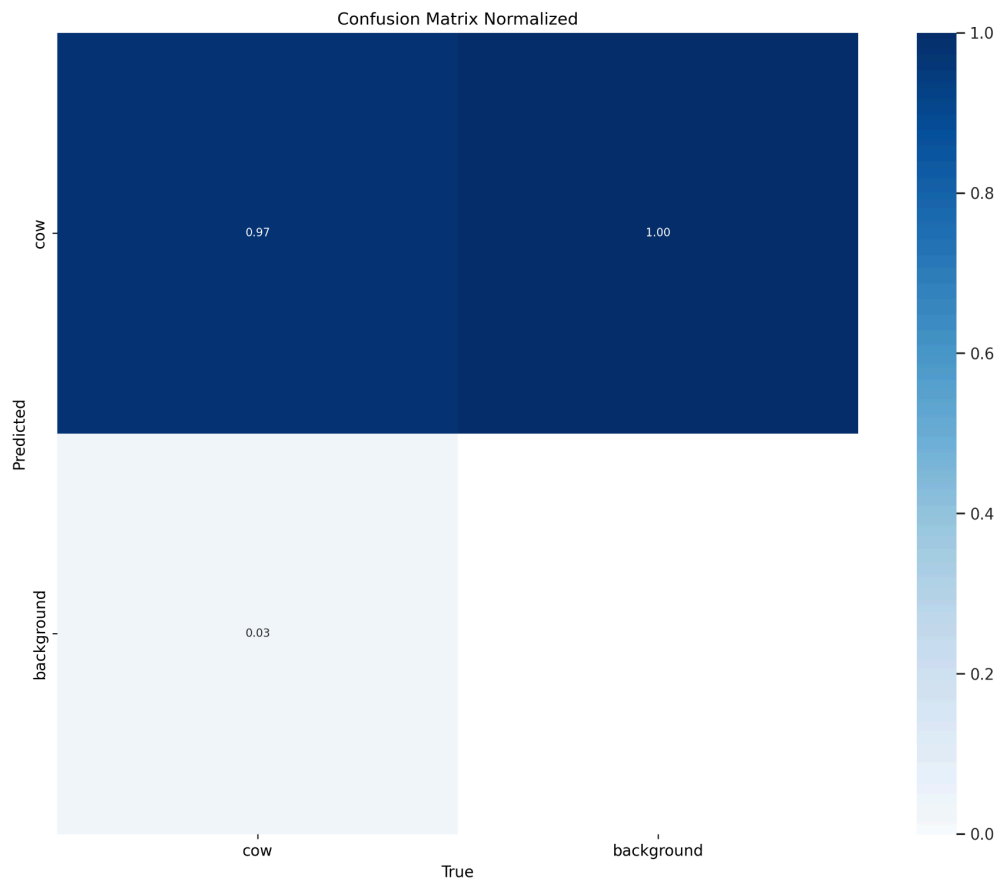


Esta gráfica es la calificación general del modelo que combina dos aspectos importantes: qué tan preciso es cuando detecta algo y cuántas cosas puede detectar del total. Un F1-score alto significa que el modelo es bueno tanto encontrando objetos como asegurándose de que sus detecciones sean correctas.

Los resultados muestran que

- ❖ El modelo alcanza su mejor rendimiento con un F1 de 0.93 (93%) cuando se configura con un umbral de confianza de 0.493. Esto significa que el modelo está acertando en 93 de cada 100 casos en su punto óptimo.

- ❖ El rendimiento se mantiene estable hasta un umbral de confianza de 0.8. Esto demuestra que el modelo es robusto y flexible en su uso.
 - Adicionalmente, demuestra que es posible ajustar el nivel de confianza en un rango amplio sin perder rendimiento.
 - La caída después de 0.8 es natural y esperada, ya que el modelo se vuelve demasiado selectivo.



Esta matriz muestra las mismas relaciones que la matriz anterior pero en porcentajes o proporciones (de 0 a 1), lo que permite entender mejor la tasa de éxito del modelo independientemente del número total de casos.

Los resultados muestran que:

- ❖ Tasa de verdaderos positivos (esquina superior izquierda): de todas las vacas reales, el modelo identifica correctamente el 97%.
- ❖ Tasa de falsos positivos (esquina superior derecha): de todos los objetos que el modelo identifica como vacas en el fondo, el error es del 100%. Sugiere que el modelo tiende a sobredetectar en áreas de fondo.

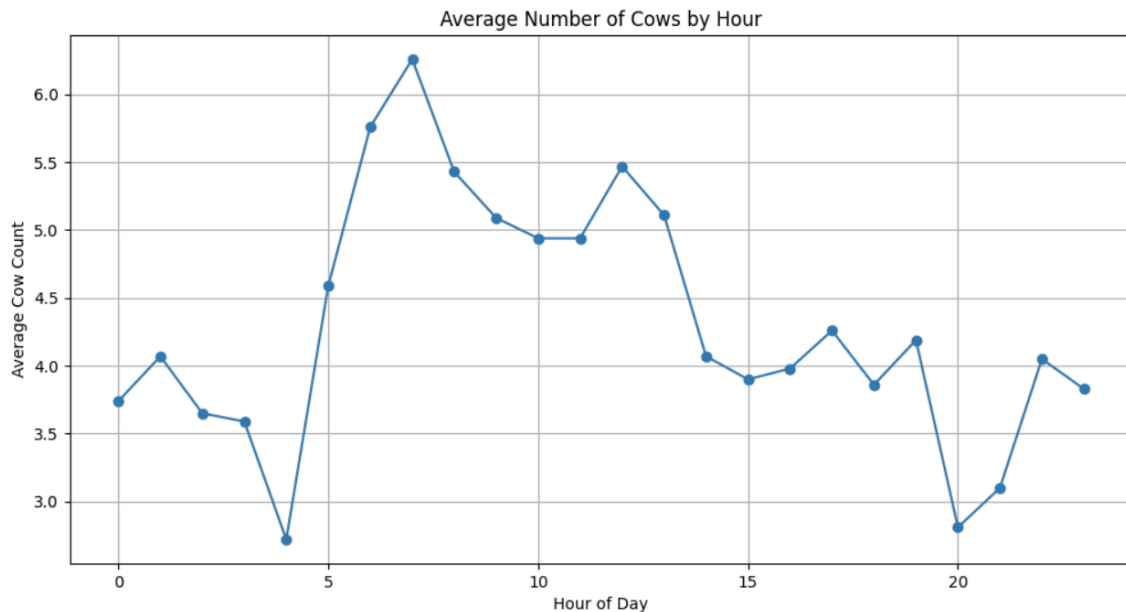
- ❖ Tasa de falsos negativos (esquina inferior izquierda): solo el 3% de las vacas reales no son detectadas. Representa una tasa de error muy baja en pérdida de detecciones.

