



Informe de Avance – Airflow en AWS

1. Infraestructura inicial (CloudFormation)

Primero **creé la VPC** utilizando un archivo `.yml` de CloudFormation. En este template **agregué subnets públicas y privadas**, un NAT Gateway y los endpoints necesarios para la comunicación. Validé que el stack se completara correctamente y que la VPC apareciera en la consola.

👉 Código: incluir el `.yml` de CloudFormation.

AWSTemplateFormatVersion: '2010-09-09'

Description: >-

VPC + 2 public subnets + 2 private subnets + IGW + NAT + EFS with AccessPoint

Parameters:

VpcCidr:

Type: String

Default: 10.10.0.0/16

PublicSubnetCidr1:

Type: String

Default: 10.10.0.0/24

PublicSubnetCidr2:

Type: String

Default: 10.10.1.0/24

PrivateSubnetCidr1:

Type: String

Default: 10.10.100.0/24

PrivateSubnetCidr2:

Type: String

Default: 10.10.101.0/24

Resources:

VPC

AirflowVPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VpcCidr

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: airflow-vpc

Internet Gateway

InternetGateway:

Type: AWS::EC2::InternetGateway

Properties:

Tags:

- Key: Name

Value: airflow-igw

AttachGateway:

Type: AWS::EC2::VPGatewayAttachment

Properties:

VpcId: !Ref AirflowVPC

```
InternetGatewayId: !Ref InternetGateway
```

```
# Public Subnets
```

```
PublicSubnet1:
```

```
Type: AWS::EC2::Subnet
```

```
Properties:
```

```
VpcId: !Ref AirflowVPC
```

```
CidrBlock: !Ref PublicSubnetCidr1
```

```
AvailabilityZone: !Select [ 0, !GetAZs " ]
```

```
MapPublicIpOnLaunch: true
```

```
Tags:
```

```
- Key: Name
```

```
Value: airflow-public-subnet-1
```

```
PublicSubnet2:
```

```
Type: AWS::EC2::Subnet
```

```
Properties:
```

```
VpcId: !Ref AirflowVPC
```

```
CidrBlock: !Ref PublicSubnetCidr2
```

```
AvailabilityZone: !Select [ 1, !GetAZs " ]
```

```
MapPublicIpOnLaunch: true
```

```
Tags:
```

```
- Key: Name
```

```
Value: airflow-public-subnet-2
```

```
# Private Subnets
```

PrivateSubnet1:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref AirflowVPC

CidrBlock: !Ref PrivateSubnetCidr1

AvailabilityZone: !Select [0, !GetAZs "]

MapPublicIpOnLaunch: false

Tags:

- Key: Name

Value: airflow-private-subnet-1

PrivateSubnet2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref AirflowVPC

CidrBlock: !Ref PrivateSubnetCidr2

AvailabilityZone: !Select [1, !GetAZs "]

MapPublicIpOnLaunch: false

Tags:

- Key: Name

Value: airflow-private-subnet-2

Route Tables

PublicRouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref AirflowVPC

Tags:

- Key: Name

- Value: airflow-public-rt

PublicRoute:

Type: AWS::EC2::Route

DependsOn: AttachGateway

Properties:

- RouteTableId: !Ref PublicRouteTable

- DestinationCidrBlock: 0.0.0.0/0

- GatewayId: !Ref InternetGateway

PublicSubnet1RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

- SubnetId: !Ref PublicSubnet1

- RouteTableId: !Ref PublicRouteTable

PublicSubnet2RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

- SubnetId: !Ref PublicSubnet2

- RouteTableId: !Ref PublicRouteTable

NAT and EIP for private subnets internet access

NatEIP:

Type: AWS::EC2::EIP

DependsOn: AttachGateway

Properties:

Domain: vpc

NatGateway:

Type: AWS::EC2::NatGateway

Properties:

AllocationId: !GetAtt NatEIP.AllocationId

SubnetId: !Ref PublicSubnet1

Tags:

- Key: Name

Value: airflow-nat

PrivateRouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref AirflowVPC

Tags:

- Key: Name

Value: airflow-private-rt

PrivateDefaultRoute:

Type: AWS::EC2::Route

Properties:

RouteTableId: !Ref PrivateRouteTable

DestinationCidrBlock: 0.0.0.0/0

NatGatewayId: !Ref NatGateway

PrivateSubnet1RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PrivateSubnet1

RouteTableId: !Ref PrivateRouteTable

PrivateSubnet2RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

SubnetId: !Ref PrivateSubnet2

RouteTableId: !Ref PrivateRouteTable

Security Groups

SGWeb:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Allow HTTP 8080 from anywhere (change before prod)

VpcId: !Ref AirflowVPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 8080

ToPort: 8080

CidrIp: 0.0.0.0/0

Tags:

- Key: Name

- Value: sg-airflow-web

SGPrivate:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Internal communication for scheduler/other

