

Documentación	
<Reto-NightWatch>	

Información del documento

Nombre: RETO-NIGHTWATCH_v1.doc

Descripción: Documentación

Registro

Elaborado por	Micaela Plada	Fecha	[05/05/2025]
----------------------	---------------	--------------	--------------

Registro de Actualizaciones

N° de Versión	Fecha	Realizado por	Resumen de Actualización
[xy]	[dd/mm/aaaa]		

Registro de Aprobaciones

N° de Versión	Fecha	Revisado por	Aprobado?
[xy]	[dd/mm/aaaa]		<Aprobado / Aprobado con Modificaciones / Rechazado>

Documentación	
<Reto-NightWatch>	

INDICE

1. Descripción general.....3

 1.1. Propósito..... 3

 1.2. Objetivo..... 3

2. Estructura del Proyecto.....4

 2.1. Contenido de directorio “terraform”5-10

 2.2. Despliegue en AWS..... 11-12

 2.3. Kubernetes..... 13-19

 2.4. Monitoreo..... 20-24

 2.5. Ansible.....25-26

3. Arquitectura.....27

Documentación	
<Reto-NightWatch>	

Agenda WEB

1. Descripción general

1.1 Propósito

Simular un entorno de infraestructura moderna basado en la nube, capaz de alojar una aplicación web, automatizando su despliegue, configuración y monitoreo. Se busca aplicar herramientas actuales como AWS, Terraform, Docker, Kubernetes, Ansible y CloudWatch para construir una solución funcional que refleje buenas prácticas en DevOps e infraestructura como código. Este sistema también permite monitorear el estado de los servicios y responder ante incidentes de forma proactiva.