Métrica	Inicio	Interpretación
Precisión	92%	92 de cada 100 detecciones son correctas
Recall	95%	Encuentra 95 de cada 100 vacas presentes
mAP50	95%	Excelente precisión con 50% IoU
mAP50-95	80%	Robusto en diferentes umbrales
F1-Score	93%	Alto balance entre precisión y recall

4.5 Validación de Objetivos

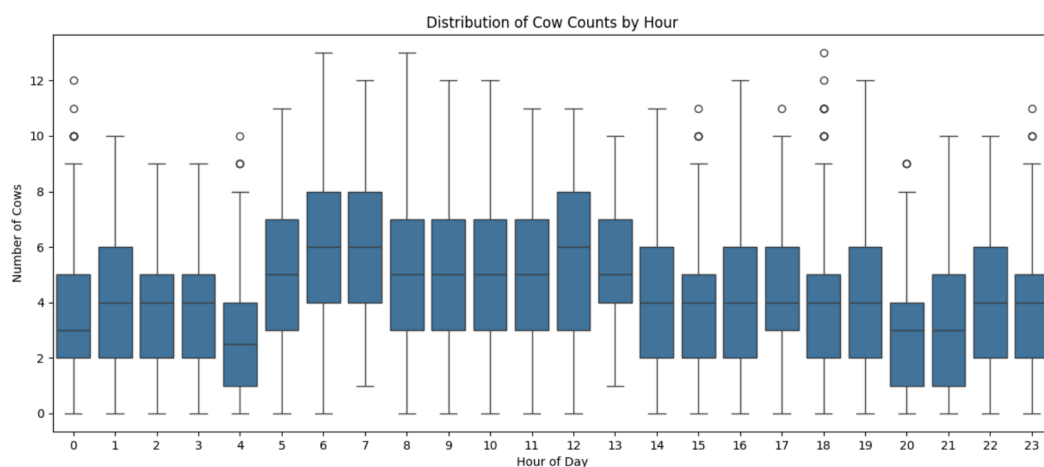
Los objetivos iniciales establecían dos umbrales: 80% de precisión para condiciones diurnas y 50% para condiciones nocturnas. El modelo que desarrollamos superó ambas metas al alcanzar un 92% de precisión general en todas las condiciones de iluminación, con una impresionante tasa de verdaderos positivos del 97% y solo un 3% de detecciones perdidas.

4.6 Hallazgos



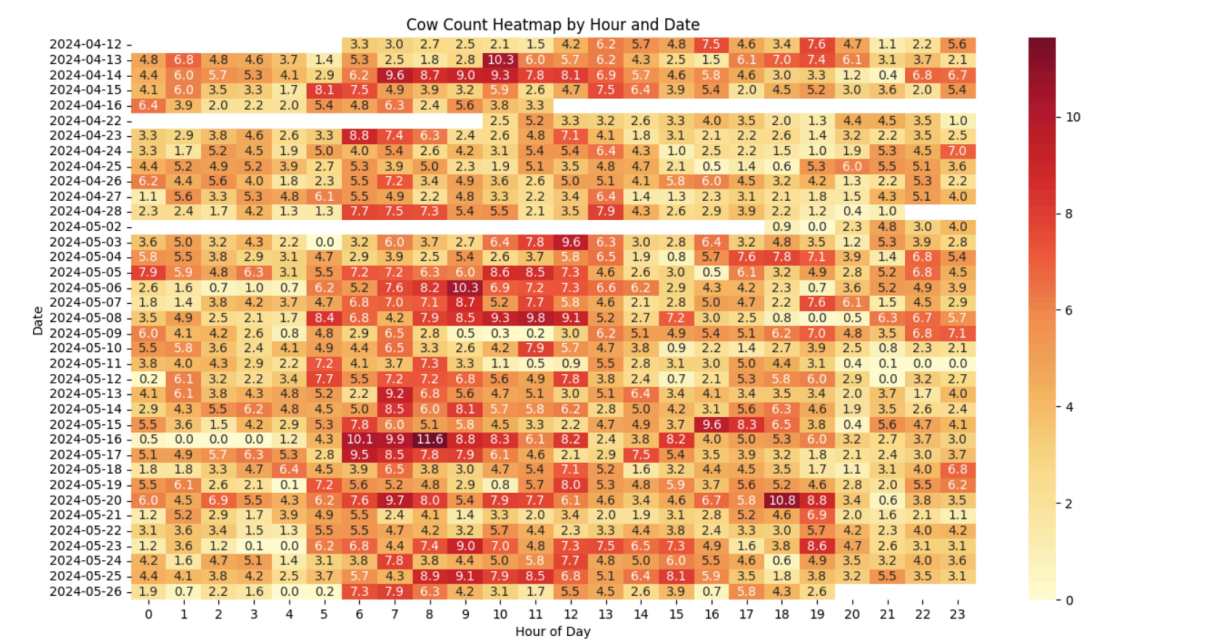
Se obtuvo una media de la cantidad de vacas formadas cada hora del día, en la primera gráfica obtenida se ve que a las 4 a.m hay una caída de forma general todos los días a un promedio de una vaca formada nada más.

Después de la caída mencionada, hay un claro incremento en el transcurso de 3 hrs llegando a un promedio de 7 vacas formadas a las 7 a.m. Durante el resto del día hay una disminución de vacas formadas hasta caer en su segundo mínimo con dos vacas promedio a las 8 p.m. El patrón mencionado anteriormente se puede visualizar en esta segunda gráfica de distribución:



