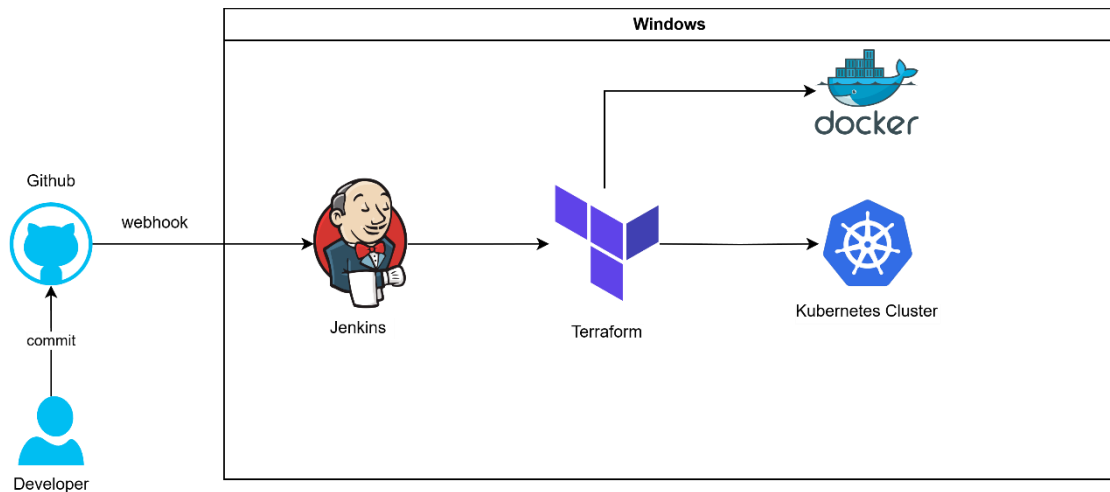


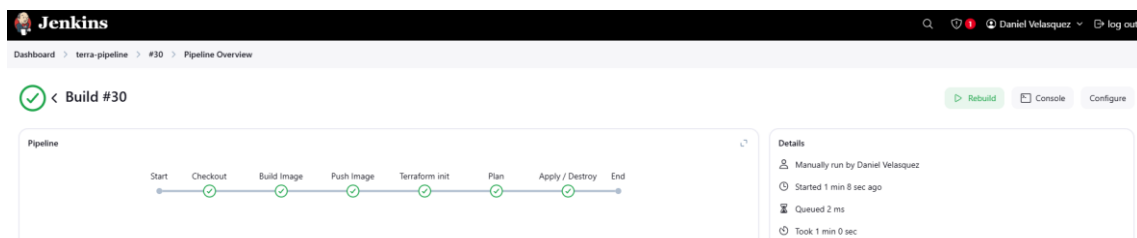
## Flujo y diseño CI/CD despliegue de aplicativo Python a Kubernetes



### Explicación:

1. developer realiza un push a un repositorio github.
2. Github dispara a través de un webhook, el pipeline a través de Jenkins.
3. Jenkins a través de “stages” arma la imagen en Docker, realiza push a dockerhub y ejecuta terraform para el despliegue del aplicativo al cluster de Kubernetes.
4. Terraform se encarga de la creación de servicios, namespaces, mapear el provider (Kubernetes cluster de Docker Desktop). Se realiza terraform init, plan y apply.

### Pruebas de ejecución paso a paso de pipeline:



Jenkins

Dashboard

terra-pipeline

#30

Pipeline Console

Search

🔔

👤 Daniel Velasquez

🚪 log out

✔️ < Build #30

🔄 Rebuild

👁 Overview

⚙ Configure

⋮

Success 3 min 43 sec ago in 1 min 0 sec

🕒 Checkout

🕒 Build Image

🕒 Push Image

🕒 Terraform init

🕒 Plan

🕒 Apply / Destroy

Stage 'Checkout'