VpcId: !Ref AirflowVPC

SecurityGroupIngress:

- IpProtocol: -1

- FromPort: -1

- ToPort: -1

- SourceSecurityGroupId: !Ref SGWeb

Tags:

- Key: Name

- Value: sg-airflow-private

SGEFS:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: EFS mount (NFS)

VpcId: !Ref AirflowVPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 2049

ToPort: 2049

CidrIp: 10.10.0.0/16

Tags:

- Key: Name

- Value: sg-airflow-efs

EFS filesystem

AirflowEFS:

Type: AWS::EFS::FileSystem

Properties:

Encrypted: false

PerformanceMode: generalPurpose

FileSystemTags:

- Key: Name

- Value: airflow-efs

EFSAccessPoint:

Type: AWS::EFS::AccessPoint

Properties:

FileSystemId: !Ref AirflowEFS

PosixUser:

Uid: "5000"

Gid: "5000"

RootDirectory:

CreateInfo:

OwnerUid: "5000"

OwnerGid: "5000"

Permissions: "0755"

Path: "/airflow"

Mount targets for EFS in private subnets

MountTarget1:

Type: AWS::EFS::MountTarget

Properties:

FileSystemId: !Ref AirflowEFS

SubnetId: !Ref PrivateSubnet1

SecurityGroups:

- !Ref SGEFS

MountTarget2:

Type: AWS::EFS::MountTarget

Properties:

FileSystemId: !Ref AirflowEFS

SubnetId: !Ref PrivateSubnet2

SecurityGroups:

- !Ref SGEFS

Outputs:

VpcId:

Description: VPC Id

Value: !Ref AirflowVPC

PublicSubnet1Id:

Description: Public subnet 1 ID

Value: !Ref PublicSubnet1

PublicSubnet2Id:

Description: Public subnet 2 ID

Value: !Ref PublicSubnet2

PrivateSubnet1Id:

Description: Private subnet 1 ID

Value: !Ref PrivateSubnet1

PrivateSubnet2Id:

Description: Private subnet 2 ID

Value: !Ref PrivateSubnet2

EFSFileSystemId:

Description: EFS filesystem ID

Value: !Ref AirflowEFS

EFSAccessPointId:

Description: EFS AccessPoint ID

Value: !Ref EFSAccessPoint

SGWebId:

Description: Security Group Web

Value: !Ref SGWeb

SGPrivateId:

Description: Security Group Private

Value: !Ref SGPrivate

SGEFSId:

Description: Security Group EFS

Value: !Ref SGEFS

2. Cluster y Servicio ECS

Luego configuré un cluster ECS llamado H-airflow-cluster. En este cluster coloque un servicio Fargate con la Task Definition H-airflow-standalone. En la Task Definition agregué el contenedor airflow-standalone usando la imagen oficial de Airflow y configuré variables de entorno (ejecutor, fernet key, conexión SQLite).

👉 Captura recomendada: consola ECS mostrando el cluster y el servicio activo.👉 Código: JSON de la Task Definition. SOLO ES UN SERVICIO, EL OTRO ES UN INTENTO PASADO Y FALLIDO. (ignorenlo)

The screenshot shows the AWS Elastic Container Service (ECS) console interface. At the top, there are two tabs: "Amazon Elastic Container Service" and "Clústeres". The "Clústeres" tab is selected, displaying two clusters: "H-airflow-cluster" and "airflow-cluster".

H-airflow-cluster Details:

- Información general sobre el clúster:** ARN: arn:aws:ecs:us-east-1:1590953801115:cluster/H-airflow-cluster, Estado: Activo.
- Servicios:** Vacíando (0), Activo (2).
- Tareas:** Pendiente (0), Ejecutando (1).
- Supervisión de CloudWatch:** Valor predeterminado.
- Instancias de contendedor registradas:** -

Services Tab: Shows a list of services: "H-airflow-scheduler-service" (ARN: arn:aws:ecs:us-east-1:1590953801115:service/H-airflow-scheduler-service, Estado: Activo, Estrategia de programación: REPLICAS, Tareas: 0/1 ejecutando), "H-airflow-standalone-service-p4p2c4ub" (ARN: arn:aws:ecs:us-east-1:1590953801115:service/H-airflow-standalone-service-p4p2c4ub, Estado: Activo, Estrategia de programación: REPLICAS, Tareas: 0/1 ejecutando), and "H-airflow-standalone-service-p4p2c4ub" (ARN: arn:aws:ecs:us-east-1:1590953801115:service/H-airflow-standalone-service-p4p2c4ub, Estado: Activo, Estrategia de programación: FARGATE, Tareas: 0/1 ejecutando).