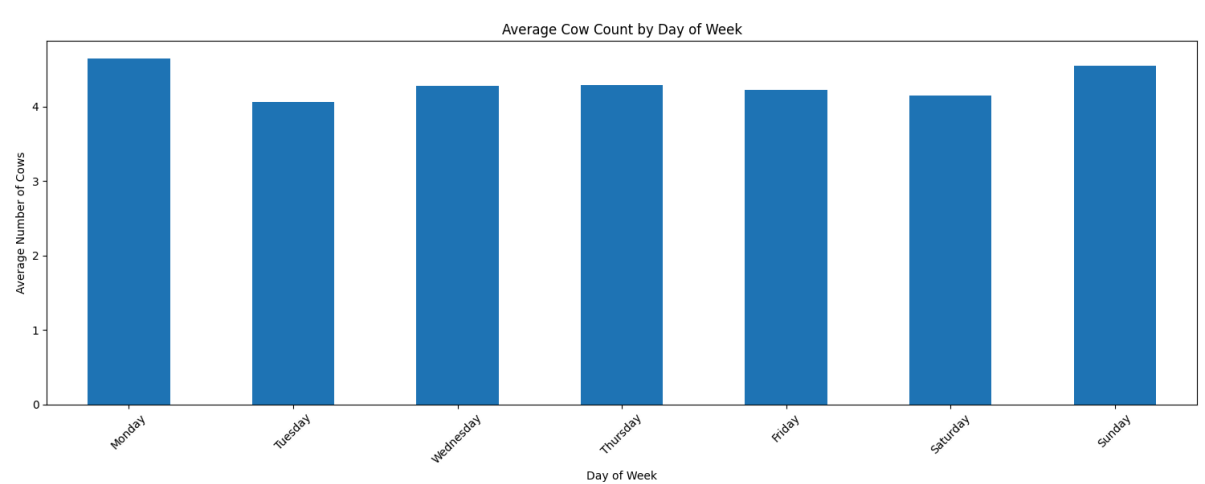
A lo largo de la mayoría de fechas se puede apreciar el mismo patrón. Ejemplos muy claros de dicho patrón pueden apreciarse específicamente en el día 2024-05-16 con mínimos

de 0 a lo largo de la 1 a.m y las 3 a.m, un máximo de 11 vacas a las 8 a.m y otra disminución extrema a 3 vacas a las 8 p.m. Y el día 2024-05-06 con mínimos de una vaca en la madrugada, un máximo de 10 vacas a las 9 a.m y una disminución a 3 vacas a las 8 p.m.

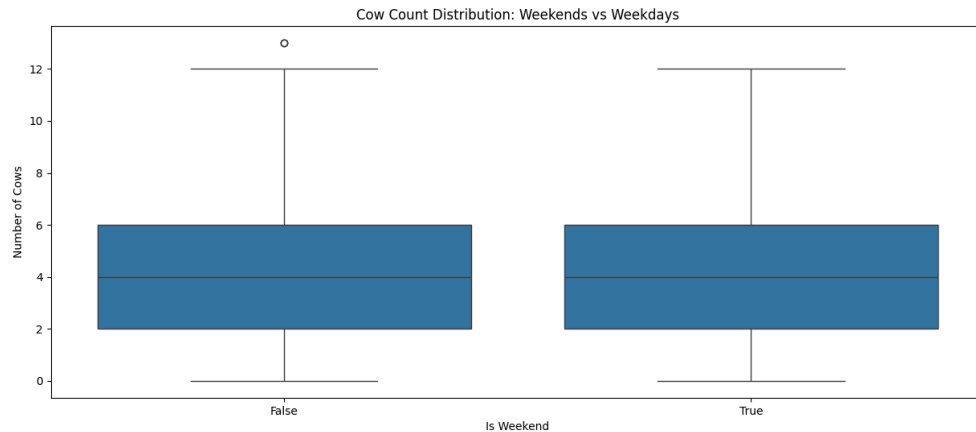


4.6.1 Hallazgos extra solicitados por el Socio Formador

Se solicitó buscar patrones entre semana y los fines de semana.



No se encontró diferencia significativa entre los 7 días de la semana.



Tampoco se ve una diferencia importante entre la cantidad de vacas promedio formada entre semana y los fines de semana, de hecho se ve casi igual.

5.0 Deployment

5.1 Requisitos previos

5.1.1 Hardware

- Cámara web compatible con Linux (conectada a /dev/video0)
- PC (Raspberry PI 4 por ejemplo, cualquier PC con estas características funciona)
 - Procesador ARM64 compatible con las dependencias de PyTorch
 - Mínimo 4GB de RAM recomendado

5.1.2 Software

- Docker Engine (versión 20.10.0 o superior)
- Docker Compose (versión 2.0.0 o superior)
- Acceso a Internet para la descarga de imágenes y modelos
- Distribución desarrollada a partir de Debian, como Ubuntu o Raspberry PI OS

5.1.3 Credenciales y accesos

- Cuenta de AWS con acceso programático
- Credenciales de AWS (Access Key ID y Secret Access Key)
- Bucket S3 previamente creado

5.2 Estructura del proyecto

```
proyecto/  
├── docker-compose.yml  
├── .env  
└── logs/
```

5.3 Configuración del entorno (En la Raspberry PI, o en la PC)

En los siguientes puntos se presentan los comandos para preparar el entorno donde se ejecutará la imagen de Docker la cual ya contiene el modelo preestablecido dentro.

El procesador de la computadora necesita ser de arquitectura ARM64 y con un sistema operativo basado en Linux, específicamente una distribución desarrollada a partir de Debian, como Ubuntu o Raspberry PI OS como ya se mencionó anteriormente.

5.3.1 Instalación de Docker Engine

Lo mejor es acceder a la documentación oficial para la instalación de Docker Engine:

- <https://docs.docker.com/engine/install/ubuntu/>

De igual manera se colocarán los comandos necesarios para la instalación aquí mismo, de encontrarse un error o de ser comandos obsoletos, favor de consultar la documentación oficial.

Primero ejecutaremos el siguiente comando para desinstalar cualquier paquete o dependencia previamente instalada en el sistema y prevenir conflictos con la nueva versión e instalación de Docker.

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc;  
do sudo apt-get remove $pkg; done
```

Ahora configuraremos el repositorio de apt para poder instalar Docker:

```
# Add Docker's official GPG key:
```

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
# Add the repository to Apt sources:
```

```
echo \
```

```
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]  
https://download.docker.com/linux/ubuntu \
```

```
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

Una vez configurado el repositorio, es momento de instalar Docker con el siguiente comando:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
docker-compose-plugin
```

Para poder ejecutar Docker sin conflictos hay que agregarlo a la lista de superusuarios:

```
sudo groupadd docker
```

```
sudo usermod -aG docker $USER
```

Y por último, para comprobar que Docker está instalado correctamente, se puede utilizar el siguiente comando:

```
sudo docker run hello-world
```

De estar todo en orden se descargará la imagen y se ejecutará mostrando un mensaje de confirmación.

