

# Boto3

---

Taller Semestral - 2021-2  
Introducción a Cloud Computing con AWS

Departamento de Informática  
Universidad Técnica Federico Santa María

24 de noviembre de 2021 - v1.0

```
self.fingerprints = set()
self.logdupes = True
self.debug = debug
self.logger = logging.getLogger(__name__)
if path:
    self.file = open(os.path.join(
        self.file.seek(0)
        self.fingerprints.update(e

@classmethod
def from_settings(cls, settings):
    debug = settings.getbool('debug')
    return cls(job_dir(settings),

def request_seen(self, request):
    fp = self.request_fingerprint()
    if fp in self.fingerprints:
        return True
    self.fingerpr
    if self.file:
        self.file

EX UMBRA IN SOLEM
```

# Índice

- 1 Descripción
- 2 Configuración
- 3 Uso

```
s = set()
True
log
logging.getLogger('
+ open(os.path.join
seek(0)
rprints.update(e.r
(cls, settings):
ings.getbool('superr
b_dir(settings), c
(self, request):
request_fingerprint(
lf.fingerprints:
True
prints.add(fp)
le:
ile.write(fp + os.l
ngerprint(self, re
```

# Descripción

- SDK de Amazon Web Services para Python

```
s = set()
True
log
logging.getLogger('
+ open(os.path.join
seek(0)
rprints.update(e, r

(cls, settings):
ings.getbool('sure
ob_dir(settings),

(self, request):
request_fingerprint(
lf.fingerprints:
True
prints.add(fp)
le:
ile.write(fp + os.l
ngerprint(self, re
```

# Credenciales

- Mismo procedimiento que para AWS-CLI

```
s = set()
True
log
logging.getLogger('
+ open(os.path.join
seek(0)
rprints.update(e.r
(cls, settings):
ings.getbool('sure
ob_dir(settings),
(self, request):
request_fingerprint(
lf.fingerprints:
True
prints.add(fp)
le:
ile.write(fp + os.l
ngerprint(self, re
```

# Credenciales

PermissionsGroupsTagsSecurity credentialsAccess Advisor

Sign-in credentials

Summary

Console passwordEnabled (last signed in today) | Manage

Assigned MFA deviceNot assigned | Manage

Signing certificatesNone

Access keys

Use access keys to make programmatic calls to AWS from the AWS CLI, 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.

For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.  
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive. [Learn more](#)

Create access key

Access key ID	Created	Last used	Status
No results			

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate access to AWS CodeCommit repositories. [Learn more](#)

Upload SSH public key

SSH key ID	Uploaded	Status
No results		

HTTPS Git credentials for AWS CodeCommit

Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can generate and store up to 2 sets of credentials. [Learn more](#)

Generate credentials

No credentials have been generated.

Credentials for Amazon Keyspaces (for Apache Cassandra)

Generate a user name and password you can use to authenticate to Amazon Keyspaces. You can generate and store up to 2 sets of credentials for Amazon Keyspaces. [Learn more](#)

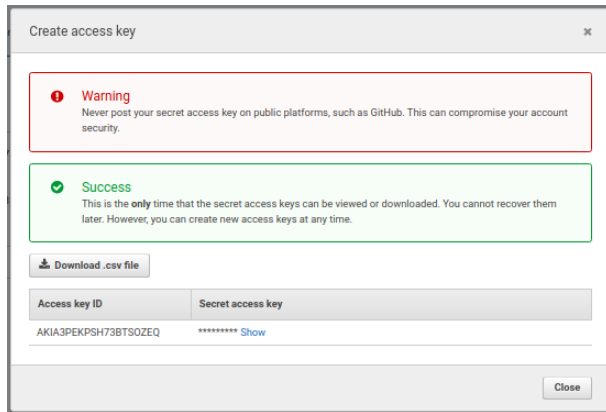
Generate credentials

No credentials have been generated.

Consola IAM para el usuario seleccionado



# Credenciales



Consola IAM para el usuario seleccionado

# Credenciales

PermissionsGroupsTagsSecurity credentialsAccess Advisor

Sign-in credentials

Summary

• Console sign-in link <https://788416926207.signin.aws.amazon.com/console>

Console passwordEnabled (last signed in Today) | [Manage](#)

Assigned MFA deviceNot assigned | [Manage](#)

Signing certificatesNone

Access keys

Use access keys to make programmatic calls to AWS from the AWS CLI, 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.