Codigo Json del Task definition:

```
{  
  "compatibilities": [  
    "EC2",  
    "FARGATE",  
    "MANAGED_INSTANCES"  
,  
  ],  
  "containerDefinitions": [  
    {  
      "command": [  
        "standalone"  
,  
        "cpu": 0,  
        "environment": [  
          {  
            "name": "AIRFLOW__CORE__EXECUTOR",  
            "value": "SequentialExecutor"  
          },  
          {  
            "name": "AIRFLOW__CORE__LOAD_EXAMPLES",  
            "value": "False"  
          },  
          {  
            "name": "AIRFLOW__CORE__FERNET_KEY",  
            "value": "RlbPxbsS_OTsRJrpDt63rfL70qBmA0pFgXqX8zxjo5U="  
          }  
        ]  
      }  
    ]  
  ]  
}
```

```
{  
    "name": "AIRFLOW__DATABASE__SQLALCHEMY_CONN",  
    "value": "sqlite:///opt/airflow/airflow.db"  
}  
,  
    "environmentFiles": [],  
    "essential": true,  
    "image":  
"590953801135.dkr.ecr.us-east-1.amazonaws.com/airflow-official@sha256:5d29ed3ba4065  
2e29e808dfc071a7bc3b5e7ce2aa72f6675de0cebc756b96941",  
    "logConfiguration": {  
        "logDriver": "awslogs",  
        "options": {  
            "awslogs-group": "/ecs/H-airflow-standalone",  
            "awslogs-create-group": "true",  
            "awslogs-region": "us-east-1",  
            "awslogs-stream-prefix": "airflow"  
        },  
        "secretOptions": []  
    },  
    "mountPoints": [  
    {  
        "containerPath": "/opt/airflow/dags",  
        "readOnly": false,  
        "sourceVolume": "airflow-dags-efs"  
    }  
,  
],
```

```
        "name": "airflow-standalone",
        "portMappings": [
            {
                "appProtocol": "http",
                "containerPort": 8080,
                "hostPort": 8080,
                "name": "airflow-standalone-8080-tcp",
                "protocol": "tcp"
            }
        ],
        "systemControls": [],
        "ulimits": [],
        "volumesFrom": []
    },
    ],
    "cpu": "512",
    "enableFaultInjection": false,
    "executionRoleArn": "arn:aws:iam::590953801135:role/LabRole",
    "family": "H-airflow-standalone",
    "memory": "3072",
    "networkMode": "awsvpc",
    "placementConstraints": [],
    "registeredAt": "2025-11-17T03:53:27.595Z",
    "registeredBy":
"arn:aws:sts::590953801135:assumed-role/voclabs/user4108430=mijail.saltsin@utec.edu.pe",
    "requiresAttributes": [
```

```
{  
    "name": "ecs.capability.execution-role-awslogs"  
},  
{  
    "name": "com.amazonaws.ecs.capability.ecr-auth"  
},  
{  
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"  
},  
{  
    "name": "com.amazonaws.ecs.capability.task-iam-role"  
},  
{  
    "name": "ecs.capability.execution-role-ecr-pull"  
},  
{  
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"  
},  
{  
    "name": "ecs.capability.task-eni"  
},  
{  
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"  
},  
{  
    "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
```

```
        },
        {
            "name": "ecs.capability.efsAuth"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
        },
        {
            "name": "ecs.capability.efs"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
        }
    ],
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "revision": 6,
    "runtimePlatform": {
        "cpuArchitecture": "X86_64",
        "operatingSystemFamily": "LINUX"
    },
    "status": "ACTIVE",
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:590953801135:task-definition/H-airflow-standalone:6",
    "taskRoleArn": "arn:aws:iam::590953801135:role/LabRole",
    "volumes": [
```

```
{  
    "efsVolumeConfiguration": {  
        "authorizationConfig": {  
            "accessPointId": "fsap-000a3d52508910ed4",  
            "iam": "DISABLED"  
        },  
        "fileSystemId": "fs-0cf3edd46d229de22",  
        "rootDirectory": "/",  
        "transitEncryption": "ENABLED"  
    },  
    "name": "airflow-dags-efs"  
},  
],  
"tags": []  
}
```

3. Application Load Balancer (ALB)

Después implementé un **ALB** para exponer la UI de Airflow en el puerto 8080. Agregué **listeners** y **target groups** y validé que las tareas quedaran registradas.

👉 Captura recomendada: consola ALB mostrando listener 8080 y target group.

Aquí los importantes son los 3 Security Groups, cada uno para su función.