5.3.2 Preparación de directorios

Ubicándonos en el lugar en donde correremos la imagen de Docker se creará un directorio con el nombre de logs/

```
mkdir -p logs
```

5.3.3 Archivo .env

Para pasarle a la imagen las credenciales de seguridad y darle acceso a la Bucket de S3 se necesita crear un archivo llamado .env con el siguiente contenido:

```
AWS_ACCESS_KEY_ID=tu_access_key_id
```

```
AWS_SECRET_ACCESS_KEY=tu_secret_access_key
```

```
AWS_REGION=tu_region
```

```
S3_BUCKET_NAME=nombre_de_tu_bucket
```

```
CAMERA_TIME=el_tiempo_entre_fotos_en_segundos
```

5.3.4 Docker Compose

Tenemos que agregar en la misma ubicación el archivo docker-compose.yml el cual está ubicado en el repositorio de GitHub, en el siguiente link:

- <https://github.com/salgue441/cow-project/blob/main/deployment/docker-compose.yml>

5.3.5 Descargar imagen de docker

La imagen se encuentra en “Docker Hub”, una plataforma donde los usuarios pueden subir sus imágenes para que sean utilizadas por el público. La imagen con el modelo corriendo se encuentra en el siguiente link:

- <https://hub.docker.com/repository/docker/sen00/auto-deploy-cow-project/general>

Por último tenemos que bajar la imagen a la Raspberry PI y necesitamos usar este comando en la misma:

```
docker pull sen00/auto-deploy-cow-project
```

5.4 AWS

Para el correcto funcionamiento del modelo es necesario generar una Bucket de S3 en el proveedor de servicios AWS.

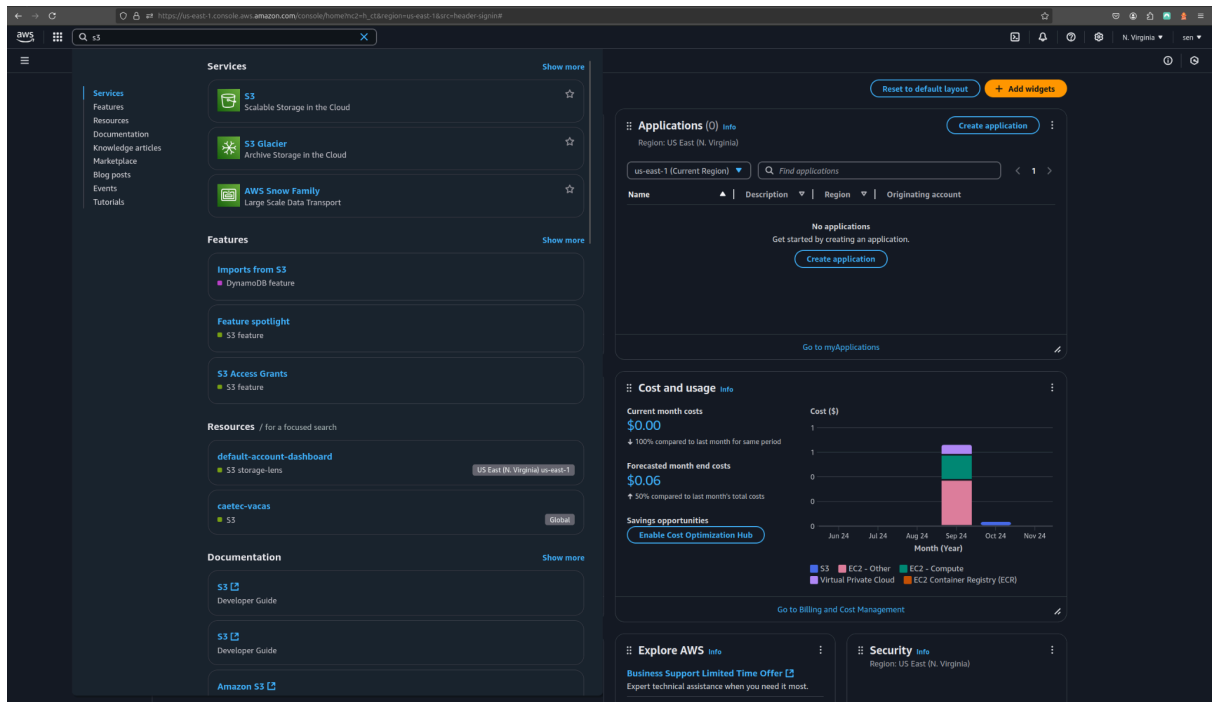
5.4.1 Creación de cuenta

Para poder acceder a los servicios de AWS primero se necesita crear una cuenta en el sistema mediante la siguiente liga:

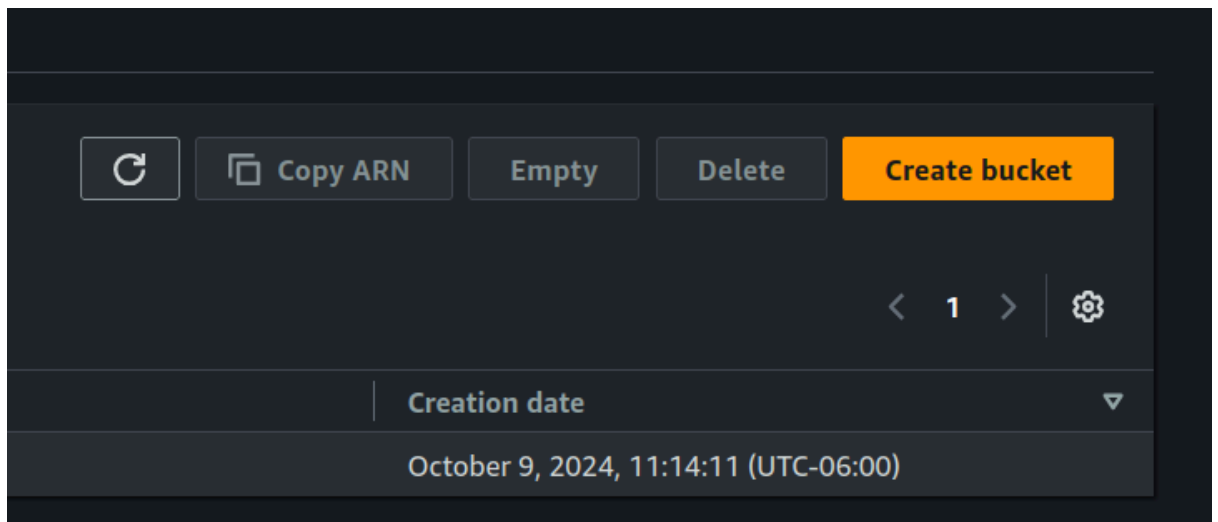
- <https://aws.amazon.com/>

5.4.2 Creación de Bucket S3

Para crear la Bucket en donde se almacenarán las imágenes procesadas, primero se necesita buscar el servicio correspondiente en la barra de búsqueda superior en la consola:



Ahora seleccionamos el botón de “Create Bucket”



En la configuración general dejamos “General purpose” y nombramos la Bucket como deseemos.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type [Info](#)

☒ **General purpose**
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

Toda la configuración adicional no la necesitamos cambiar en nada y al final le damos “Create Bucket”. El valor que pusiste en “Bucket name” en este caso “nombre-test” es el que es necesario en el archivo .env mencionado en el punto 5.1.