For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.  
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive. [Learn more](#)

Create access key

Access key ID	Created	Last used	Status
AKIA3PEKPSH73BTSQZEQ	2021-11-17 17:23 ADT	N/A	Active   <a href="#">Make inactive</a>

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate access to AWS CodeCommit repositories. [Learn more](#)

Upload SSH public key

SSH key ID	Uploaded	Status
No results		

HTTPS Git credentials for AWS CodeCommit

Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can generate and store up to 2 sets of credentials. [Learn more](#)

Generate credentials

No credentials have been generated.

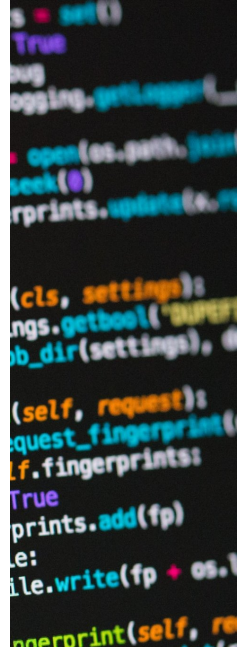
Credentials for Amazon Keyspaces (for Apache Cassandra)

Generate a user name and password you can use to authenticate to Amazon Keyspaces. You can generate and store up to 2 sets of credentials for Amazon Keyspaces. [Learn more](#)

Generate credentials

No credentials have been generated.

Consola IAM para el usuario seleccionado



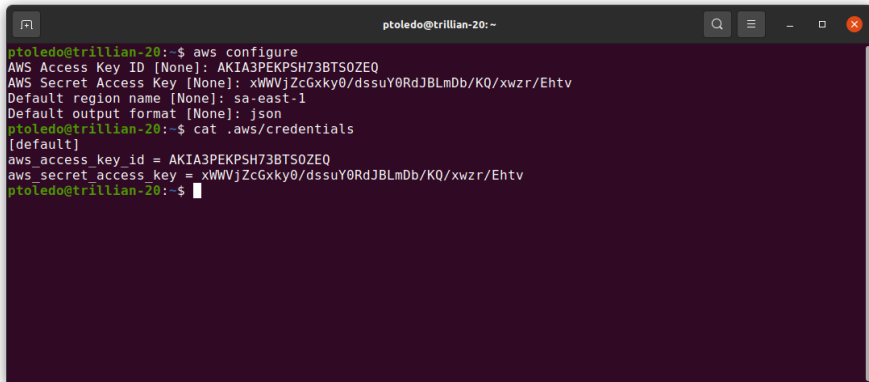
# Instalación

```
pip install boto3
```

```
s = set()
True
log
logging.getLogger('
+ open(os.path.join
seek(0)
rprints.update(e.r
(cls, settings):
ings.getbool('sure
ob_dir(settings),
(self, request):
request_fingerprint(
lf.fingerprints:
True
prints.add(fp)
le:
ile.write(fp + os.l
ngerprint(self, re
```



# Configuración de credenciales

A terminal window with a dark background and light green text. The window title is 'ptoledo@trillian-20: ~'. The user has run 'aws configure' and entered their AWS Access Key ID, Secret Access Key, and region. Then they ran 'cat .aws/credentials' to show the resulting configuration file.

```
ptoledo@trillian-20:~$ aws configure
AWS Access Key ID [None]: AKIA3PEKPSH73BTS0ZEQ
AWS Secret Access Key [None]: xWwVjZcGxky0/dssuY0RdJBLmDb/KQ/xwzr/Ehtv
Default region name [None]: sa-east-1
Default output format [None]: json
ptoledo@trillian-20:~$ cat .aws/credentials
[default]
aws_access_key_id = AKIA3PEKPSH73BTS0ZEQ
aws_secret_access_key = xWwVjZcGxky0/dssuY0RdJBLmDb/KQ/xwzr/Ehtv
ptoledo@trillian-20:~$
```

Consola

# Instalación

```
import boto3
# boto3.resource('dynamodb')
```

```
s = self()
True
log
logging.getLogger('
+ open(os.path.join
seek(0)
rprints.update(e, r
(cls, settings):
ings.getbool('surfer
ob_dir(settings), e
(self, request):
request_fingerprint(
lf.fingerprints:
True
prints.add(fp)
le:
ile.write(fp + os.l
ngerprint(self, re
```

## ¿Preguntas?

Fotografía:  
<https://unsplash.com/@cdr6934>

Tema de la presentación basado en:  
<https://gitlab.com/matteodelabre/beamertHEME-bruno>