Groups of security (3/9) Information

Name	ID de grupo de seguridad	Nombre del grupo de seguridad	ID de la VPC	Descripción	Propietario	Número de reglas de entrada
<input checked="" type="checkbox"/> sg-airflow-web	sg-0bf4d6609030b156	H-airflow-network-SGWeb-pfU4tLUMPIO	vpc-014bbab05414a506d	Allow HTTP 8080 from anywhere (chan...	9095353001135	1 Entrada de permiso
<input checked="" type="checkbox"/> sg-airflow-private	sg-09aae083068850c2	H-airflow-network-SGPrivate-YBrayxOa...	vpc-014bbab05414a506d	Internal communication for scheduler/o...	9095353001135	1 Entrada de permiso
-	sg-002423770fb6a10	default	vpc-0a57dbfc190a46a8	default VPC security group	9095353001135	1 Entrada de permiso
-	sg-06c0d04a93a220dc	airflow-db-access-sg	vpc-014bbab05414a506d	Created by RDS management console	9095353001135	1 Entrada de permiso
<input checked="" type="checkbox"/> sg-airflow-efs	sg-0c79967c0f4c2a59	H-airflow-network-SGEFS-FVFMifezoyzs	vpc-014bbab05414a506d	EFS mount (NFS)	9095353001135	2 Entradas de permisos
-	sg-0645d7928a24032a8	launch-wizard-2	vpc-0a57dbfc190a46a8	launch-wizard-2 created 2025-11-16T0...	9095353001135	3 Entradas de permisos
-	sg-07097f12b053223fe	launch-wizard-1	vpc-0a57dbfc190a46a8	launch-wizard-1 created 2025-11-15T1...	9095353001135	3 Entradas de permisos
-	sg-0e7e7a3b93b8e5b1	default	vpc-014bbab05414a506d	default VPC security group	9095353001135	1 Entrada de permiso
-	sg-093ba6abedef8f80a	default	vpc-0c1ab3bec11093e05	default VPC security group	9095353001135	1 Entrada de permiso

Introducing token validation of JWTs for ALB
Authenticate machine-to-machine and service-to-service communications by validating JSON Web Tokens (JWTs) directly at the load balancer level.

H-airflow-alb

Detalles

Tipo de equilibrador de carga	Estado	VPC	Tipo de dirección IP del equilibrador de carga
APLICATION	Activo	vpc-014bbab05414a506d	IPv4
EQUILIBRADOR	Zona hospedada	Zonas de disponibilidad	Fecha creada
Internet-facing	2SS53DQTRQ7X7K	us-east-1b (use1-az1) us-east-1c (use1-az2)	16 de noviembre de 2025, 17:07 (UTC-05:00)
ARN del equilibrador de carga	Nombre de DNS	Nombre de DNS interno	
arn:aws:elasticloadbalancing:us-east-1:590953001135:loadbalance/app/H-airflow-alb/bf0b4084c97e4587	H-airflow-alb-2A0464167.us-east-1.elb.amazonaws.com (Registro A)		

Agentes de escucha y reglas

Protocolo/Port	Acción predeterminada	Reglas	ARN	Política de seguridad	Certificado SSL/TLS predeterminado	mTLS	Trust store
HTTP:8000	Reenviar al grupo de destino	tg-airflow-8000-ip-correto-1:1 (100%)	1 regla	ARN	No aplicable	No aplicable	No aplicable
	Permanencia del grupo de destino:	desactivada					

4. EFS para DAGs

Creé un **EFS** con Access Point en **/airflow**. En el bastion EC2 monté el **EFS** en **/mnt/airflow/airflow** y ahí **copié mi archivo demo_dag.py**. Validé con **ls -la** que el archivo estuviera en el sistema de archivos compartido.

👉 Captura recomendada: consola EFS mostrando Access Point y bastion EC2 mostrando **demo_dag.py**.

Sistemas de archivos (2)

Nombre	ID del sistema de archivos	Cifrado	Tamaño total	Tamaño en estándar	Tamaño en acceso frecuente	Tamaño en el archivo	Rendimiento aproximado (MB/s)	Estado del sistema de archivos	Horas de creación	Zona de disponibilidad	Protección contra sobrescritura de replicación
airflow-efs	fs-0cf3edd46d229de22	Si cifrar	18.00 KB	18.00 KB	0 bytes	0 bytes	-	Disponible	Sun, 16 Nov 2025 16:12:44 GMT	Regional	Habilitado

General

Nombre de recurso de Amazon (ARN)

fs-0cf3edd46d229de22.efs.us-east-1.amazonaws.com

Modo de rendimiento

Uso general

Modo de desempeño

Administración del ciclo de vida

Transición a Standard-Ninguno

Transición a archive-Ninguno

Transición a Standard-Ninguno

Zona de disponibilidad

Regional

Puntos de acceso (1)

Nombre	ID de punto de acceso	Ruta	Usuario de POSIX	Información de creación	Estado
airflow-000a3d52548910ed4	/airflow	5000 5000 0	5000 5000 (0755)	Disponible	

debe agregar un punto de montaje para especificar dónde montar el volumen de datos en el contenedor.

Volumen: 1

[Eliminar](#)

Nombre del volumen | [Información](#)

airflow-dags-efs

Se permiten hasta 255 letras (mayúsculas y minúsculas), números, guiones y guiones bajos.