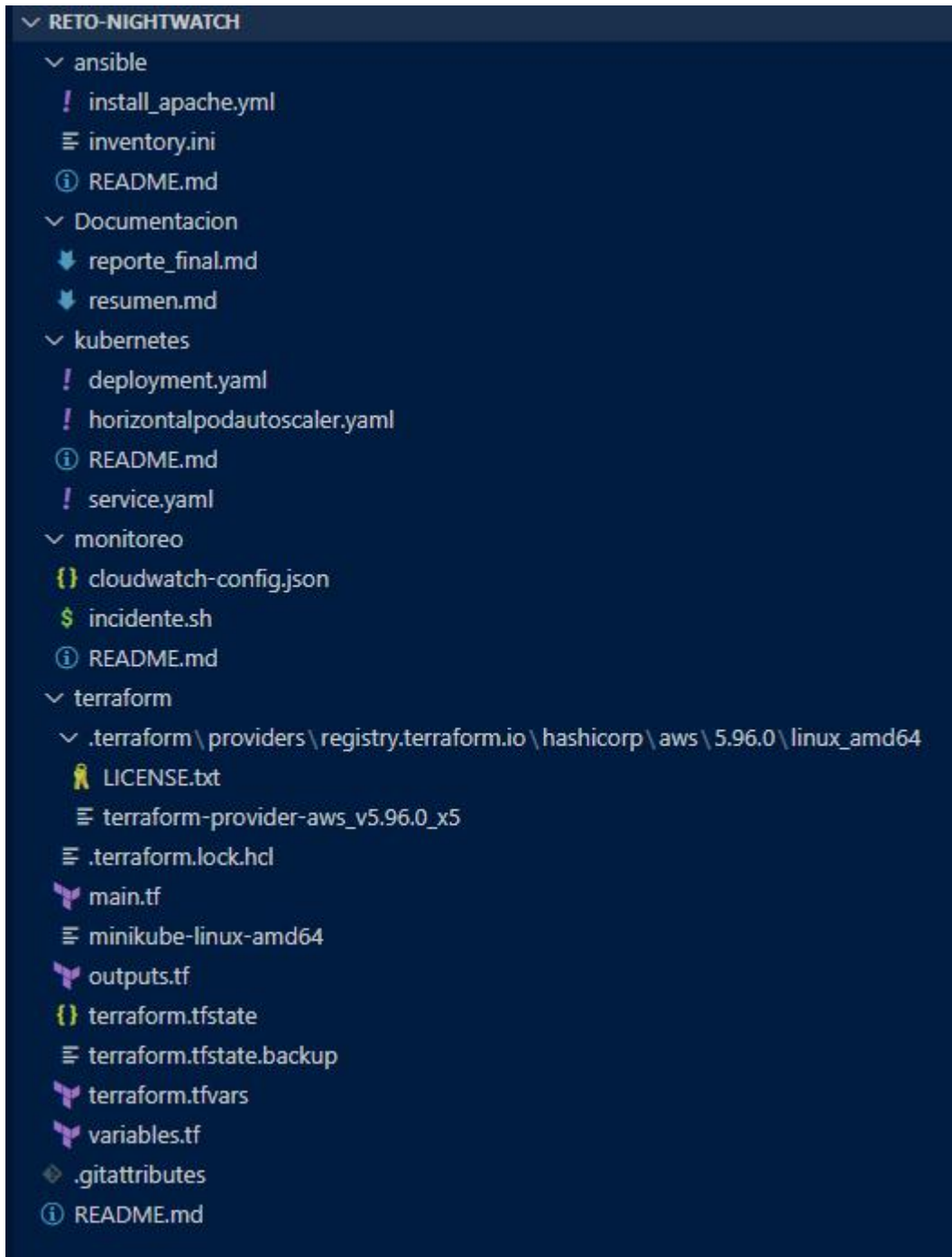
1.2 Objetivo

Desarrollar una solución técnica que incluya:

- La creación de infraestructura en AWS utilizando Terraform.
- El uso de Ansible para automatizar la instalación y configuración de un servidor web Apache en EC2.
- El despliegue de una aplicación web dentro de un contenedor Docker en un clúster Kubernetes local (Minikube).
- Utilizar CloudWatch para el monitoreo de métricas y posibles eventos críticos.
- Ejecutar un contenedor con CloudWatch Agent para enviar métricas al panel de monitoreo.
- Simular alta disponibilidad y escalabilidad básica usando contenedores.
- Gestionar la infraestructura desde un entorno WSL Ubuntu con Visual Studio Code.

Documentación	
<Reto-NightWatch>	

2. Estructura del Proyecto



Documentación	
<Reto-NightWatch>	

2.1 Contenido de directorio "terraform"

En main.tf:

```
1  provider "aws" {
2    |   region = var.aws_region
3  }
4
5  # VPC
6  resource "aws_vpc" "main" {
7    |   cidr_block = var.vpc_cidr
8  }
9
10 # Internet Gateway
11 resource "aws_internet_gateway" "gw" {
12 |   vpc_id = aws_vpc.main.id
13 }
14
15 # Subredes Públicas
16 resource "aws_subnet" "public_subnet" {
17 |   count                = 2
18 |   vpc_id               = aws_vpc.main.id
19 |   cidr_block           = var.public_subnets_cidr[count.index]
20 |   map_public_ip_on_launch = true
21 }
22
23 # Subredes Privadas
24 resource "aws_subnet" "private_subnet" {
25 |   count                = 2
26 |   vpc_id               = aws_vpc.main.id
27 |   cidr_block           = var.private_subnets_cidr[count.index]
28 }
29
30 # Security Group para instancias públicas
31 resource "aws_security_group" "public_sg" {
32 |   name                = "public-sg"
33 |   description         = "Permite SSH y HTTP"
34 |   vpc_id              = aws_vpc.main.id
35
36 |   ingress {
37 |     description = "Acceso SSH"
38 |     from_port   = 22
39 |     to_port     = 22
40 |     protocol    = "tcp"
41 |     cidr_blocks = ["0.0.0.0/0"]
42 |   }
43 }
```