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

☐ Disable

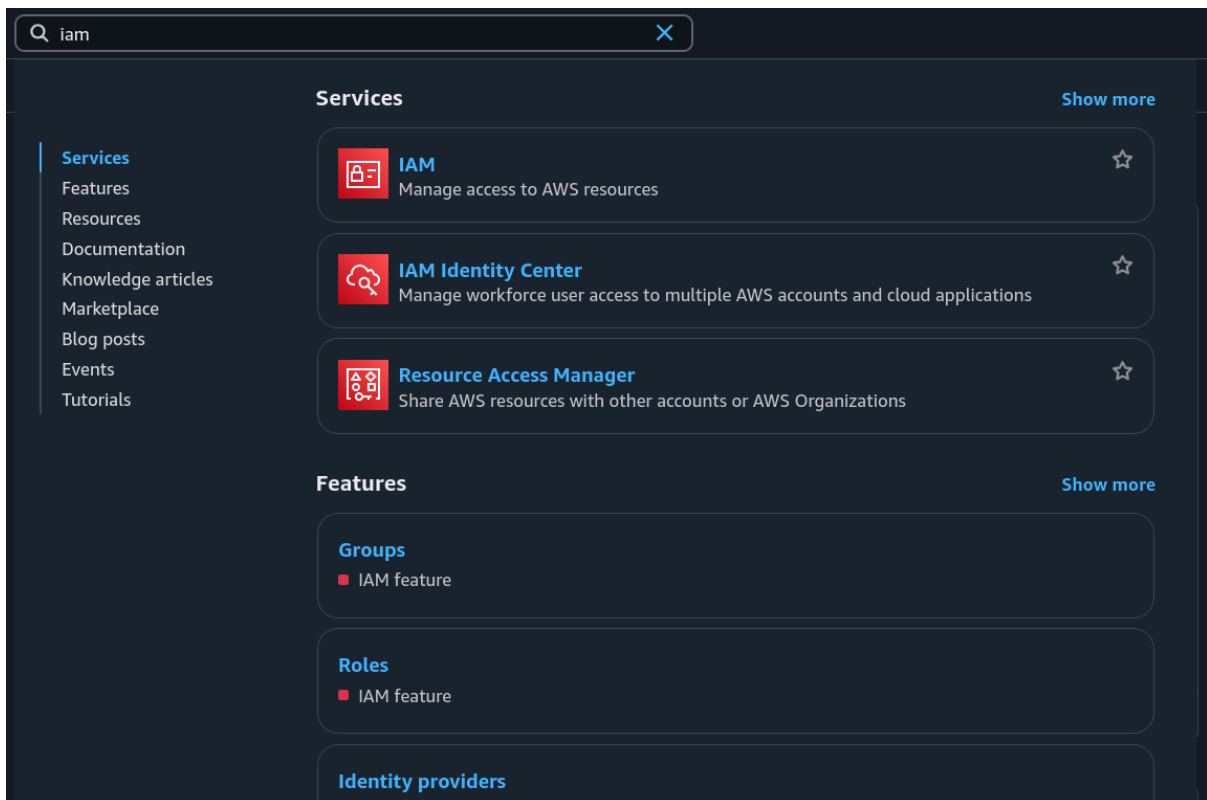
☒ Enable

► **Advanced settings**

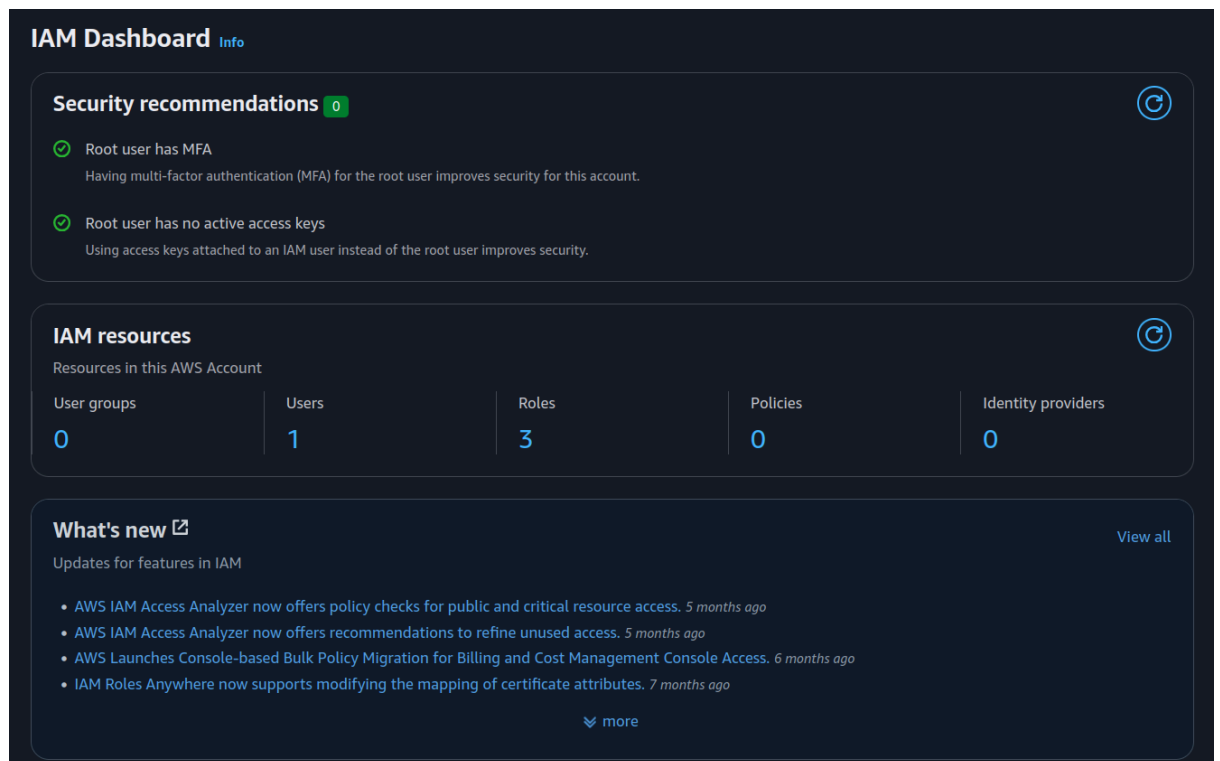
Info After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

