



KeepCoding Masters

Ruben Martin

Yilis Ramirez

Francesc Blanco

Jonás Márquez

Proyecto Final de BootCamp DevOps V5



KeepCoding Masters Team



Jonás Márquez



Ruben Martin



Yilis Ramirez



Francesc Blanco



Cliente: KeepCoding Academy

Servicio:

Desarrollamos aplicaciones de gestión de datos para nuestros clientes.

Es de vital importancia garantizar la disponibilidad de la información, así como el cumplimiento de normas ISO.

Además, es fundamental que las aplicaciones estén siempre disponibles.

Por todo esto, el cliente ha decidido replicar toda su infraestructura actual On-Premise en la Nube a manera de DR.



Cliente: KeepCoding Academy

El cliente ha seleccionado AWS como proveedor de servicios en la nube, y la infraestructura que se debe replicar se basa en un Cluster de Kubernetes, para lo cual han seleccionado la opción de RANCHER como solución de Gestión de K8s.

Esta opción les permite actualizar los nodos fácilmente y hacer o restaurar snapshots rápidamente en caso de fallo, además de ser la solución implementada On-Premise, lo que les permite tener alineados los entornos.





Cliente: KeepCoding Academy

Parámetros o requerimientos

- Toda la infraestructura debe estar **definida en código**.
- Se debe **automatizar el despliegue** todo lo posible mediante **GitOps**.
- Debe ser posible **trabajar de forma colaborativa** con el código de los despliegues.
- Los **secretos** y los **datos sensibles** deben almacenarse en una ubicación segura y **no se deben hardcodear en el código**.
- La infraestructura debe estar presente en **distintas zonas de disponibilidad**, para garantizar la alta disponibilidad y resiliencia.
- Toda la infraestructura debe implementarse **sólo en una región eu-central-1** y la posibilidad de implementarse en otras zonas debe estar limitada.
- El clúster debe poder **escalar fácilmente bajo demanda** en cualquier momento.
- El proyecto **debe estar documentado** en formato Markdown.



Cliente: KeepCoding Academy

Propuesta

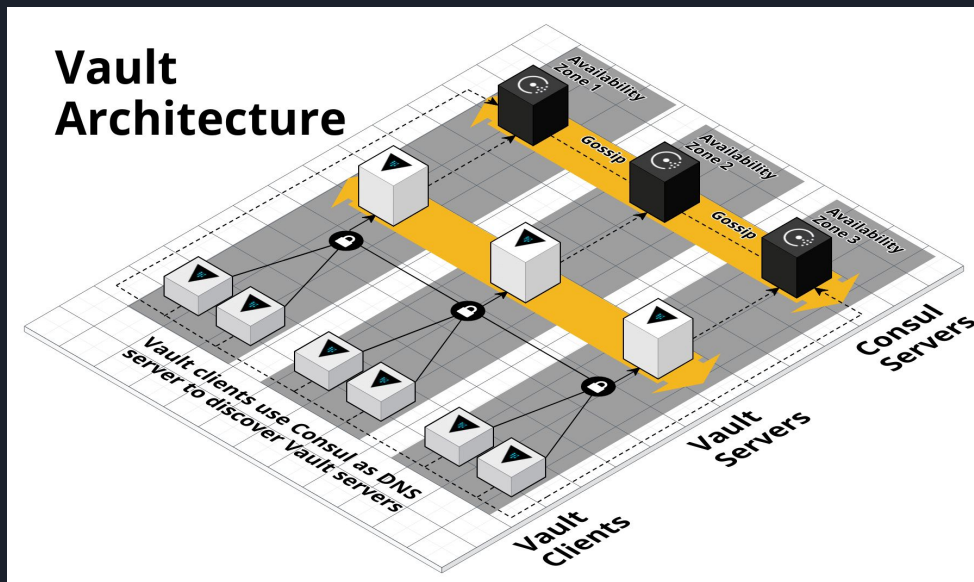
- La infraestructura se definirá en **HashiCorp Terraform**
- **Atlantis** se utilizará como herramienta de despliegue de infraestructura (CI/CD)
- Automatización de los despliegues de aplicación usando **ArgoCD**
- Se desplegará el clúster de **Kubernetes (Rancher RKE)**
- Todos los estados remotos de Terraform se guardarán en **Bucket S3** y **DynamoDB Table**, lo que permitirá el trabajo colaborativo entre desarrolladores/administradores.
- Las imágenes AMI utilizadas en el despliegue serán personalizadas con **HashiCorp Packer**.