Documentación	
<Reto-NightWatch>	

```
44 ingress {
45     description = "Acceso HTTP"
46     from_port   = 80
47     to_port     = 80
48     protocol    = "tcp"
49     cidr_blocks = ["0.0.0.0/0"]
50 }
51
52 egress {
53     from_port = 0
54     to_port   = 0
55     protocol  = "-1"
56     cidr_blocks = ["0.0.0.0/0"]
57 }
58 }
59
60 # Instancias públicas
61 resource "aws_instance" "public_instance" {
62     ami                = "ami-053b0d53c279acc90" # Ubuntu 22.04 Para us-east-1
63     instance_type      = var.instance_type
64     subnet_id          = aws_subnet.public_subnet[0].id
65     key_name            = var.key_name
66     vpc_security_group_ids = [aws_security_group.public_sg.id]
67
68     tags = {
69         Name = "PublicInstance"
70     }
71 }
72
73 # Instancias privadas
74 resource "aws_instance" "private_instance" {
75     ami                = "ami-053b0d53c279acc90" # Ubuntu 22.04 Para us-east-1
76     instance_type      = var.instance_type
77     subnet_id          = aws_subnet.private_subnet[0].id
78     key_name            = var.key_name
79     vpc_security_group_ids = [aws_security_group.public_sg.id]
80
81     associate_public_ip_address = false
82 }
```

Documentación	
<Reto-NightWatch>	


```
82
83     tags = {
84     |     Name = "PrivateInstance"
85     |     }
86     }
87
88     # Bucket S3
89     resource "aws_s3_bucket" "bucket" {
90     |     bucket = var.s3_bucket_name
91     |     force_destroy = true
92     |     }
93
```

Documentación	
<Reto-NightWatch>	

Outputs.tf:

```
terraform >  outputs.tf > ...
1  output "public_instance_ip" {
2      description = "IP pública de la instancia pública"
3      value       = aws_instance.public_instance.public_ip
4  }
5
6  output "private_instance_private_ip" {
7      description = "IP privada de la instancia privada"
8      value       = aws_instance.private_instance.private_ip
9  }
10
11 output "bucket_name" {
12     description = "Nombre del bucket creado"
13     value       = aws_s3_bucket.bucket.bucket
14 }
15
```

Terraform.tfvars:

```
terraform >  terraform.tfvars > ...
1  aws_region = "us-east-1"
2
3  vpc_cidr = "10.0.0.0/16"
4
5  public_subnets_cidr = ["10.0.10.0/24", "10.0.20.0/24"]
6
7  private_subnets_cidr = ["10.0.30.0/24", "10.0.40.0/24"]
8
9  instance_type = "t2.micro"
10
11 key_name = "reto-nightwatch-key"
12
13 s3_bucket_name = "reto-nightwatch-bucket"
14
```


Documentación	
<Reto-NightWatch>	

Variables.tf:

```
terraform > variables.tf > ...
1  variable "aws_region" {
2      description = "AWS region"
3      type        = string
4  }
5
6  variable "vpc_cidr" {
7      description = "VPC CIDR block"
8      type        = string
9  }
10
11 variable "public_subnets_cidr" {
12     description = "List of public subnet CIDRs"
13     type        = list(string)
14 }
15
16 variable "private_subnets_cidr" {
17     description = "List of private subnet CIDRs"
18     type        = list(string)
19 }
20
21 variable "instance_type" {
22     description = "Type of EC2 instance"
23     type        = string
24 }
25
26 variable "key_name" {
27     description = "SSH key pair name"
28     type        = string
29 }
30
31 variable "s3_bucket_name" {
32     description = "Name of the S3 bucket"
33     type        = string
34 }
35
```

Documentación	
<Reto-NightWatch>	

Para comprobar que lo que configuramos en terraform funcione hay que ir ejecutando en una terminal, los siguientes comandos:

