

Manual de Creación de Infraestructura en Azure y Despliegue de Aplicación

Este manual documenta los pasos realizados para el desarrollo del Test – Solution and Explanations para la creación de una infraestructura básica en **Azure** utilizando **Terraform**, la configuración de **Docker** y **contenedores**, y el despliegue continuo con **Azure DevOps**.

Todo el proceso se realizó desde un terminal Linux y la última versión de **Terraform** y **Azure CLI**, empleando **Visual Studio Code** como entorno de desarrollo para editar el script y crear una APP en Node.js, Express y MongoDB para implementarlo en los puntos 2 y 3 del test.

Creación de infraestructura en Azure usando Terraform

1. Configuración del Entorno

Herramientas utilizadas:

- **Sistema Operativo:** Linux
- **Visual Studio Code:** Editor de texto utilizado para escribir el script Terraform.
- **Terraform:** Herramienta de infraestructura como código (IaC), versión más reciente.
- **Azure CLI:** Utilizada para autenticar en Azure mediante el comando `az login`.

```
alaincervantes:~/DevOps/Terraform/DevOps_Test_Solutions$ az login
```

Una vez autenticado con Azure CLI, se procedió a trabajar con los scripts de Terraform.

2. Archivo principal main.tf

```
alaincervantes:~/DevOps/Terraform/DevOps_Test_Solutions/script-01$ touch main.tf
```

Este archivo contiene todo el código para crear los recursos en Azure. A continuación, se describe cada uno de los recursos creados:

3. Definición del Proveedor

El primer paso es definir el proveedor de Azure y la suscripción a utilizar. En este caso, se estableció la suscripción de Azure mediante su ID.

```
main.tf  X
script-01 > main.tf
1  provider "azurerm" {
2      features {}
3
4      subscription_id = "30968ef3-435a-4ee2-8c06-7bb66267b87d"
5  }
6
```

4. Creación de Recursos

a. Grupo de Recursos (Resource Group)

El grupo de recursos es el contenedor lógico en Azure donde se almacenan todos los recursos.

```

37 resource "azurerm_resource_group" "rg" {
38   name      = "rg-acervantes"
39   location  = "Brazil South"
40 }
41

```

b. Red Virtual (Virtual Network) y Subred

Se creó una red virtual para permitir la comunicación entre los recursos y una subred específica para la máquina virtual.

```

42 resource "azurerm_virtual_network" "default" {
43   name            = "vnet-acervantes"
44   address_space   = ["10.0.0.0/16"]
45   location        = azurerm_resource_group.rg.location
46   resource_group_name = azurerm_resource_group.rg.name
47 }
48
49 resource "azurerm_subnet" "default" {
50   name            = "subnet-acervantes-app"
51   resource_group_name = azurerm_resource_group.rg.name
52   virtual_network_name = azurerm_virtual_network.default.name
53   address_prefixes   = ["10.0.1.0/24"]
54 }
55

```

c. Máquina Virtual (Linux VM)

Se creó una máquina virtual basada en Ubuntu con un tamaño estándar.

```

68 resource "azurerm_linux_virtual_machine" "vm" {
69   name            = "vm-acervantes-web01"
70   resource_group_name = azurerm_resource_group.rg.name
71   location        = azurerm_resource_group.rg.location
72   size            = "Standard_DS1_v2"
73   network_interface_ids = [azurerm_network_interface.nic.id]
74
75   os_disk {
76     caching           = "ReadWrite"
77     storage_account_type = "Standard_LRS"
78   }
79
80   source_image_reference {
81     publisher = "Canonical"
82     offer     = "UbuntuServer"
83     sku       = "18.04-LTS"
84     version   = "latest"
85   }
86
87   admin_username = "alaincervantesp"
88   admin_password = "Ac890816*"
89   disable_password_authentication = false
90 }
91