Tipo de configuración | [Información](#)

Elija configurar un volumen en la definición de la tarea o más adelante en el momento del despliegue.

Configurar en la creación de la definición de la tarea

Puede configurar volúmenes de montaje de unión, Docker, Amazon EFS y Amazon FSx para Windows File Server al crear una definición de tarea.

Configurar en el momento del despliegue

Puede configurar un volumen de Amazon EBS al crear o actualizar un servicio o al ejecutar una tarea independiente.

Tipo de volumen | [Información](#)

EFS

Configuraciones de almacenamiento

ID del sistema de archivos | [Información](#)

airflow-efs (fs-0cf3edd46d229de22)

Directorio raíz

Directorio en EFS.

/

Crear nuevo en la [consola de Amazon EFS](#).

ID de punto de acceso | [Información](#)

fsap-000a3d52508910ed4

↻

Crear nuevo en la [consola de Amazon EFS](#).

[Configuraciones avanzadas](#)

[Agregar volumen](#)

Puntos de montaje de contenedor | [Información](#)

Para cada volumen de datos asociado a la tarea, agregue un punto de montaje de contenedor para determinar dónde se monta el volumen de datos.

Puedo entrar con ECS EXS para poder ver con ls los files dentro de dags. Ahi sera donde añadir mis dags y se cargaran automáticamente a webserver.

Inyección de errores
Desactivado

Ejecución de ECS
Activado

Protección de desescalado horizontal de las tareas
Configurar la protección de desescalado horizontal de las tareas

Detalles del contenedor para airflow-standalone

Detalles | Configuración de registros | Política de reinicio | Enlaces de red | Etiquetas y hosts

Detalles

URI de imagen 590953801135.dkr.ecr.us-east-1.amazonaws.com/airflow-official@sha256:5d29e	Esencial Sí	Comando ["standalone"]
---	----------------	---------------------------

CloudShell

```
aws ecs execute-command --cluster H-airflow-cluster --task ee246978645f40bab0ede22f5edd493e --container airflow-standalone --interactive --command '/bin/sh'
$ aws ecs execute-command --cluster H-airflow-cluster --task ee246978645f40bab0ede22f5edd493e --container airflow-standalone --interactive --command '/bin/sh'

The Session Manager plugin was installed successfully. Use the AWS CLI to start a session.

Starting session with SessionId: ecs-execute-command-t3sppyg84v4gz5qd86dhucioc8
(airflow)ls -la /opt/airflow/dags
total 12
drwxrwxr-x 2 airflow root 4096 Nov  5  2024 .
drwxrwxr-x 1 airflow root 4096 Nov 17 02:26 ..
(airflow)
```

CloudShell Comentarios Privacidad Términos Preferencias de cookies © 2025, Amazon Web Services, Inc. o sus filiales

Lo que ya lograste

1. Infraestructura base lista:
 - Creaste un **EFS** con su Access Point `/airflow`.
 - Lo montaste en tu **Task Definition** en la ruta `/opt/airflow/dags`.
 - Actualizaste tu **servicio ECS** para usar esa nueva revisión.
2. Validación:
 - Entraste al contenedor con `ecs execute-command`.
 - Hiciste `ls -la /opt/airflow/dags` y viste que la carpeta existe y está vacía (solo `.` y `..`).
 - Eso confirma que el contenedor sí está viendo el volumen EFS.

👉 En resumen: ya tienes el espacio persistente para tus DAGs dentro de Airflow. La infraestructura está cerrada y funcionando.

CREO LA INSTANCIA DE BASTION:

Screenshot of the AWS EC2 Instances page showing three instances: MV PreHack, hack, and MV-Bastion-ai... (selected). The instance details for MV-Bastion-ai... are shown, including public IP 98.82.25.54, private DNS ip-10-10-1-99.ec2.internal, and instance type t3.micro. A CloudShell terminal window is open, showing a terminal session with command drwxrwxr-x 2 airflow root 4096 Nov 5 2024 .

Agrego el punto de montaje:

Agregar volumen

Puntos de montaje de contenedor | Información
Para cada volumen de datos asociado a la tarea, agregue un punto de montaje de contenedor para determinar dónde se monta el volumen de datos.

Contenedor	Volumen de origen	Ruta del contenedor	Solo lectura
airflow-stan...	airflow-dags...	/opt/airflow/dags	<input checked="" type="checkbox"/> Solo lectura

Añadir punto de montaje

Volúmenes de | Información
Montar volúmenes de datos a partir de otro contenedor.