- terraform init
- terraform plan
- terraform apply

Una vez que se haya ejecutado terraform apply y haya quedado todo OK:

```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

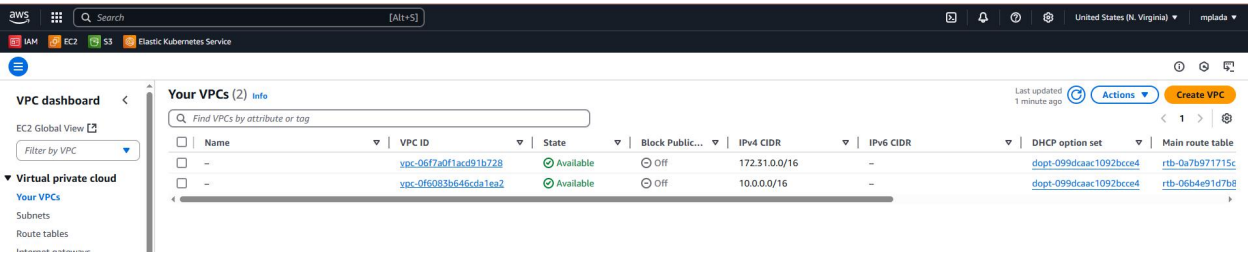
Outputs:

bucket_name = "reto-nightwatch-bucket"
private_instance_private_ip = "10.0.30.145"
public_instance_ip = "3.85.224.45"
micap@DESKTOP-3QS7DRR:/mnt/c/Users/User/Documents/reto-nightwatch/terraform$
```

2.2 Despliegue en AWS

Automaticamente en AWS debería ir tomando la siguiente configuración:

VPC:



Subredes privadas y publicas:

Subnet associations (4)					
Filter subnet associations					
Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	
-	subnet-0ea0e30a96ec9a3a3	acl-0453632d44a75211d	us-east-1f	10.0.30.0/24	
-	subnet-08377763ed8b2810a	acl-0453632d44a75211d	us-east-1f	10.0.20.0/24	
-	subnet-05581e2f66e2d8c5a	acl-0453632d44a75211d	us-east-1f	10.0.40.0/24	
-	subnet-0ed563d0dc5b64a	acl-0453632d44a75211d	us-east-1f	10.0.10.0/24	

Documentación

<Reto-NightWatch>

Security Groups:

VPC dashboard

EC2 Global View

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Security Groups (3)

Find security groups by attribute or tag

	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input type="checkbox"/>	-	sg-01d5bce5a3531c18	default	vpc-06f7a0f1acd91b728	default VPC security group	008076722074
<input type="checkbox"/>	-	sg-00a5514b0cfd9f1c	default	vpc-0f6083b646cda1ea2	default VPC security group	008076722074
<input type="checkbox"/>	-	sg-0cdaf145dc55d990e	public-sg	vpc-0f6083b646cda1ea2	Permite SSH y HTTP	008076722074

Permite SSH y HTTP:

EC2

Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

sg-0cdaf145dc55d990e - public-sg

Details

Security group name: public-sg

Security group ID: sg-0cdaf145dc55d990e

Description: Permite SSH y HTTP

VPC ID: vpc-0f6083b646cda1ea2

Owner: 008076722074

Inbound rules count: 2 Permission entries

Outbound rules count: 1 Permission entry

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (2)

	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-0d83b5ad5398dbf8d	IPv4	SSH	TCP	22	0.0.0.0/0	Acceso SSH
<input type="checkbox"/>	-	sgr-0d81206d0557f265	IPv4	HTTP	TCP	80	0.0.0.0/0	Acceso HTTP

EC2 (dos instancias, una publica y otra privada):

Instances (2)

Find Instance by attribute or tag (case-sensitive)

All states

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	PublicInsta...	i-0158ca486a4d42a42	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1f	-	54.87.228.183	54.87.228.183
<input type="checkbox"/>	PrivateInsta...	i-0c80f5d779857b81	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1f	-	-	-

Bucket S3:

Amazon S3

General purpose buckets

Directory buckets

Table buckets

Access Grants

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Account snapshot - updated every 24 hours

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

View Storage Lens dashboard

General purpose buckets

Directory buckets

General purpose buckets (1)

Buckets are containers for data stored in S3.

Find buckets by name

	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	reto-nightwatch-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	May 1, 2025, 23:22:24 (UTC-03:00)

Documentación	
<Reto-NightWatch>	

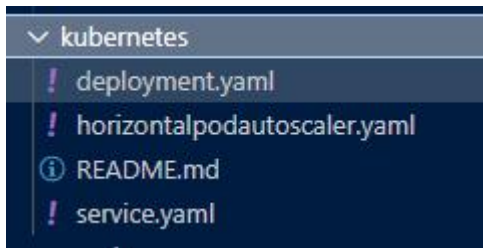
2.3 Kubernetes

Objetivo del Reto:

Crear un clúster de Kubernetes (puede ser local), desplegar un servicio web que responda solicitudes, y analizar escalabilidad y buenas prácticas para alta disponibilidad.