🕒 Started 3 min 42 sec ago

🕒 Queued 0 ms

🕒 Took 2 sec

🕒 Success

🖥 Running on Jenkins

📄 View as plain text

🕒 git

2 sec

🔍

🔄

⌵

```
0 The recommended git tool is: NONE
1 No credentials specified
2 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\workspace\terra-pipeline\.git # timeout=10
3 Fetching changes from the remote git repository
4 > git.exe config remote.origin.url https://github.com/danielvh01/devops-python.git # timeout=10
5 Fetching upstream changes from https://github.com/danielvh01/devops-python.git
6 > git.exe --version # timeout=10
7 > git --version # "git version 2.45.2.windows.1"
8 > git.exe fetch --tags --force --progress -- https://github.com/danielvh01/devops-python.git +refs/heads/*:refs/remotes/origin/* # timeout=10
9 > git.exe rev-parse "refs/remotes/origin/develop" (commit) # timeout=10
10 Checking out Revision 5f4c1c324bad7e8d1c88ee4e388e41878f77ddee (refs/remotes/origin/develop)
11 > git.exe config core.sparsecheckout # timeout=10
12 > git.exe checkout -f 5f4c1c324bad7e8d1c88ee4e388e41878f77ddee # timeout=10
13 > git.exe branch -a -v --no-abbrev # timeout=10
14 > git.exe branch -D develop # timeout=10
15 > git.exe checkout -b develop 5f4c1c324bad7e8d1c88ee4e388e41878f77ddee # timeout=10
```

Jenkins

Dashboard

terra-pipeline

#30

Pipeline Console

Search

🔔

👤 Daniel Velasquez

🚪 log out

✔️ < Build #30

🔄 Rebuild

👁 Overview

⚙ Configure

⋮

Success 3 min 43 sec ago in 1 min 0 sec

🕒 Checkout

🕒 Build Image

🕒 Push Image

🕒 Terraform init

🕒 Plan

🕒 Apply / Destroy

Windows Batch Script

13 sec

🔍

🔄

⌵

```
16 #4 DONE 0.0s
17
18 #5 [internal] load build context
19 #5 transferring context: 62.96MB 1.1s done
20 #5 DONE 1.1s
21
22 #6 [2/5] WORKDIR /app
23 #6 CACHED
24
25 #7 [3/5] COPY requirements.txt requirements.txt
26 #7 CACHED
27
28 #8 [4/5] RUN pip3 install -r requirements.txt
29 #8 CACHED
30
31 #9 [5/5] COPY . .
32 #9 DONE 0.2s
33
34 #10 exporting to image
35 #10 exporting layers 0.1s done
36 #10 writing image sha256:12ac26716e032ef7891f15d8e4d918b2348dc4085162ce351e947638b3ad89 done
37 #10 naming to docker.io/danielvh01/python_django_api:latest done
38 #10 DONE 0.1s
39
40 What's Next?
41 View a summary of image vulnerabilities and recommendations - docker scout quickview
```

Jenkins

Dashboard

terra-pipeline

#30

Pipeline Console

Search

🔔

👤 Daniel Velasquez

🚪 log out

✔️ < Build #30

🔄 Rebuild

👁 Overview

⚙ Configure

⋮

Success 3 min 43 sec ago in 1 min 0 sec

🕒 Checkout

🕒 Build Image

🕒 Push Image

🕒 Terraform init

🕒 Plan

🕒 Apply / Destroy

Stage 'Push Image'

🕒 Started 3 min 37 sec ago

🕒 Queued 0 ms

🕒 Took 40 sec

🕒 Success

🖥 Running on Jenkins

📄 View as plain text

✔️ Successfully authenticated. Pushing the image...

Print Message

15 ms

🔍

🔄

⌵

✔️ Checks if running on a Unix-like node

16 ms

🔍

🔄

⌵

✔️ docker tag "%JD\_ID%" "%JD\_TAGGED\_IMAGE\_NAME%"

Windows Batch Script

0.38 sec

🔍

🔄

⌵

✔️ Checks if running on a Unix-like node

11 ms

🔍

🔄

⌵

✔️ docker push "%JD\_TAGGED\_IMAGE\_NAME%"

Windows Batch Script

39 sec

🔍

🔄

⌵

✔️ Image pushed successfully!