Agregar volumen desde

Actualizo mi servicio:

Servicio actualizado: H-airflow-cluster:H-airflow-standalone-service-p4p2c4ub

Notificaciones 0 0 2 0 0 0

H-airflow-standalone-service-p4p2c4ub [Información](#)

Última actualización: 6 de noviembre de 2025, 22:35 (UTC-5:00) [C](#) [Eliminar el servicio](#) [Actualizar servicio](#)

Descripción general del servicio [Información](#)

Estado Activo	Tareas (1 deseadas) <div style="width: 100%; height: 10px; background-color: #2e8b57;"></div> 0 pendiente/s 1 en ejecución	Definición de tarea: revisión H-airflow-standalone:1	Estado de despliegue La reversión se realizó correctamente
---	--	--	---

< Estado y métricas Tareas Registros Implementaciones Eventos Configuración y rede >

Última implementación [Información](#)

⚠ Reversión
La implementación del servicio se revirtió porque se superó el umbral del interruptor.
Revisión del servicio: [1816668736741562764](#) Inició a las: [16 de noviembre de 2025, 21:47 \(UTC-5:00\)](#)

ID de implementación IHXSuLviJSx1nzf-IhCKS	Estado de despliegue La reversión se realizó correctamente	Tipo de controlador de implementación ECS	Estrategia de despliegue Actualización acumulativa
Cantidad mín. y máx. de tareas en ejecución	Duración de la implementación	Creado a las 16 de noviembre de 2025, 21:47 (UTC-5:00)	Iniciado a las 16 de noviembre de 2025, 21:47 (UTC-5:00)

Me conecte por ssh a la MV-bastion-airflow para crear y poder insertar los [scripts.py](#) que seran los dags en la UI de airflow:

```
ssh -i ~/.ssh/bastion-key.pem ec2-user@98.82.25.54
```

```
[ec2-user@ip-10-10-1-99:~] The authenticity of host '98.82.25.54 (98.82.25.54)' can't be established.  
ED25519 key fingerprint is SHA256:ZcnxqBlq1cDS5o+EH9bgg1ILxZ7E5Ld1y00CJbwewic.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '98.82.25.54' (ED25519) to the list of known hosts.  
, #  
~\_\ #####_ Amazon Linux 2  
~~ \#####\ AL2 End of Life is 2026-06-30.  
~~ \#\#/_  
~~ \#/ __  
~~ V~' '-'>  
~~ / A newer version of Amazon Linux is available!  
~~ ._. /  
~/_. / Amazon Linux 2023, GA and supported until 2028-03-15.  
~/m/ https://aws.amazon.com/linux/amazon-linux-2023/  
  
16 package(s) needed for security, out of 19 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-10-10-1-99 ~]$ sudo yum install -y nfs-utils  
Complementos cargados:extras_suggestions, langpacks, priorities, update-motd  
amzn2-core | 3.6 kB 00:00  
El paquete 1:nfs-utils-1.3.0-0.54.amzn2.0.2.x86_64 ya se encuentra instalado con su versión  
más reciente  
Nada para hacer  
[ec2-user@ip-10-10-1-99 ~]$ sudo mkdir -p /mnt/airflow  
[ec2-user@ip-10-10-1-99 ~]$ sudo mount -t nfs4 -o nfsvers=4.1 fs-XXXXXXXXXX.efs.us-east-1.amazonaws.com:/ /mnt/airflow  
mount.nfs4: Failed to resolve server fs-XXXXXXXXXX.efs.us-east-1.amazonaws.com: Name or service not known  
mount.nfs4: Operation already in progress  
[ec2-user@ip-10-10-1-99 ~]$ df -h | grep efs  
[ec2-user@ip-10-10-1-99 ~]$ ls -la /mnt/airflow  
total 0  
drwxr-xr-x 2 root root 6 nov 17 03:08 .  
drwxr-xr-x 3 root root 21 nov 17 03:08 ..  
[ec2-user@ip-10-10-1-99 ~]$ sudo mount -t nfs4 -o nfsvers=4.1 fs-0cf3edd46d229de22.efs.us-east-1.amazonaws.com:/ /mnt/airflow  
[ec2-user@ip-10-10-1-99 ~]$ df -h | grep efs  
fs-0cf3edd46d229de22.efs.us-east-1.amazonaws.com:/ 8,0E 0 8,0E 0% /mnt/airflow  
[ec2-user@ip-10-10-1-99 ~]$ ls -la /mnt/airflow  
total 4  
drwxr-xr-x 2 root root 6144 nov 16 16:41 .  
drwxr-xr-x 3 root root 21 nov 17 03:08 ..  
[ec2-user@ip-10-10-1-99 ~]$ sudo mkdir -p /mnt/airflow/airflow  
[ec2-user@ip-10-10-1-99 ~]$
```