Elección Tecnológica:

Se utilizó Minikube desplegado sobre Docker como solución local para simular un entorno Kubernetes funcional, con todas las capacidades necesarias para pruebas y desarrollo.



Documentación	
<Reto-NightWatch>	

En deployment.yaml se despliega un servidor web **Apache** dentro de un clúster local de Kubernetes usando Minikube.

El mismo crea un “Deployment” con 2 réplicas del servidor Apache (httpd:latest). Lo que significa, permite balanceo básico y tolerancia a fallos.

```
kubernetes > ! deployment.yaml
1 #Despliegue de Apache
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: apache-deployment
6 spec:
7   replicas: 2
8   selector:
9     matchLabels:
10      app: apache
11   template:
12     metadata:
13       labels:
14         app: apache
15     spec:
16       containers:
17         - name: apache
18           image: httpd:latest
19           ports:
20             - containerPort: 80
21
```

Documentación	
<Reto-NightWatch>	

En service.yaml expone el servicio mediante NodePort para acceder desde el navegador.

```
kubernetes > ! service.yaml
1  #Para exponer al servicio
2  apiVersion: v1
3  kind: Service
4  metadata:
5    name: apache-service
6  spec:
7    selector:
8      app: apache
9    type: NodePort
10   ports:
11     - port: 80
12       targetPort: 80
13       nodePort: 30080
14
15
```

Para aplicar los archivos yaml, se ejecuta los comandos:

```
kubectl apply -f deployment.yaml
```

```
kubectl apply -f service.yaml
```

Utilicé la herramienta kubectl (CLI para interactuar con Kubernetes).

Se instala minikube en terminal:

```
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/terraform$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 119M 100 119M 0 0 21.7M 0 0:00:05 0:00:05 --:--:-- 29.0M
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/terraform$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/terraform$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/terraform$
```

Documentación	
<Reto-NightWatch>	

Se le indica que se inicie minikube desde docker con el comando “*minikube start --driver=docker*”. Se ejecuta el comando “*kubectl get nodes*” para obtener una lista de los nodos que tiene un cluster.

```

🔧 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
  ▪ Generating certificates and keys ...
  ▪ Booting up control plane ...
  ▪ Configuring RBAC rules ...
🔗 Configuring bridge CNI (Container Networking Interface) ...
🔍 Verifying Kubernetes components...
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
★ Enabled addons: storage-provisioner, default-storageclass

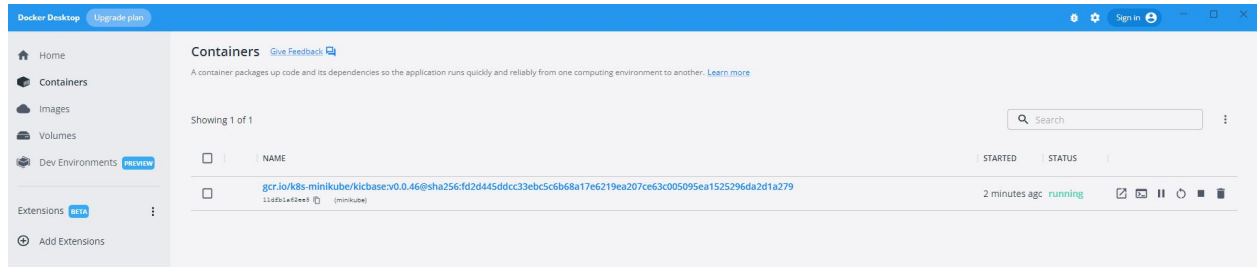
! /usr/local/bin/kubectl is version 1.21.13, which may have incompatibilities with Kubernetes 1.32.0.
  ▪ Want kubectl v1.32.0? Try 'minikube kubectl -- get pods -A'
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
micap@DESKTOP-3QS7DRR:/mnt/c/Users/User/Documents/reto-nightwatch/terraform$ kubectl get nodes
NAME        STATUS    ROLES    AGE   VERSION
minikube    Ready     control-plane   33s   v1.32.0
micap@DESKTOP-3QS7DRR:/mnt/c/Users/User/Documents/reto-nightwatch/terraform$

```


Documentación	
<Reto-NightWatch>	

En Docker:

En la siguiente imagen se puede ver activo el nodo minikube creado.