Secretos y datos sensibles

Los datos sensibles se almacenan en un **clúster gossip de HashiCorp Consul** que se usa como Backend Storage

Todo se gestiona a través de HashiCorp Vault, que ofrece los secretos a Atlantis de forma segura en la instancia EC2

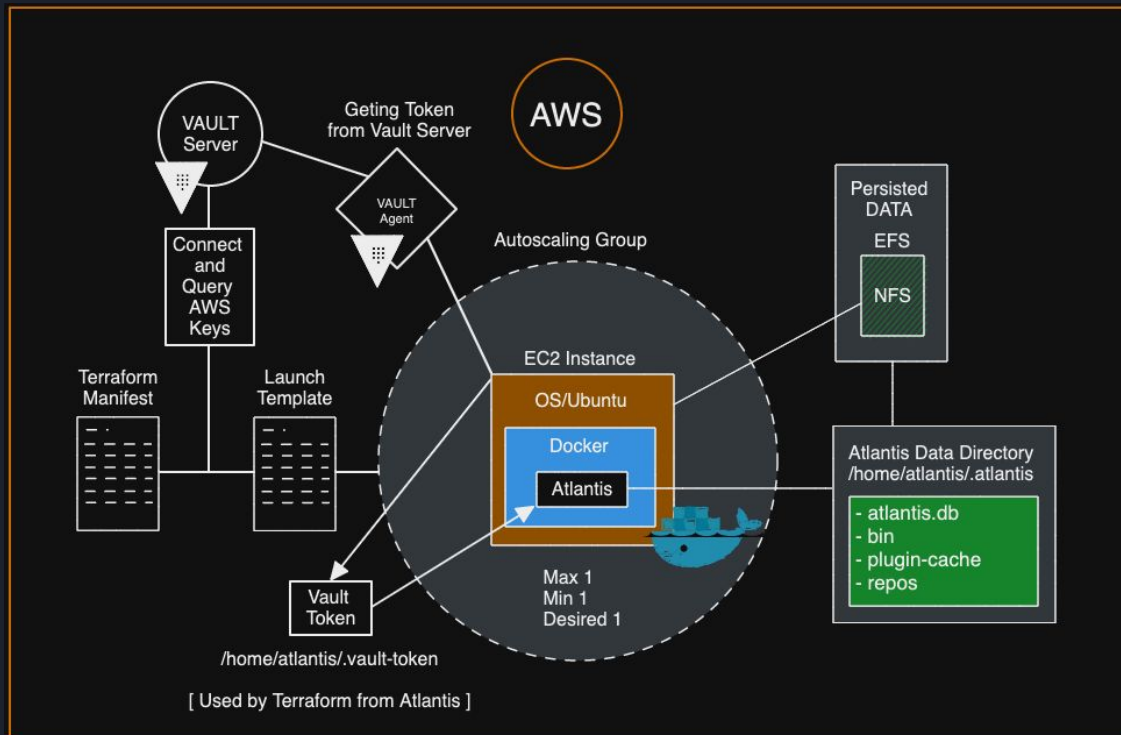
Esto evita tener datos sensibles en el código y garantiza la capa de seguridad