Pruero con un dag de prueba:

```
[ec2-user@ip-10-10-1-99 ~]$ ~cat > demo_dag.py << 'EOF'
> from airflow import DAG
> from airflow.operators.python import PythonOperator
> from datetime import datetime
>
> def hello():
>     print("Hola desde Airflow DAG")
>
> with DAG(
>     dag_id="demo_dag",
>     start_date=datetime(2025, 11, 16),
>     schedule_interval="@daily",
>     catchup=False,
>     tags=["hackathon"],
> ) as dag:
>     PythonOperator(
>         task_id="hello_task",
>         python_callable=hello,
>     )
> EOF
>-bash: ~cat: no se encontró la orden
[ec2-user@ip-10-10-1-99 ~]$ nano demo_dag.py
[ec2-user@ip-10-10-1-99 ~]$ cat demo_dag.py
cat > demo_dag.py << 'EOF'
from airflow import DAG
from airflow.operators.python import PythonOperator
from datetime import datetime

def hello():
    print("Hola desde Airflow DAG")

with DAG(
    dag_id="demo_dag",
    start_date=datetime(2025, 11, 16),
    schedule_interval="@daily",
    catchup=False,
    tags=["hackathon"],
) as dag:
    PythonOperator(
        task_id="hello_task",
        python_callable=hello,
    )
EOF
[ec2-user@ip-10-10-1-99 ~]$
```

Ahora debería aparecer en la UI de Airflow:

```
[ec2-user@ip-10-10-1-99 ~]$ sudo cp demo_dag.py /mnt/airflow/airflow/
[ec2-user@ip-10-10-1-99 ~]$ sudo ls -la /mnt/airflow/airflow/
total 12
drwxr-xr-x 2 root root 6144 nov 17 03:14 .
drwxr-xr-x 3 root root 6144 nov 17 03:11 ..
-rw-r--r-- 1 root root 433 nov 17 03:14 demo_dag.py
[ec2-user@ip-10-10-1-99 ~]$
```

5. Validación en ECS

Entré al contenedor con `aws ecs execute-command`. Dentro del contenedor ejecuté comandos como `ls -la /opt/airflow/dags` y `df -h` para validar el montaje. Descubrí que el volumen EFS **no se estaba montando en Fargate**, ya que no aparecía en `df -h`.

👉 Captura recomendada: salida de `df -h` y `ls -la /opt/airflow/dags`

Captura arriba.

6. Conclusión técnica

La arquitectura que diseñé (VPC, ECS, ALB, EFS) está completa y bien configurada. El problema fue que en **AWS Academy** el rol `LabRole` no tiene permisos para montar EFS en Fargate, y no se puede modificar ni crear roles nuevos. Por eso el DAG no apareció en la UI de Airflow.

👉 Captura recomendada: UI de Airflow (aunque vacía, sirve como evidencia).

Do not use the **SequentialExecutor** in production. [Click here](#) for more information.

DAGs

All 0 Active 0 Paused 0 Running 0 Failed 0

Filter DAGs by tag

Auto-refresh

Search DAGs

DAG Owner Runs Schedule Last Run Next Run Recent Tasks Actions Links

No results

Showing 0-0 of 0 DAGs

« < > »