Escalabilidad:

Se piensa en un Autoscaler que escala automáticamente el número de réplicas de “Deployment” según el uso de CPU.

```
kubernetes > ! horizontalpodautoscaler.yaml
1  apiVersion: autoscaling/v2
2  kind: HorizontalPodAutoscaler
3  metadata:
4    name: apache-hpa
5  spec:
6    scaleTargetRef:
7      apiVersion: apps/v1
8      kind: Deployment
9      name: apache-deployment
10 minReplicas: 2
11 maxReplicas: 5
12 metrics:
13 - type: Resource
14   resource:
15     name: cpu
16     target:
17       type: Utilization
18       averageUtilization: 50
19
```

Requiere que el **Metrics Server** esté instalado en el cluster. Como utilice minikube, se instala ejecutando lo siguiente en la terminal: *minikube addons enable metrics-server*

Documentación	
<Reto-NightWatch>	

```
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$ minikube addons enable metrics-server
! metrics-server is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
★ The 'metrics-server' addon is enabled
```

Para aplicarlo, basta con ejecutar el siguiente comando:

kubectl apply -f horizontalpodautoscaler.yaml

```
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$ kubectl apply -f horizontalpodautoscaler.yaml
horizontalpodautoscaler.autoscaling/apache-hpa created
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$
```

Alta Disponibilidad:

- Uso de réplicas múltiples (se utiliza al menos 2) para evitar un único punto de fallo.
- Servicio que balancea la carga entre los pods disponibles.
- Posibilidad de escalar automáticamente según demanda.
- Kubernetes recrea los pods fallidos de forma automática.

Probamos comportamiento de alta disponibilidad. Primero comenzamos en ver la cantidad de pods, ejecutando el comando “*kubectl get pods*”:

```
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$ kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
apache-deployment-5fd955856f-6h5m2  0/1     ContainerCreating  0          5s
apache-deployment-5fd955856f-h5rp7  0/1     ContainerCreating  0          5s
```

Se hace la prueba de eliminar un pod ejecutando “*kubectl delete pod apache-deployment-5fd955856f-6h5m2*”:

```
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$ kubectl delete pod apache-deployment-5fd955856f-6h5m2
pod "apache-deployment-5fd955856f-6h5m2" deleted
```

Luego, verificamos que Kubernetes cree uno ejecutando el comando “*kubectl get pods -w*”:

```
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$ kubectl get pods -w
NAME                                READY   STATUS    RESTARTS   AGE
apache-deployment-5fd955856f-dnqmz  1/1     Running   0          45s
apache-deployment-5fd955856f-h5rp7  1/1     Running   0          4m16s
```

Se monitorea utilizando: *kubectl get hpa* (lista de HPA definidos en el cluster) y *kubectl top pods* (muestra el uso de recursos en un cluster)

```
^micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$ kubectl get hpa
NAME      REFERENCE                               TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
apache-hpa  Deployment/apache-deployment            cpu: <unknown>/50%  2         5         2          45m
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$ kubectl top pods
W0506 08:27:15.972178    2991 top_pod.go:140] Using json format to get metrics. Next release will switch to protocol-buffers, switch early by passing --use-protocol-buffers flag
NAME                                CPU(cores)   MEMORY(bytes)
apache-deployment-5fd955856f-dnqmz  1m           29Mi
apache-deployment-5fd955856f-h5rp7  1m           29Mi
```

Documentación	
<Reto-NightWatch>	

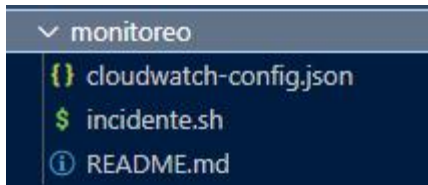
En resumen:

Utilicé, de forma local, Minikube en Docker ya que simula un entorno Kubernetes real, ideal para pruebas y entornos de desarrollo, y permite desplegar aplicaciones, exponer servicios, escalar pods, etc.