Print Message

7 ms

🔍

🔄

⌵

```
0 🚫 Image pushed successfully!
```

Jenkins

🔍

🔔

👤 Daniel Velasquez

🔗 log out

Dashboard > terra-pipeline > #30 > Pipeline Console

✔️ < Build #30

▶ Rebuild

🔍 Overview

⚙️ Configure

⋮

Success 3 min 43 sec ago in 1 min 0 sec

🕒 Checkout

🕒 Build Image

🕒 Push Image

🕒 Terraform init

🕒 Plan

🕒 Apply / Destroy

Stage "Terraform init"

🕒 Started 2 min 56 sec ago

🕒 Queued 0 ms

🕒 Took 4,1 sec

🟢 Success

🖨️ Running on Jenkins

📄 View as plain text

terraform init

Windows PowerShell Script

4,1 sec

🔍

🔗

⬆

```
0 Initializing the backend...
1 Initializing provider plugins...
2   - Reusing previous version of hashicorp/kubernetes from the dependency lock file
3   - Installing hashicorp/kubernetes v2.35.1...
4   - Installed hashicorp/kubernetes v2.35.1 (signed by HashiCorp)
5
6 Terraform has been successfully initialized!
7
8 You may now begin working with Terraform. Try running "terraform plan" to see
9 any changes that are required for your infrastructure. All Terraform commands
10 should now work.
11
12 If you ever set or change modules or backend configuration for Terraform,
13 rerun this command to reinitialize your working directory. If you forget, other
14 commands will detect it and remind you to do so if necessary.
```

Jenkins

🔍

🔔

👤 Daniel Velasquez

🔗 log out

Dashboard > terra-pipeline > #30 > Pipeline Console

✔️ < Build #30

▶ Rebuild

🔍 Overview

⚙️ Configure

⋮

Success 3 min 43 sec ago in 1 min 0 sec

🕒 Checkout

🕒 Build Image

🕒 Push Image

🕒 Terraform init

🕒 Plan

🕒 Apply / Destroy

Windows PowerShell Script

4,1 sec

🔍

🔗

⬆

```
205 + publish_not_ready_addresses = false
206 + selector = {
207   + "app" = "python-api"
208 }
209 + session_affinity = "None"
210 + type = "LoadBalancer"
211
212 + port {
213   + node_port = (known after apply)
214   + port = 80
215   + protocol = "TCP"
216   + target_port = "8080"
217 }
218
219 + session_affinity_config (known after apply)
220 }
221 }
222
223 Plan: 6 to add, 0 to change, 0 to destroy.
224
225 -----
226
227 Saved the plan to: tfplan
228
229 To perform exactly these actions, run the following command to apply:
230 terraform apply "tfplan"
```

**Jenkins** 🔍 🔔 👤 Daniel Velasquez 🚪 log out

Dashboard > terra-pipeline > #30 > Pipeline Console

🟢 < Build #30 🔄 Rebuild 📄 Overview ⚙️ Configure ⋮

Success 3 min 43 sec ago in 1 min 0 sec

- 🟢 Checkout
- 🟢 Build Image
- 🟢 Push Image
- 🟢 Terraform init
- 🟢 Plan
- 🟡 Apply / Destroy

```

213     + node_port = (known after apply)
214     + port      = 80
215     + protocol  = "TCP"
216     + target_port = "8000"
217   }
218   + session_affinity_config (known after apply)
219 }
220 }
221 }
222
223 Plan: 6 to add, 0 to change, 0 to destroy.
224 kubernetes_namespace.python_api: Creating...
225 kubernetes_namespace.python_api: Creation complete after 0s [id-python-api]
226 kubernetes_config_map.app_config: Creating...
227 kubernetes_secret.app_secret: Creating...
228 kubernetes_config_map.app_config: Creation complete after 0s [id-python-api/app-config]
229 kubernetes_secret.app_secret: Creation complete after 0s [id-python-api/app-secret]
230 kubernetes_deployment.app: Creating...
231 kubernetes_deployment.app: Creation complete after 4s [id-python-api/python-api]
232 kubernetes_service.app_service: Creating...
233 kubernetes_horizontal_pod_autoscaler.app_hpa: Creating...
234 kubernetes_horizontal_pod_autoscaler.app_hpa: Creation complete after 0s [id-python-api/python-api-hpa]
235 kubernetes_service.app_service: Creation complete after 1s [id-python-api/python-api-service]
236
237 Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
238