```

d. Base de Datos MySQL Flexible

Se desplegó una instancia de servidor MySQL con una base de datos específica.

```
resource "azurerm_mysql_flexible_server" "default" {
  name                       = "mysql-acervantes-db"
  location                  = azurerm_resource_group.rg.location
  resource_group_name       = azurerm_resource_group.rg.name
  sku_name                  = "GP_Standard_D2ds_v4"
  backup_retention_days     = 7
  geo_redundant_backup_enabled = false
  administrator_login       = random_string.name.result
  administrator_password    = random_password.password.result
  private_dns_zone_id       = azurerm_private_dns_zone.default.id
  version                   = "8.0.21"

  storage {
    iops    = 360
    size_gb = 20
  }

  depends_on = [azurerm_private_dns_zone_virtual_network_link.default]
}

resource "azurerm_mysql_flexible_database" "main" {
  charset       = "utf8mb4"
  collation     = "utf8mb4_unicode_ci"
  name          = "db-acervantes-app"
  resource_group_name = azurerm_resource_group.rg.name
  server_name    = azurerm_mysql_flexible_server.default.name
}
```

e. IP Pública y Balanceador de Carga

Se creó una dirección IP pública y un balanceador de carga para distribuir el tráfico a través de los recursos.

```
128 resource "azurerm_public_ip" "public_ip" {
129   name           = "acervantes-ip-publica"
130   location       = azurerm_resource_group.rg.location
131   resource_group_name = azurerm_resource_group.rg.name
132   allocation_method = "Static"
133 }
134
135 resource "azurerm_lb" "loadbalancer" {
136   name           = "lb-acervantes-web"
137   location       = azurerm_resource_group.rg.location
138   resource_group_name = azurerm_resource_group.rg.name
139
140   frontend_ip_configuration {
141     name           = "PublicIPAddress"
142     public_ip_address_id = azurerm_public_ip.public_ip.id
143   }
144 }
145
```

5. Ejecución de los Comandos

Se utilizaron los siguientes comandos de Terraform para crear la infraestructura:

1. **terraform init:** Inicializa el directorio de trabajo y descarga los proveedores necesarios.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
alaincervantes:~/DevOps/Terraform/DevOps_Test_Solutions/script-01$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/random...
- Finding latest version of hashicorp/azurerm...
- Installing hashicorp/random v3.6.3...
- Installed hashicorp/random v3.6.3 (signed by HashiCorp)
- Installing hashicorp/azurerm v4.4.0...
- Installed hashicorp/azurerm v4.4.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
alaincervantes:~/DevOps/Terraform/DevOps_Test_Solutions/script-01$

```

2. **terraform plan:** Muestra el plan de ejecución, indicando los recursos que se van a crear, modificar o destruir.

```

rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
alaincervantes:~/DevOps/Terraform/DevOps_Test_Solutions/script-01$ terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# azure_rm_loadbalancer will be created
+ resource "azure_rm_loadbalancer" "loadbalancer" {
+ id = (known after apply)
+ location = "brazilsouth"
+ name = "lb-acervantes-web"
+ private_ip_address = (known after apply)
+ private_ip_addresses = (known after apply)
+ resource_group_name = "rg-acervantes"
+ sku = "Standard"
+ sku_tier = "Regional"

+ frontend_ip_configuration {
+ gateway_load_balancer_frontend_ip_configuration_id = (known after apply)
+ id = (known after apply)
+ inbound_nat_rules = (known after apply)
+ load_balancer_rules = (known after apply)
+ name = "PublicIPAddress"
+ outbound_rules = (known after apply)
+ private_ip_address = (known after apply)
+ private_ip_address_allocation = (known after apply)
+ private_ip_address_version = (known after apply)
+ public_ip_address_id = (known after apply)
+ public_ip_prefix_id = (known after apply)
+ subnet_id = (known after apply)
}

# azure_rm_loadbalancer_backend_address_pool.backend_pool will be created
+ resource "azure_rm_loadbalancer_backend_address_pool" "backend_pool" {
+ backend_ip_configurations = (known after apply)
+ id = (known after apply)
+ inbound_nat_rules = (known after apply)
+ load_balancing_rules = (known after apply)
+ loadbalancer_id = (known after apply)
+ name = "backend-pool"
+ outbound_rules = (known after apply)
}

# azure_rm_loadbalancer_probe.lb_probe will be created
+ resource "azure_rm_loadbalancer_probe" "lb_probe" {
+ id = (known after apply)
+ interval_in_seconds = 5
+ load_balancing_rules = (known after apply)
+ loadbalancer_id = (known after apply)
+ name = "tcp-probe"
+ number_of_probes = 2
}

```

3. terraform apply: Aplica los cambios definidos en los archivos de Terraform y despliega la infraestructura en Azure.

