

CD CON AWS

ELASTIC BEANSTALK- ENVIROMENTS

aws

Services

Search

[Alt+S]

N. Virginia

edango09

Step 1

Configure environment

Step 2

[Configure service access](#)

Step 3 - optional

[Set up networking, database, and tags](#)

Step 4 - optional

Configure instance traffic and scaling

Step 5 - optional

Configure updates, monitoring, and logging

Step 6

Review

Configure environment

Info

Environment tier

Info

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

☒ Web server environment

Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

☐ Worker environment

Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information

Info

Application name

calculadoraPython

Maximum length of 100 characters.

Application tags (optional)

Environment information

Info

Choose the name, subdomain and description for your environment. These cannot be changed later.

CloudShell

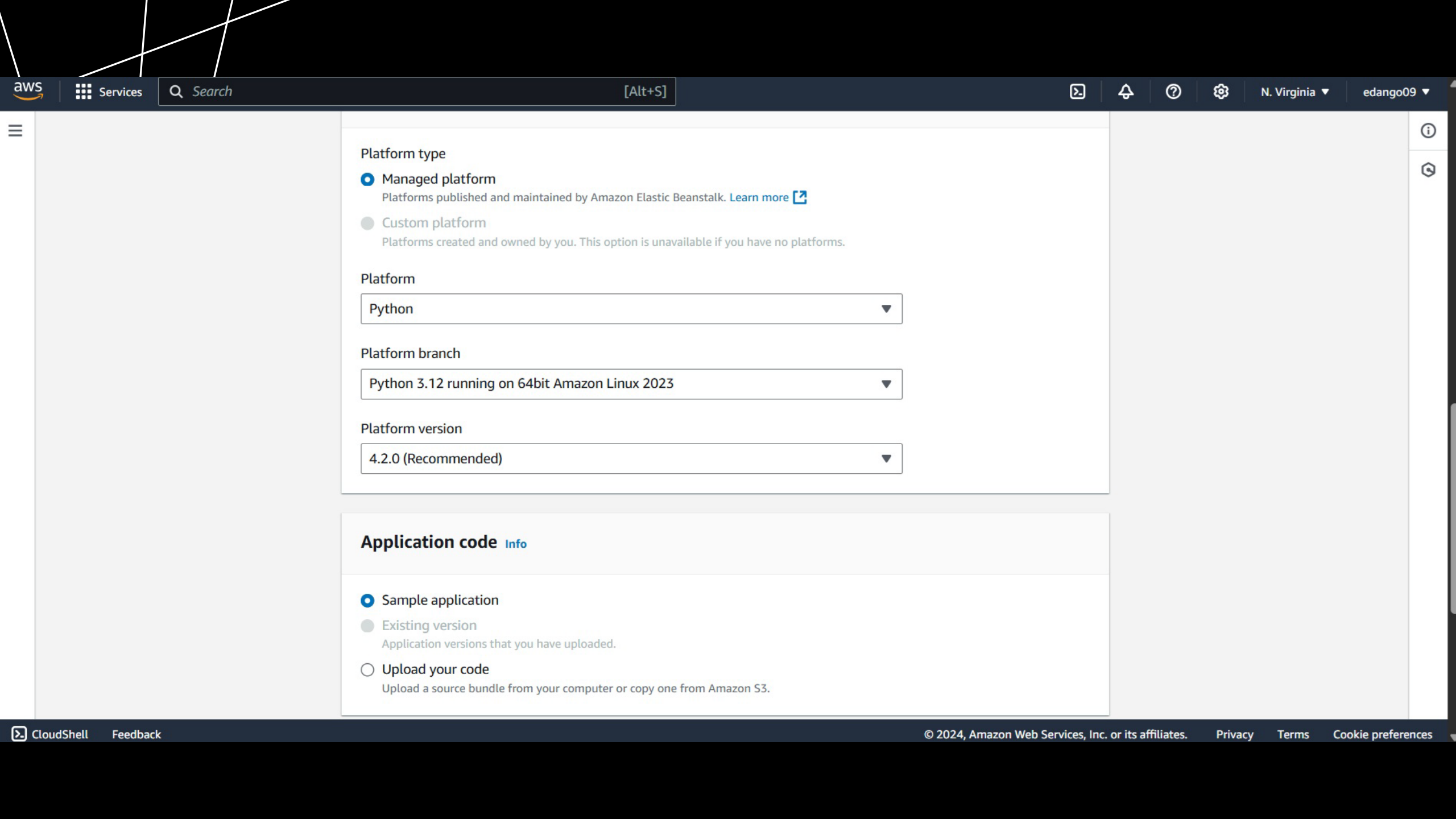
Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences



Services

Search

[Alt+S]



N. Virginia

edango09



Platform type

☒ Managed platform

Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

☐ Custom platform

Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Python

Platform branch

Python 3.12 running on 64bit Amazon Linux 2023

Platform version

4.2.0 (Recommended)

Application code [Info](#)

☒ Sample application

☐ Existing version

Application versions that you have uploaded.

☐ Upload your code

Upload a source bundle from your computer or copy one from Amazon S3.

CODEBUILD - PROJECT

aws

Services

Search

[Alt+S]

Ohio

edango09

Provisioning model [Info](#)

☒ On-demand
Automatically provision build infrastructure in response to new builds.

☐ Reserved capacity
Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.

Environment image

☒ Managed image
Use an image managed by AWS CodeBuild

☐ Custom image
Specify a Docker image

Compute

☒ EC2
Optimized for flexibility during action runs

☐ Lambda
Optimized for speed and minimizes the start up time of workflow actions

Operating system

Amazon Linux

Runtime(s)

Standard

Image

aws/codebuild/amazonlinux2-x86_64-standard:5.0

Image version

Always use the latest image for this runtime version

☐ Use GPU-enhanced compute

Service role

☒ New service role
Create a service role in your account

☐ Existing service role
Choose an existing service role from your account


CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Buildspec

Build specifications

 If the primary source for the build project is "No source", a valid buildspec command must be provided

☒ **Insert build commands**
Store build commands as build project configuration

☐ **Use a buildspec file**
Store build commands in a YAML-formatted buildspec file

Build commands

Enter commands you want to run during the build phase. Separate each build command with "&&." For example, "mvn test && mvn package." Use a buildspec file to run commands in other phases or if you have a long list of commands.

e.g. echo "Hello World"

[Switch to editor](#)

AWS CODEPIPELINE - PIPELINE

Services

Search

[Alt+S]

Ohio

edango09

Developer Tools

CodePipeline

▶ Source • CodeCommit

▶ Artifacts • CodeArtifact

▶ Build • CodeBuild

▶ Deploy • CodeDeploy

▼ Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

▶ Settings

Go to resource

Feedback

Developer Tools > CodePipeline > Pipelines > pycalcx

pycalcx

Notify

Edit

Stop execution

Clone pipeline

Release change

Pipeline type: V2

Execution mode: QUEUED

✔ Source

Succeeded

Pipeline execution ID: [0c3a5924-34c7-41d4-8a8a-d88300fa0182](#)

Source

[GitHub \(Version 1\)](#)

✔ Succeeded - 20 minutes ago

[c85be83b](#)

View details

[c85be83b](#) Source: Update README.md

Disable transition

✔ Build

Succeeded

Pipeline execution ID: [0c3a5924-34c7-41d4-8a8a-d88300fa0182](#)

Build

[AWS CodeBuild](#)

✔ Succeeded - 16 minutes ago

View details

✔

✔

✔

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

AWS CODEPIPELINE - PIPELINE

pycalcx

Pipeline type: **V2** Execution mode: **QUEUED**

 Source Succeeded

Pipeline execution ID: [0c3a5924-34c7-41d4-8a8a-d88300fa0182](#)

Source

[GitHub \(Version 1\)](#) 

✔ Succeeded - 20 minutes ago

[c85be83b](#)

[View details](#)

[c85be83b](#) Source: Update README.md

Disable transition

 Build Succeeded

Pipeline execution ID: [0c3a5924-34c7-41d4-8a8a-d88300fa0182](#)