Desplugué una aplicación web (Apache) usando un Deployment con réplicas, y la expuse mediante un Service tipo NodePort. También, incluí un “HorizontalPodAutoscaler” para mostrar cómo el sistema puede escalar según el uso de CPU. En cuanto a la alta disponibilidad, destaqué la importancia de tener múltiples réplicas y balanceo de carga, aún en entornos locales.

Documentación	
<Reto-NightWatch>	

2.4 Monitoreo



Se configura Monitoreo y Alertas usando CloudWatch y custom metrics.

Dado que se configuro a nivel local y no se puede usar CloudWatch Metrics de EKS/EC2 directamente, se instala el CloudWatch Agent como DaemonSet, configurado para: enviar métricas básicas de CPU y memoria desde los nodos e incluir métricas personalizadas.

Se configura una alarma en AWS CloudWatch vía consola y se selecciona la opción de que envíe alerta a un SNS topic para que me llegue a mi mail personal.

Documentación	
<Reto-NightWatch>	

Se ejecuta el CloudWatch Agent en un contenedor usando Docker, con el objetivo de enviar métricas desde una instancia o entorno local a Amazon CloudWatch.

Se utiliza comando docker run para ejecutar CloudWatch-agent:

```
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$ docker run -d --name cloudwatch-agent \
WS_A> -e AWS_ACCESS_KEY_ID=AKIAQDYLI76NIZYDQZEJ \
AWS> -e AWS_SECRET_ACCESS_KEY=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 \
> -e REGION=us-east-1 \
> -v /var/run/docker.sock:/var/run/docker.sock \
> amazon/cloudwatch-agent
Unable to find image 'amazon/cloudwatch-agent:latest' locally
latest: Pulling from amazon/cloudwatch-agent
d43edd530f6c: Pull complete
083c77d3529a: Pull complete
a8fa4858fc1c: Pull complete
Digest: sha256:7aaed409cc7bfc61799f2eebeaf7f07d753530a893fc1e0ffee0e375a7f9d52e
Status: Downloaded newer image for amazon/cloudwatch-agent:latest
8d3025abcc11cfccdbabde8d15a76a421c3346ef0960253ebe350d6a3ee8e27
micap@DESKTOP-3Q57DRR:/mnt/c/Users/User/Documents/reto-nightwatch/kubernetes$
```

En docker:

Containers
[Give Feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Showing 2 of 2

<input type="checkbox"/>	NAME
<input type="checkbox"/>	gcr.io/k8s-minikube/kicbase:v0.0.46@sha256:fd2d445ddcc33ebc5c6b68a17e6219ea207ce63c005095ea1525296da2d1a279 11dfb1a62ee5 (minikube)
<input type="checkbox"/>	amazon/cloudwatch-agent 8d3025abcc11 (cloudwatch-agent)

Documentación	
<Reto-NightWatch>	

Creamos un archivo de configuración para el agente Cloudwatch (en el archivo se define las metricas a utilizar, en este caso, CPU):

```
monitoreo > {} cloudwatch-config.json > {} metrics
1  {
2    "metrics": {
3      "namespace": "Minikube/Metrics",
4      "metrics_collected": {
5        "cpu": {
6          "measurement": [
7            "cpu_usage_idle",
8            "cpu_usage_user",
9            "cpu_usage_system"
10         ],
11         "metrics_collection_interval": 30
12       },
13       "mem": {
14         "measurement": [
15           "mem_used_percent"
16         ],
17         "metrics_collection_interval": 30
18       }
19     }
20   }
21 }
22
```

<h1>Documentación</h1>	
<h2><Reto-NightWatch></h2>	

Se crea una alarma para que de envíe una notificación a una casilla de correo:

CloudWatch > Alarms > Create alarm

Alarm recommendations available
Turn on Recommendations to pre-populate the wizard with the recommended alarms.