Cliente: KeepCoding Academy

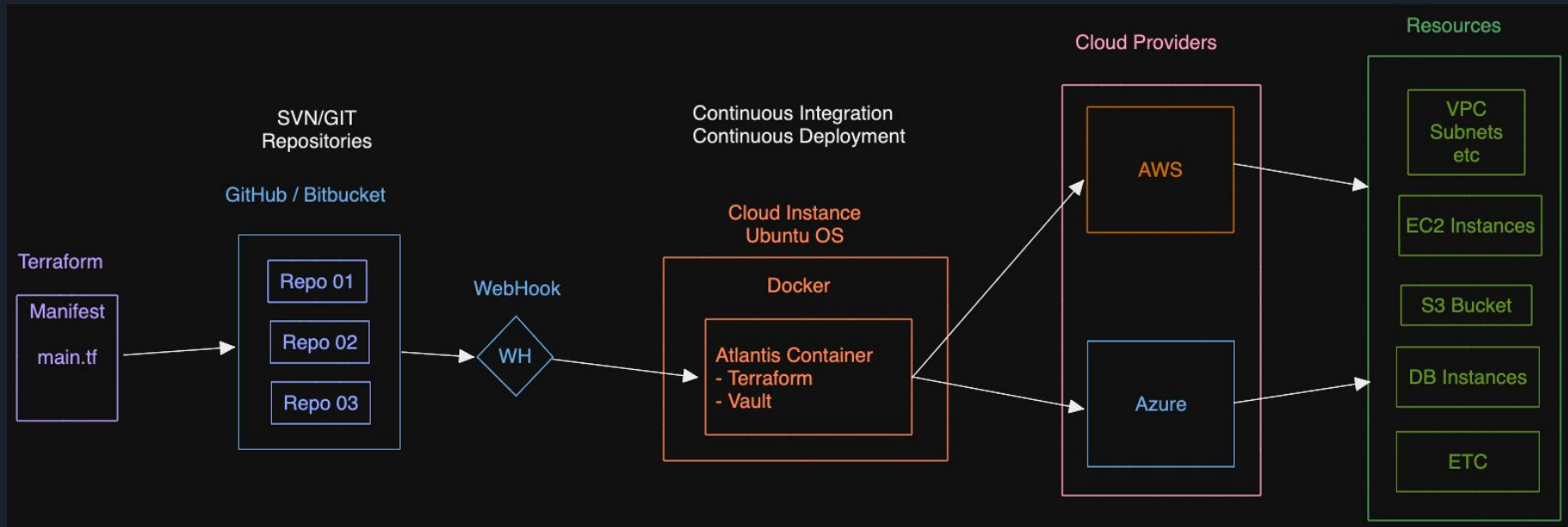
Atlantis (CI/CD)

- Grupo de Auto **Escalado**
- **Persistencia** de Datos en AWS EFS
- Acceso a **datos sensibles** en Vault
- Creación de ACCESS y SECRET keys en cada despliegue con un **TTL muy bajo**
- Todo **automatizado**



Cliente: KeepCoding Academy

Atlantis CI/CD Workflow



Ciente: KeepCoding Academy

Cluster Rancher k8s

Toda la infraestructura (incluyendo networking, storage, etc) está definida con **Terraform** y está desplegada en **distintas zonas de disponibilidad** para garantizar un entorno resiliente, todo en la región eu-central-1

El clúster de Kubernetes se implementa con **plantillas y pools de nodos**, lo que permite escalar hacia arriba o hacia abajo manualmente con un clic si es necesario tanto nodos MASTER como WORKERS.

Machine Pools						
Provisioning Log Snapshots Conditions Related Resources						
<div>Scale Down Download YAML Delete Filter</div>						
<input type="checkbox"/> State	Name	Node	OS	Roles	Age	
Pool: rancher-master-za-0 (1) Amazon EC2 - eu-central-1a / t3a.medium (template-t3a-medium-zona-a)						
<input type="checkbox"/> Active	m-4ffqr	rancher-master-za-0	Linux	Control Plane, Etcd	3.6 days	<div>- + ⋮</div>
Pool: rancher-master-zb-0 (1) Amazon EC2 - eu-central-1b / t3a.medium (template-t3a-medium-zona-b)						
<input type="checkbox"/> Active	m-grngw	rancher-master-zb-0	Linux	Control Plane, Etcd	3.6 days	<div>- + ⋮</div>
Pool: rancher-master-zc-0 (1) Amazon EC2 - eu-central-1c / t3a.medium (template-t3a-medium-zona-c)						
<input type="checkbox"/> Active	m-94g9d	rancher-master-zc-0	Linux	Control Plane, Etcd	3.6 days	<div>- + ⋮</div>
Pool: rancher-worker-za-0 (1) Amazon EC2 - eu-central-1a / t3a.medium (template-t3a-medium-zona-a)						
<input type="checkbox"/> Active	m-k79fb	rancher-worker-za-0	Linux	Worker	3.6 days	<div>- + ⋮</div>