Build

AWS CodeBuild

✔ Succeeded - 16 minutes ago

[View details](#) **Notify** ▼

Edit

Stop execution

Clone pipeline

Release change

Start rollback

CÓDIGO

```
1 from flask import Flask, request
2 from calc import Calculadora
3
4 app = Flask(__name__)
5 calculadora = Calculadora()
6
7 @app.route('/')  # edang
8 def index():
9     return "Bienvenido! Esta aplicación ayuda a elevar números cualquiera a sus potencias."
10
11 @app.route('/hola')  # edang
12 def hola():
13     nombre = request.args.get('nombre', 'Mundo')
14     return f"Hola {nombre}!"
15
16 @app.route('/cuadrado')  # edang
17 def cuadrado():
18     numero = request.args.get('numero', default=0, type=float)
19     resultado = calculadora.cuadrado(numero)
20     return f"El cuadrado de {numero} es {resultado}"
21
22 @app.route('/cubo')  # edang
23 def cubo():
24     numero = request.args.get('numero', default=0, type=float)
25     resultado = calculadora.cubo(numero)
26     return f"El cubo de {numero} es {resultado}"
27
28 if __name__ == '__main__':
29     app.run(port=5000)
30
```

Flask: Microframework para Python que permite desarrollar aplicaciones web rápidamente sin la complejidad de un framework más robusto, como Django.

Proporciona las herramientas básicas necesarias para crear rutas, manejar solicitudes y renderizar plantillas, lo que hace que sea perfecto para proyectos que no requieren una estructura compleja o funcionalidades avanzadas.


```

1  class Calculadora: 4 usages 1 edang
2      def cuadrado(self, numero): 4 usages 1 edang
3          return numero * numero
4
5      def cubo(self, numero): 3 usages 1 edang
6          return numero * numero * numero
7

```

Unittest: Ideal para probar proyectos pequeños como una calculadora en línea, permite crear pruebas unitarias de manera simple y efectiva.

```

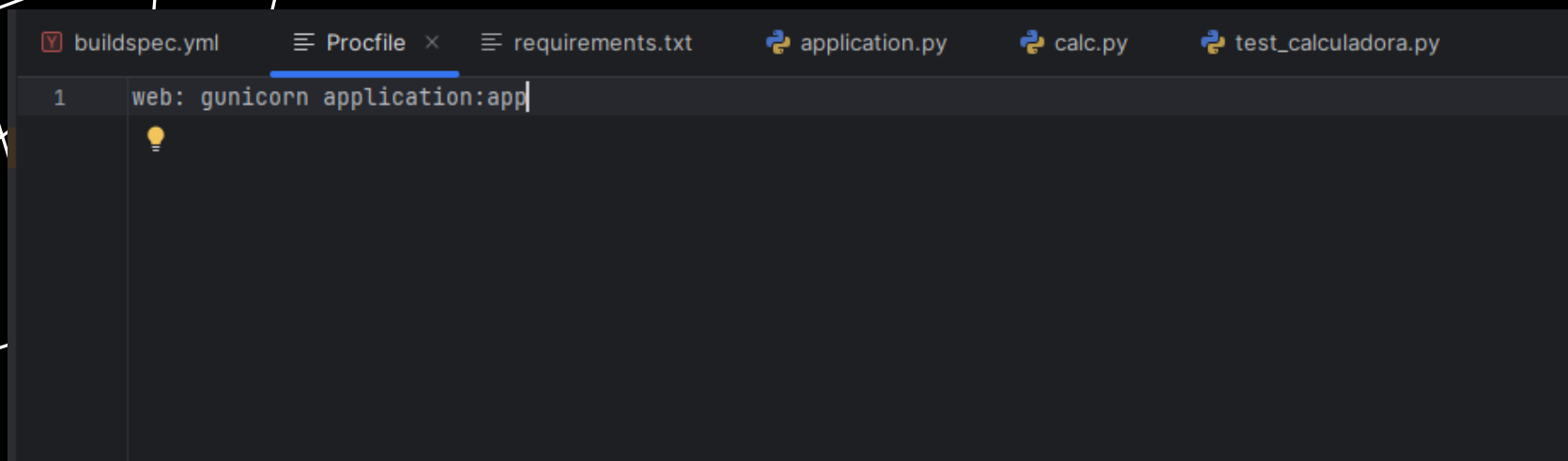
1  import unittest
2  from calc import Calculadora
3
4
5  class TestCalculadora(unittest.TestCase): 1 edang
6
7      def setUp(self): 1 edang
8          self.calc = Calculadora()
9
10     def test_cuadrado_positivo(self): 1 edang
11         self.assertEqual(self.calc.cuadrado(5.0), second: 25.0)
12
13     def test_cuadrado_cero(self): 1 edang
14         self.assertEqual(self.calc.cuadrado(0.0), second: 0.0)
15
16     def test_cuadrado_negativo(self): 1 edang
17         self.assertEqual(self.calc.cuadrado(-4.0), second: 16.0)
18
19     def test_cubo_positivo(self): 1 edang
20         self.assertEqual(self.calc.cubo(5.0), second: 125.0)
21
22     def test_cubo_negativo(self): 1 edang
23         self.assertEqual(self.calc.cubo(-4.0), -64.0)
24
25     if __name__ == '__main__':
26         unittest.main()
27