5.4.3 Creación de cuenta para acceso externo a la Bucket

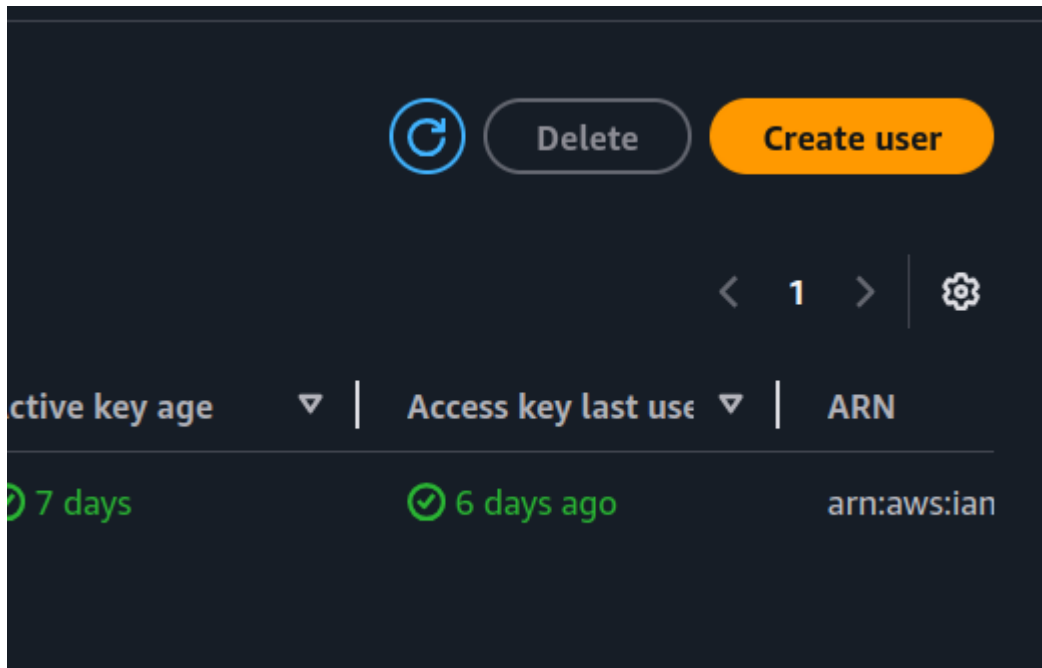
Esta vez en la barra superior buscaremos iam para acceder a las cuentas y sus permisos.



Seleccionamos “Users”.



Y una vez dentro de los usuarios seleccionamos “Create User”.



El nombre del usuario puede ser el que se desee y “Provide user access...” se dejará desmarcado. Ahora se seleccionará “Next”.

A screenshot of the 'Specify user details' form in the AWS IAM console. The form has a section titled 'User details' with a 'User name' input field. Below the input field, there is a note: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)'. There is a checkbox labeled 'Provide user access to the AWS Management Console - optional' with a sub-note: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.' At the bottom right, there are 'Cancel' and 'Next' buttons.

Ahora seleccionaremos “Attach policies directly”.

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 (1280)

Filter by Type
All types

< 1 2 3 4 5 6 7 ... 64 >

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	AccessAnalyzerServiceRolePolicy	AWS managed	0
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function	0
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	0
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeans...	AWS managed	0
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	0
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	0

Y seleccionamos el permiso “AmazonS3FullAccess”.

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 (1/1280)

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

All types 14 matches

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	AmazonDMSRedshiftS3Role	AWS managed	0
<input checked="" type="checkbox"/>	AmazonS3FullAccess	AWS managed	1
<input type="checkbox"/>	AmazonS3ObjectLambdaExecutionRol...	AWS managed	0
<input type="checkbox"/>	AmazonS3OutpostsFullAccess	AWS managed	0
<input type="checkbox"/>	AmazonS3OutpostsReadOnlyAccess	AWS managed	0
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed	0
<input type="checkbox"/>	AWSBackupServiceRolePolicyForS3Ba...	AWS managed	0
<input type="checkbox"/>	AWSBackupServiceRolePolicyForS3Res...	AWS managed	0
<input type="checkbox"/>	AWSQuickSetupSSMDeploymentS3Bu...	AWS managed	0
<input type="checkbox"/>	AWSS3OnOutpostsServiceRolePolicy	AWS managed	0
<input type="checkbox"/>	IVSRecordToS3	AWS managed	0
<input type="checkbox"/>	QuickSightAccessForS3StorageManag...	AWS managed	0
<input type="checkbox"/>	S3StorageLensServiceRolePolicy	AWS managed	0
<input type="checkbox"/>	S3UnlockBucketPolicy	AWS managed	0

► Set permissions boundary - optional

Seleccionamos en la parte inferior “Next” y en la siguiente ventana seleccionamos “Create User”.

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name user-test	Console password type None	Require password reset No
------------------------	-------------------------------	------------------------------