Cliente: KeepCoding Academy

Despliegue en Atlantis

Ejemplo de despliegue del cluster de **Rancher** mediante un PR con un apply de toda la plataforma definida en **Terraform**

```
# KeepCoding DevOps & Cloud Computing Full Stack Bootcamp (Final Project)
# Team: KeepCoding Masters
# Staff: Jonas Marquez, Ruben Martin, Yllis Ramirez, Francesc Blanco
# IaC (Infrastructure as Code) & GitOps with Terraform + Atlantis & K8s (Rancher RKE in AWS)
# Technologies: HashiCorp Terraform & Packer, AWS, Docker, Atlantis
# -----

version: 3
automerge: false
delete_source_branch_on_merge: false
parallel_plan: false
parallel_apply: false
projects:
  - name: rancher
    dir: ../rancher
    workspace: nonprod
    terraform_version: v1.1.4
    delete_source_branch_on_merge: false
    autoplan:
      when_modified: [ "*.tf" ]
      enabled: true
    apply_requirements: [ mergeable, approved ]
    workflow: nonprod
```



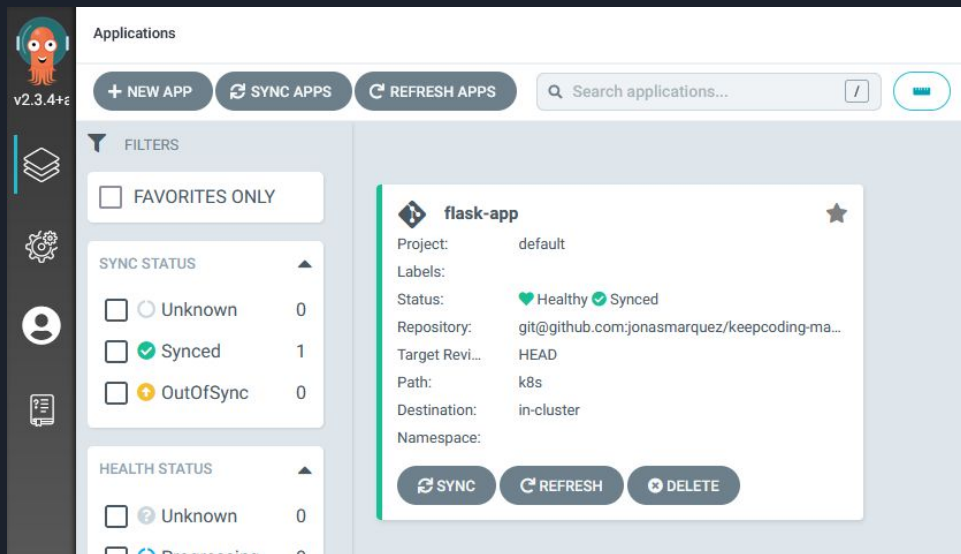
The screenshot displays a GitHub pull request interface. At the top, a green checkmark icon indicates that 'jonasmartinez' approved the changes on May 10. Below this, a comment from 'jonasmartinez' reads 'Go ahead!'. Further down, a comment from 'jonasmartinezkc' shows the command 'atlantis apply -p rancher'. Another comment from 'jonasmartinez' states 'Ran Apply for project: rancher dir: rancher workspace: nonprod' with a 'Show Output' link. At the bottom, a merge commit 'jonasmartinezkc merged commit 606024f into main on May 10' is shown, with '4 checks passed' and buttons for 'View details' and 'Revert'.

<https://github.com/jonasmartinezkc/atlantis-rancher-cluster/pull/5>

Ciente: KeepCoding Academy

GitOps con Argo CD

Todos los despliegues se aplican mediante **Argo CD** usando GitOps continuous delivery sobre el Cluster. Cualquier aplicación definida en el repositorio puede mantenerse actualizada automáticamente.



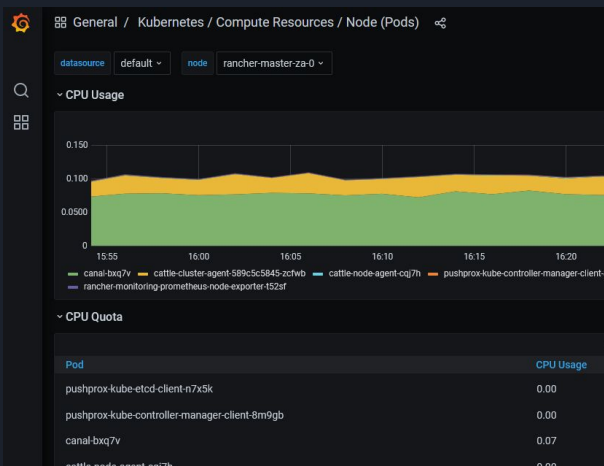
Cliente: KeepCoding Academy

Monitorización

Todos los componentes del cluster de **Rancher** están monitorizados con **Prometheus** y **Grafana**, incluyendo los pods de aplicación y los elementos de **ArgoCD**.