```

```
1 version: 0.2
2
3 phases:
4   install:
5     runtime-versions:
6       python: 3.8
7     commands:
8       - python -m venv venv
9       - source venv/bin/activate
10      - pip install --upgrade pip
11      - pip install -r requirements.txt
12
13   build:
14     commands:
15       - mkdir -p test-reports
16       - echo "Running tests with coverage..."
17       - source venv/bin/activate
18       - export PYTHONPATH=$PYTHONPATH:$(pwd)
19       - pytest --junitxml=test-reports/junit-report.xml --cov=.
20       - coverage xml -o coverage.xml
21       - coverage report --fail-under=40
22
23   post_build:
24     commands:
25       - echo "Build and tests completed successfully."
26
27 artifacts:
28   files:
29     - '**/*'
30   discard-paths: no
31
32 reports:
33   junit_reports:
34     files:
35     • - test-reports/junit-report.xml
36
```

La opción `--junitxml` es una forma de exportar los resultados de las pruebas en un formato compatible con JUnit. Muchas herramientas de integración continua (CI) aceptan resultados en formato JUnit para mostrar informes detallados de pruebas.

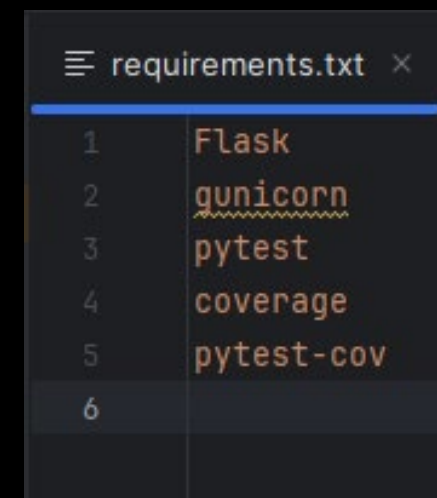
Aunque las pruebas están en `unittest`, se genera el reporte en formato JUnit para integrarlo fácilmente en el pipeline de CI/CD.



The screenshot shows a code editor with several tabs: buildspec.yml, Procfile, requirements.txt, application.py, calc.py, and test_calculadora.py. The Procfile tab is active and shows the following content:

```
1 web: gunicorn application:app
```

A lightbulb icon is visible below the code, indicating a suggestion or tip.



The screenshot shows a code editor with the requirements.txt tab active. The file contains the following dependencies:

```
1 Flask
2 gunicorn
3 pytest
4 coverage
5 pytest-cov
6
```

El Procfile y gunicorn permiten ejecutar la aplicación en producción de forma eficiente.

El Procfile indica a AWS cómo iniciar la aplicación, especificando que el proceso web debe usar gunicorn para cargar y ejecutar la instancia de la aplicación (application:app).

Gunicorn, a su vez, es un servidor WSGI diseñado para manejar múltiples solicitudes simultáneas, optimizando el rendimiento y la estabilidad en producción, lo cual es crucial en un entorno en la nube donde la escalabilidad y la capacidad de respuesta son esenciales.



GRACIAS