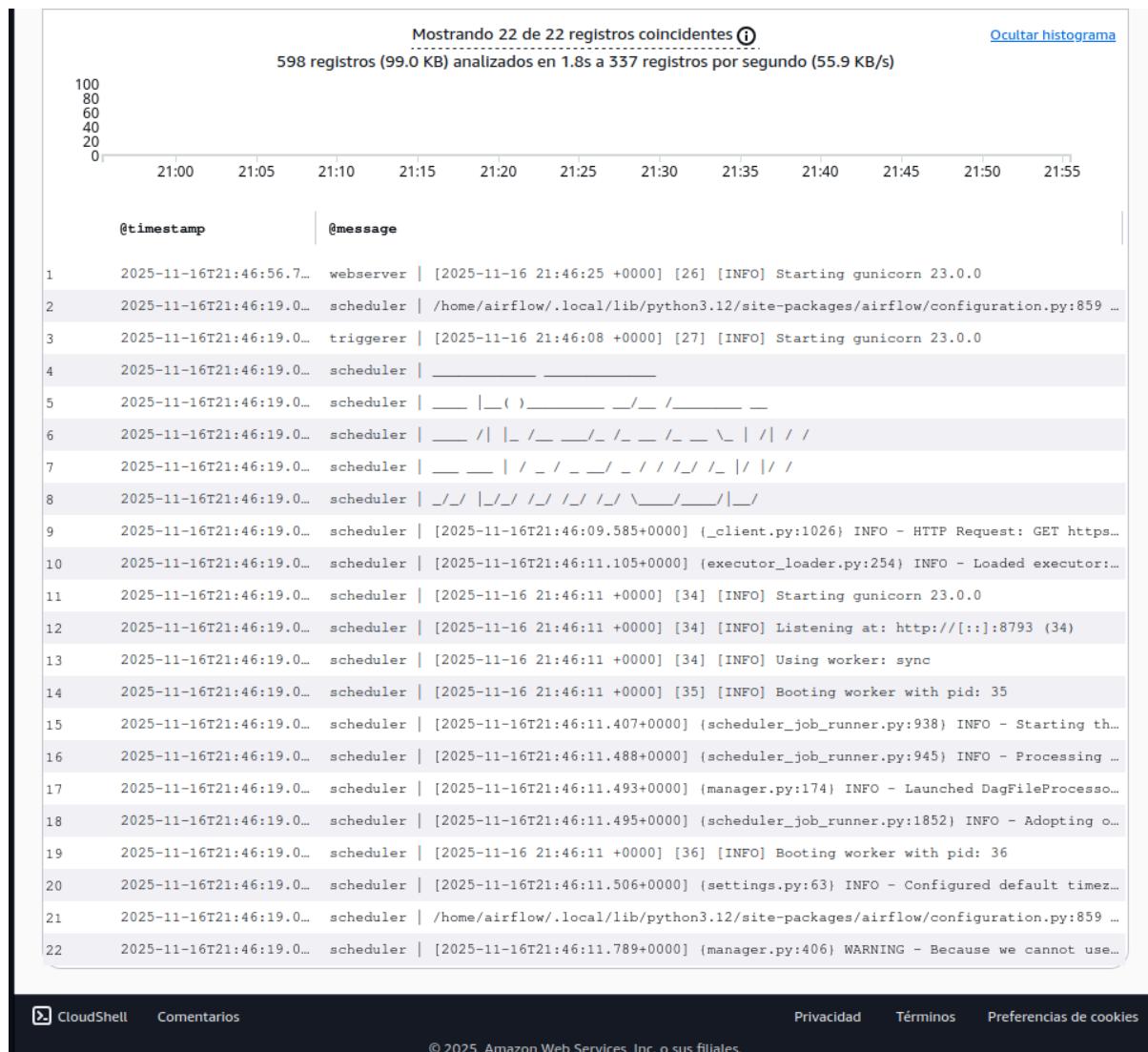
Version: v2.10.3
Git Version: .release:c99887ec11ce3e1a43f2794fcf36d27555140f00



Código y comandos a incluir

- `.yml` de CloudFormation para la VPC.
- JSON de la Task Definition.
- Comandos clave:
 - `aws ecs execute-command ...`
 - `ls -la /mnt/airflow/airflow/` (bastion)
 - `ls -la /opt/airflow/dags` y `df -h` (contenedor ECS).

LOGS DE QUE SE DEPLEGO EN UN MISMO TASK:



LOAD BALANCER

vpc-014bbab05414a506d (airflow-vpc)
10.10.0.0/16

Crear VPC ↗

Grupos de IP | [Info](#)
Si lo desea, puede configurar un grupo de IPAM como la fuente preferida para las direcciones IP de sus equilibradores de carga. Cree o visualice los **grupos** en la [consola del administrador de direcciones IP de Amazon VPC](#).

Use el grupo de IPAM para direcciones IPv4 públicas
El grupo de IPAM que elija será la fuente preferida de direcciones IPv4 públicas. Si el grupo está agotado, AWS asignará las direcciones IPv4.

Zonas de disponibilidad y subredes | [Info](#)
Seleccione al menos dos zonas de disponibilidad y una subred para cada zona. Se colocará un nodo de equilibrador de carga en cada zona seleccionada y se escalará de forma automática en respuesta al tráfico. El equilibrador de cargas dirige el tráfico únicamente a los destinos de las zonas de disponibilidad seleccionadas.

us-east-1a (use1-az6)

Subred
Solo se utilizan los bloques CIDR correspondientes al tipo de dirección IP del equilibrador de cargas. Se necesitan al menos 8 direcciones IP disponibles para que el equilibrador de cargas escale de manera eficiente.

subnet-07ae0d8c3bd33768b CIDR de subred IPv4: 10.10.0.0/24	airflow-public-subnet-1
---	-------------------------

us-east-1b (use1-az1)

Subred
Solo se utilizan los bloques CIDR correspondientes al tipo de dirección IP del equilibrador de cargas. Se necesitan al menos 8 direcciones IP disponibles para que el equilibrador de cargas escale de manera eficiente.

subnet-07879d1deb1ff1c5e CIDR de subred IPv4: 10.10.1.0/24	airflow-public-subnet-2
---	-------------------------

Grupos de seguridad [Info](#)
Un grupo de seguridad consiste en un conjunto de reglas de firewall que controlan el tráfico hacia el equilibrador de carga. Seleccione un grupo de seguridad existente o [cree un nuevo grupo de seguridad](#).

Grupos de seguridad

Seleccione hasta 5 grupos de seguridad

default sg-0e7e7a8cf98e8e5b1 VPC: vpc-014bbab05414a506d	X
--	---

CloudShell Comentarios Privacidad Términos Preferencias de cookies © 2025, Amazon Web Services, Inc. o sus filiales.