Cluster Metrics



Pods Metrics



Etcd Metrics

Cliente: KeepCoding Academy

Aplicación ejemplo

Se ha desplegado una aplicación web usando **Python, Flask y MySQL**. Esta aplicación hace una conexión de la base de datos de MySQL desde Flask, y realiza operaciones de crear, insertar y mostrar, para mantener una BD de estudiantes.

1. <http://18.193.71.161:32001/create-table>
2. <http://18.193.71.161:32001/add-students>
3. <http://18.193.71.161:32001/>

```
app.py
5 app = Flask(__name__)
6
7 app.config['MYSQL_USER'] = os.environ['MYSQL_USER']
8 app.config['MYSQL_PASSWORD'] = os.environ['MYSQL_PASSWORD']
9 app.config['MYSQL_HOST'] = os.environ['MYSQL_HOST']
10 app.config['MYSQL_DB'] = os.environ['MYSQL_DB']
11 mysql = MySQL(app)
12
13
14 @app.route('/create-table')
15 def createtable():
16     cursor = mysql.connection.cursor()
17     cursor.execute('CREATE TABLE IF NOT EXISTS students(id INT(5) PRIMARY KEY NOT NULL AUTO_INCREMENT,
18                                                         name VARCHAR(50) NOT NULL,
19                                                         email VARCHAR(100) NOT NULL,
20                                                         phone INT NOT NULL,
21                                                         address VARCHAR(250) NOT NULL)')
22     cursor.close()
23     return 'Tabla Creada'
24
25
26 @app.route('/add-students')
27 def addstudents():
28     cursor = mysql.connection.cursor()
29     cursor.execute('INSERT IGNORE INTO students (id,name,email,phone,address) VALUES('1','Pedro Romero','pedro_romero@gmail.com',657798564,'San Joan Despi')
30                                                         INSERT IGNORE INTO students (id,name,email,phone,address) VALUES('2','Nazaret Olivieri','nazaret_olivieri@gmail.com',610432907,'Cornella')')
31     cursor.close()
32     return 'Estudiantes añadidos del primer año'
33
34
35 @app.route('/')
36 def students():
37     s = "<table style='border:1px solid red'>"
38
39     cursor = mysql.connection.cursor()
40     cursor.execute('SELECT * FROM students;')
41     for row in cursor.fetchall():
42         s = s + "<tr>"
43         for x in row:
44             s = s + "<td>" + str(x) + "</td>"
45         s = s + "</tr>"
46
47     cursor.close()
```



Cliente: KeepCoding Academy

Demo time!





Cliente: KeepCoding Academy

Repositorios

- Repositorio con nuestro Proyecto final de KeepCoding DevOps V5

<https://github.com/jonasmarquez/keepcoding-masters-proyecto-final>

- Definición del cluster de k8s mediante Terraform y Atlantis

<https://github.com/jonasmarquezkc/atlasntis-rancher-cluster>



Cliente: KeepCoding Academy

¿Qué te ha aportado el desarrollar este proyecto?

La visión holística de pensar, planificar y desplegar todos los componentes necesarios en una solución real, desde la infraestructura hasta la aplicación final, incluyendo toda la automatización, documentación, monitorización y pipelines.

¿Qué has aprendido?

Automatización y estandarización para conseguir una integración y despliegue continuos.



Cliente: KeepCoding Academy

¿Qué es lo que no volverías a hacer de la misma manera?

Orientar el desarrollo dirigido a un único proveedor de servicios en la nube.

¿Qué cosas seguirías haciendo en el futuro para mejorar el proyecto?

Todo lo relativo a la seguridad (SSL/TLS).

Creación de nuevos templates con diversas configuraciones tanto para Rancher como para Workflows de Atlantis e imágenes AMI con Packer para agilizar la velocidad de los despliegues.

Despliegues con Helm.

Documentación automática (Swagger, DocFX).

Registro de métricas desde la Aplicación.

Persistencia de datos.

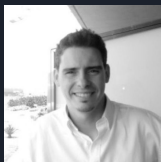


KeepCoding Masters Team

¡¡Muchas gracias!!



Jonás Márquez



Ruben Martin



Yilis Ramirez



Francesc Blanco