Permissions summary

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

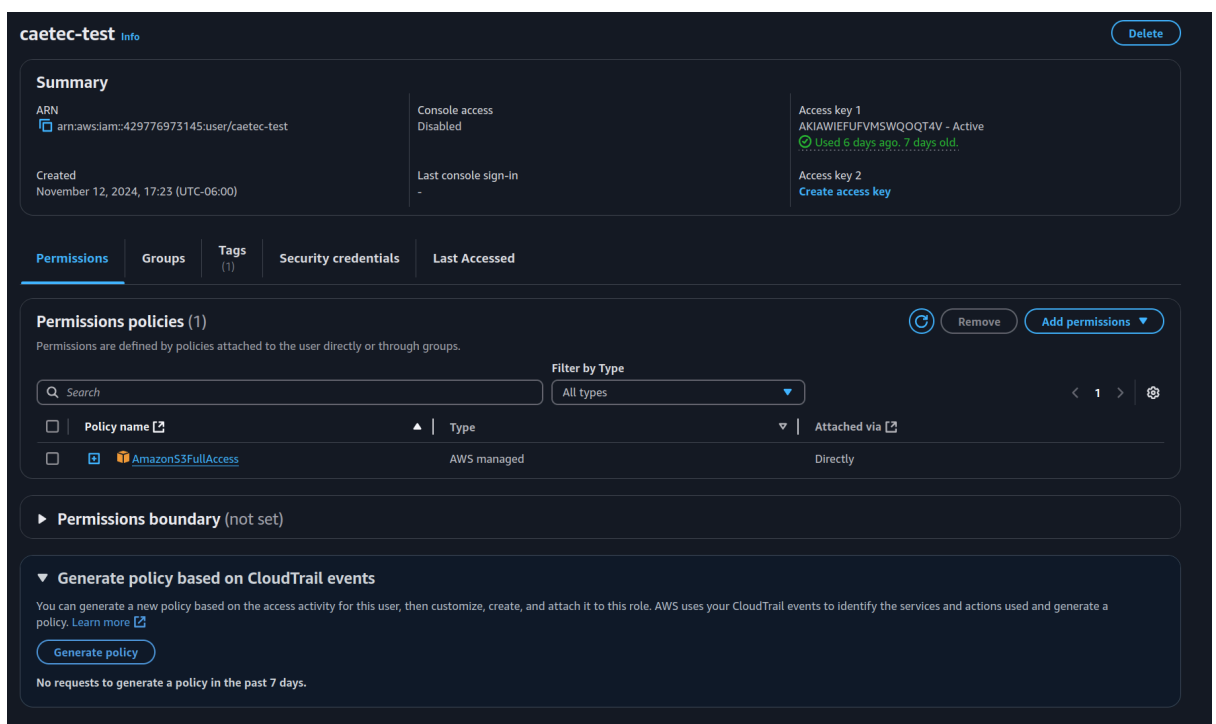
[Cancel](#) [Previous](#) [Create user](#)

5.4.4 Creación de claves de acceso

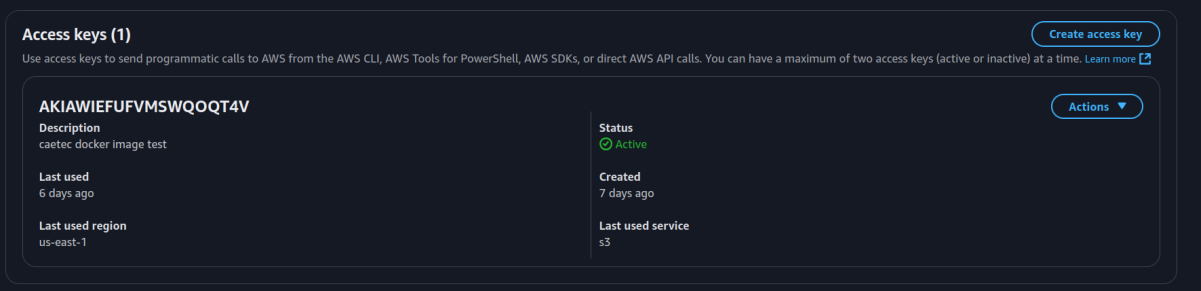
Volvemos a buscar en la parte superior iam y accedemos a los usuarios de nuevo. Ahora seleccionamos el nombre del usuario recién creado.



En la nueva ventana se puede ver la información general del usuario y el permiso de S3 que se le añadió durante su creación. En esta ventana seleccionamos el apartado de “Security Credentials”.



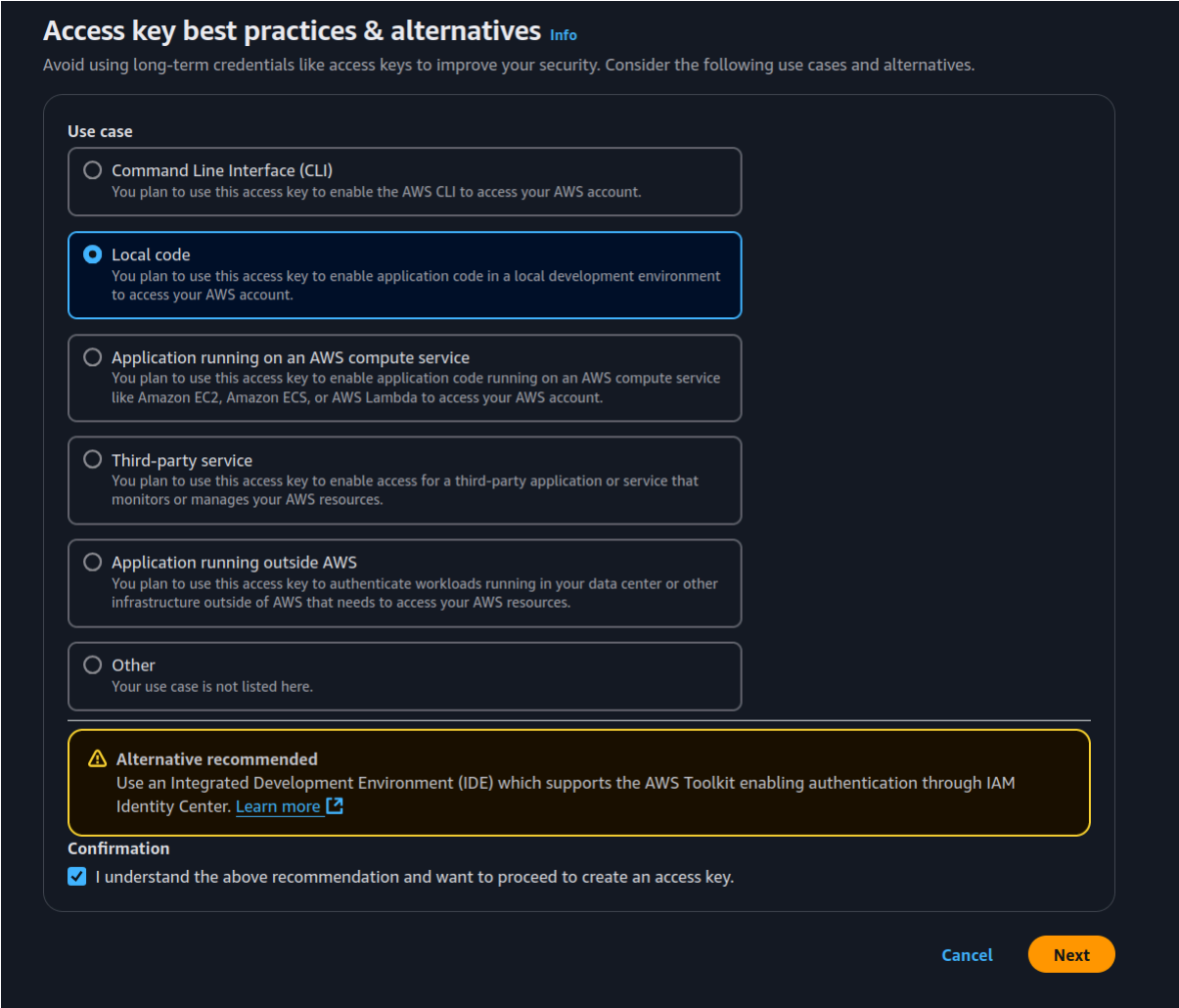
Más abajo en “Access Keys” seleccionamos “Create access key”.