```

## Validación de pods en kubectl

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

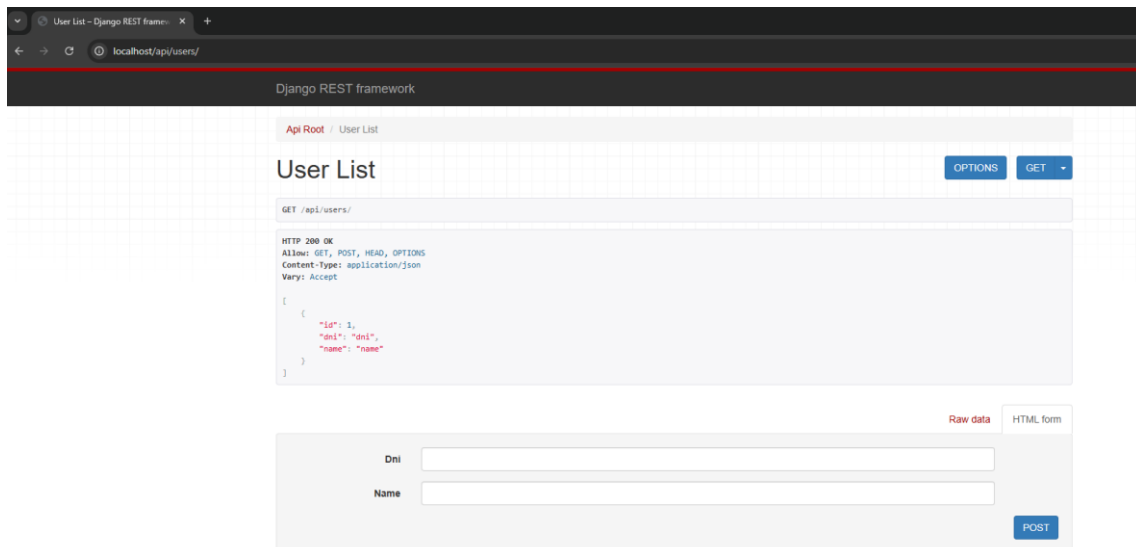
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\PC> kubectl get namespace
NAME                STATUS AGE
default             Active 180d
development         Active 23d
kube-node-lease     Active 180d
kube-public         Active 180d
kube-system         Active 180d
python-api          Active 9m24s
PS C:\Users\PC> kubectl get pods --namespace=python-api
NAME                                READY STATUS RESTARTS AGE
python-api-7597ccb4c4-k77nz        1/1   Running 0      9m39s
python-api-7597ccb4c4-vqrqf        1/1   Running 0      9m39s
PS C:\Users\PC> kubectl get pods --namespace=python-api -o wide
NAME                                READY STATUS RESTARTS AGE IP          NODE             NOMINATED NODE READINESS GATES
python-api-7597ccb4c4-k77nz        1/1   Running 0      10m 10.1.0.44  docker-desktop  <none>          <none>
python-api-7597ccb4c4-vqrqf        1/1   Running 0      10m 10.1.0.45  docker-desktop  <none>          <none>
PS C:\Users\PC>

```

```
Windows PowerShell
python-api-7597ccb4c4-k77nz 1/1 Running 0 10m 10.1.0.44 docker-desktop <none> <none>
python-api-7597ccb4c4-vqrqf 1/1 Running 0 10m 10.1.0.45 docker-desktop <none> <none>
PS C:\Users\PC> kubectl describe pod --namespace=python-api python-api-7597ccb4c4-k77nz
Name: python-api-7597ccb4c4-k77nz
Namespace: python-api
Priority: 0
Service Account: default
Node: docker-desktop/192.168.65.3
Start Time: Tue, 11 Feb 2025 23:05:27 -0600
Labels: app=python-api
pod-template-hash=7597ccb4c4
Annotations: <none>
Status: Running
IP: 10.1.0.44
IPs:
IP: 10.1.0.44
Controlled By: ReplicaSet/python-api-7597ccb4c4
Containers:
  python-api:
    Container ID: docker://281a090824950b804405a33911c096aff4b254103ef2b985f9ab6e94902f9ce
    Image: danielvh01/python_django_api:latest
    Image ID: docker-pullable://danielvh01/python_django_api@sha256:44857266731ff1e5888119797af59ae47010f14f51d03d8953a2fcc48a426494
    Port: 8000/TCP
    Host Port: 0/TCP
    State: Running
```

Prueba de uso de aplicativo Python hosteado en Kubernetes mediante load balancer expuesto en puerto 80.



Repositorio publico de la soluci3n:

<https://github.com/danielvh01/devops-python.git>