Step 1: Specify metric and conditions
Step 2: **Configure actions**
Step 3: Add name and description
Step 4: Preview and create

Configure actions

Notification

Alarm state trigger
Define the alarm state that will trigger this action.

☒ **In alarm**
The metric or expression is outside of the defined threshold.

☐ **OK**
The metric or expression is within the defined threshold.

☐ **Insufficient data**
The alarm has just started or not enough data is available.

Send a notification to the following SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

☐ Select an existing SNS topic
☒ **Create new topic**
☐ Use topic ARN to notify other accounts

Create a new topic...
The topic name must be unique.

Default_CloudWatch_Alarms_Topic

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (_).

Email endpoints that will receive the notification...
Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

micaela.plada5@gmail.com

user1@example.com, user2@example.com

[Create topic](#)
[Add notification](#)

Confirmación:

AWS Notification - Subscription Confirmation Recibidos x

AWS Notifications <no-reply@sns.amazonaws.com> 0:55 (hace 2 minutos)

para mí

[Traducir al español](#)

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:008076722074:Default_CloudWatch_Alarms_Topic

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

[Responder](#) [Reenviar](#) [😊](#)

Documentación
<Reto-NightWatch>

CloudWatch

Favorites and recents

Dashboards

Al Operations Preview

Alarms 0 0 0 0

In alarm

All alarms

Billing

Logs New

CloudWatch > Alarms

Successfully created alarm CPU-ApacheMinikube. View alarm

Alarms (1)

Hide Auto Scaling alarms

Clear selection

Create composite alarm

Actions

Create alarm

Search

Alarm state: Any

Alarm type: Any

Actions status: Any

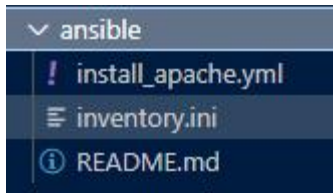
< 1 >

	Name	State	Last state update (UTC)	Conditions
<input type="checkbox"/>	CPU-ApacheMinikube	Insufficient data	2025-05-06 03:57:15	CPUUtilization > 70 for 1 datapoints with minutes

Se pueden crear mas alarmas, como por ejemplo, StatusCheckFailed_Instance (detecta si hay fallas en el sistema).

Documentación	
<Reto-NightWatch>	

2.5 Ansible



Este playbook instala y configura un servidor Apache en una instancia EC2 (con Ubuntu) usando Ansible y luego crea una página web de prueba.

```

ansible > ! install_apache.yml
You, 18 hours ago | 1 author (You)
1  - name: Instalar Apache en EC2 Ubuntu
2    hosts: web
3    become: yes
4    tasks:
5
6      - name: Actualizar el caché de apt
7        apt:
8          update_cache: yes
9          cache_valid_time: 3600 # Fuerza una actualización del caché de los repositorios de paquetes antes de instalar
10
11     - name: Instalar Apache2
12       apt:
13         name: apache2 #Especifica el paquete que se va a instalar
14         state: present #Asegura que el paquete esté instalado. Si ya está instalado, no hará nada.
15
16     - name: Iniciar el servicio de Apache2
17       service:
18         name: apache2
19         state: started
20         enabled: yes
21
22     - name: Crear página de prueba
23       copy:
24         dest: /var/www/html/index.html
25         content: "<h1>Apache desplegado con Ansible</h1>"

```

Documentación	
<Reto-NightWatch>	

Definimos un host llamado ec2 dentro de un grupo de servidores llamado [web] y especifica cómo conectarse a esa máquina con SSH.

```
ansible >  inventory.ini
You, 18 hours ago | 1 author (You)
1  [web]
2  ec2 ansible_host=54.87.228.183 ansible_user=ubuntu ansible_ssh_private_key_file=~/.ssh/reto-key.pem
3
```

Para verificar que apache este corriendo, primero hay que conectarse de la siguiente manera:

```
ssh -i ~/.ssh/reto-key.pem ubuntu@54.87.228.183
```

Corroboramos su estado:

```
sudo systemctl status apache2
```

Si dice inactive o failed, entonces ejecutamos:

```
sudo systemctl restart apache2
```

Documentación	
<Reto-NightWatch>	

3. Arquitectura