The screenshot shows the 'Access keys (1)' page in the AWS IAM console. At the top right is a 'Create access key' button. Below it is a table with one access key. The key ID is AKIAWIEFUFVMSWQQQT4V. The description is 'caetec docker image test'. The status is 'Active'. It was created 7 days ago and last used 6 days ago. The last used region is 'us-east-1' and the last used service is 's3'. An 'Actions' dropdown menu is visible to the right of the key details.

Access Key ID	Description	Status	Created	Last used	Last used region	Last used service
AKIAWIEFUFVMSWQQQT4V	caetec docker image test	Active	7 days ago	6 days ago	us-east-1	s3

En la nueva ventana seleccionamos “Local code” y marcamos “I understand the above...”.



The screenshot shows the 'Access key best practices & alternatives' dialog box. It contains a section for 'Use case' with several radio button options. The 'Local code' option is selected. Below this is a section for 'Alternative recommended' which suggests using an IDE. At the bottom is a 'Confirmation' section with a checked checkbox indicating understanding of the recommendation. 'Cancel' and 'Next' buttons are at the bottom right.

Access key best practices & alternatives [Info](#)

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

- ☐ Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
- ☒ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- ☐ Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- ☐ Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- ☐ Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- ☐ Other
Your use case is not listed here.

Alternative recommended
Use an Integrated Development Environment (IDE) which supports the AWS Toolkit enabling authentication through IAM Identity Center. [Learn more](#)

Confirmation
☒ I understand the above recommendation and want to proceed to create an access key.

[Cancel](#) [Next](#)

Ponemos cualquier descripción que nos ayude a identificar la clave de acceso en la consola y seleccionamos “Create access key”.

Set description tag - optional [Info](#)

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @



[Cancel](#) [Previous](#) [Create access key](#)

En la última ventana nos presentan el “Access key” y el “Secret access key”.

et access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Retrieve access keys [Info](#)

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
 AKIAWIEFUFVMVQY3CM63	 ***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

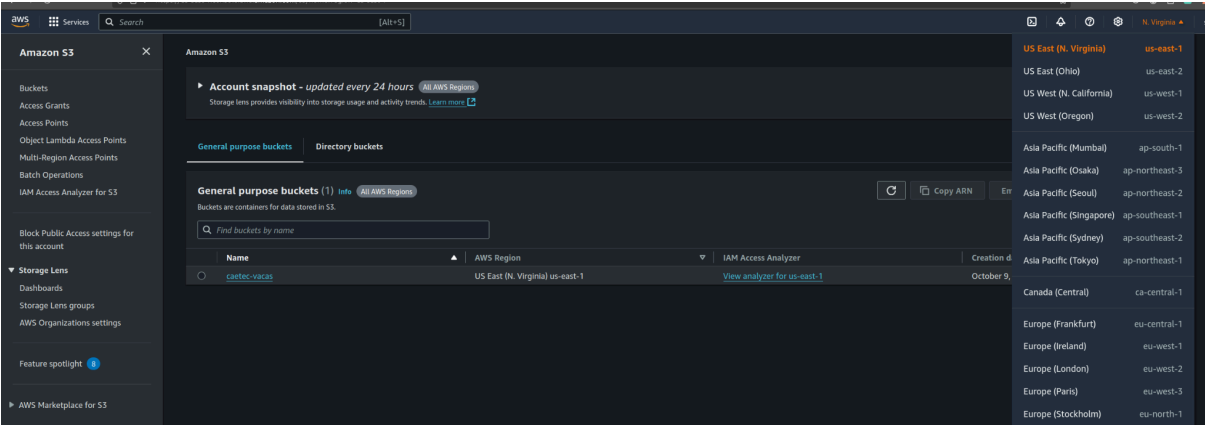
For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

Estos valores deberán ser guardados, ya que el “Secret access key no podrá ser consultado una vez que se seleccione “Done”. Estos valores son los que van en el .env creado en el punto 4.3.

5.4 Región

AWS permite el despliegue de infraestructura en distintas regiones o zonas del planeta, para ver la región en la que se encuentra la Bucket, en el menú s3 al que habíamos accedido en el paso 5.1, se puede revisar la esquina superior derecha.



En este caso, la región sería “us-east-1”. Este valor también es necesario en el .env mencionado en el punto 5.1.

US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1

5.5 Ejecución (En la Raspberry PI, o en la PC)

Una vez realizados los pasos anteriores, nos ubicamos en el índice del proyecto, o la carpeta principal del mismo. En el caso de la estructura de proyecto presentada antes, este sería en la carpeta proyecto/ y una vez ahí ejecutaremos el comando

- `docker composer up -d`

5.5.1 Verificar estado del contenedor

Para poder comprobar el estado del contenedor y del modelo en ejecución se pueden ejecutar los siguientes dos comandos:

- `docker ps`
- `docker logs cow-detector`

5.5.2 Verificar logs del contenedor

Para revisar los logs del contenedor en cuestión se puede utilizar el siguiente comando:

- `docker logs -f cow-detector`

5.5.3 Revisar AWS

Una vez que el contenedor y el modelo se ejecuten sin problemas, las imágenes serán clasificadas y subidas a la Bucket S3 en AWS de manera automática en carpetas de 0 a n, siendo n el número de vacas encontradas en la imagen.

Se puede acceder al menú de S3 en la consola de AWS, en la cual se podrán ver las carpetas y las imágenes subidas, con las bounding boxes correspondientes.

5.5.4 Detener contenedor y modelo

Para detener el contenedor de Docker solo se necesita ubicar en el índice del proyecto, en el mismo lugar donde se encuentra el docker-compose.yml mencionado en el paso 4.4. Una vez ahí solo se necesita usar el comando:

- `docker compose down`