```
azurerm_lb_rule.lb_rule: Creation complete after 6s [id=/subscriptions/30968ef3-435a-4ee2-8c06-7bb66267b87d/resourceGroups/rg-acervantes/providers/Microsoft.Network/loadBalancers/lb-acervantes-web/loadBalancingRules/http-lb-rule]
azurerm_private_dns_zone.virtual_network_link.default: Still creating... [10s elapsed]
azurerm_linux_virtual_machine.vm: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.vm: Creation complete after 28s [id=/subscriptions/30968ef3-435a-4ee2-8c06-7bb66267b87d/resourceGroups/rg-acervantes/providers/Microsoft.Compute/virtualMachines/vm-acervantes-web01]
azurerm_private_dns_zone.virtual_network_link.default: Still creating... [20s elapsed]
azurerm_private_dns_zone.virtual_network_link.default: Creation complete after 34s [id=/subscriptions/30968ef3-435a-4ee2-8c06-7bb66267b87d/resourceGroups/rg-acervantes/providers/Microsoft.Network/privateDnsZones/acervantes.mysql.database.azure.com/virtualNetworkLinks/mysql5VnetZoneuxcfqhl.com]
azurerm_mysql_flexible_server.default: Creating...
azurerm_mysql_flexible_server.default: Still creating... [10s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [20s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [30s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [40s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [50s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [1m0s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [1m10s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [1m20s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [1m30s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [1m40s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [1m50s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [2m0s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [2m10s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [2m20s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [2m30s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [2m40s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [2m50s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [3m0s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [3m10s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [3m20s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [3m30s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [3m40s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [3m50s elapsed]
azurerm_mysql_flexible_server.default: Still creating... [4m0s elapsed]
azurerm_mysql_flexible_server.default: Creation complete after 4m7s [id=/subscriptions/30968ef3-435a-4ee2-8c06-7bb66267b87d/resourceGroups/rg-acervantes/providers/Microsoft.DBforMySQL/flexibleServers/mysql-acervantes-db]
azurerm_mysql_flexible_database.main: Creating...
azurerm_mysql_flexible_database.main: Still creating... [10s elapsed]
azurerm_mysql_flexible_database.main: Creation complete after 19s [id=/subscriptions/30968ef3-435a-4ee2-8c06-7bb66267b87d/resourceGroups/rg-acervantes/providers/Microsoft.DBforMySQL/flexibleServers/mysql-acervantes-db/databases/db-acervantes-app]

Apply complete! Resources: 13 added, 0 changed, 0 destroyed.
alaincervantes@DevOps:~/Terraform/DevOps_Test_Solutions/script-01$
```

6. Resultados

Después de ejecutar los comandos, los siguientes recursos fueron creados:

- **Grupo de Recursos:** rg-acervantes
- **Máquina Virtual:** vm-acervantes-web01
- **Base de Datos MySQL:** mysql-acervantes-db
- **Balanceador de Carga:** lb-acervantes-web
- **Red Virtual y Subred:** vnet-acervantes y subnet-acervantes-app
- **IP Pública:** acervantes-ip-publica

Microsoft Azure

Actualización

Buscar recursos, servicios y documentos (G+)

Copilot

Inicio

Todos los recursos

Directorio predeterminado (alaincervantespgmail.onmicrosoft.com)

Crear

Administrar vista

Actualizar

Exportar a CSV

Abrir consulta

Asignar etiquetas

Eliminar

Filtrar por cualquier cam...

Suscripción es igual a todo

Grupo de recursos es igual a todo

Tipo es igual a todo

Ubicación es igual a todo

Agregar filtro

0 Recursos no seguros

0 Recomendaciones

9 Changed resources

Nombre	Tipo	Grupo de recursos	Ubicación
acervantes-ip-publica	Dirección IP pública	rg-acervantes	Brazil South
acervantes.mysql.database.azure.com	Zona DNS privada	rg-acervantes	Global
lb-acervantes-web	Equilibrador de carga	rg-acervantes	Brazil South
mysql-acervantes-db	Servidor flexible de Azure Database for MySQL	rg-acervantes	Brazil South
NetworkWatcher_brazilsouth	Network Watcher	NetworkWatcherRG	Brazil South
nic-acervantes-web	Interfaz de red	rg-acervantes	Brazil South
vm-acervantes-web01	Máquina virtual	rg-acervantes	Brazil South
vm-acervantes-web01_disk1_3d6882f8cdc3438a98c65d6128433867	Disco	RG-ACERVANTES	Brazil South
vnet-acervantes	Red virtual	rg-acervantes	Brazil South

Grupo de Recursos: rg-acervantes

Microsoft Azure

Actualización

Buscar recursos, servicios y documentos (G+)

Copilot

alaincervantes@gmail.com

Inicio > Grupos de recursos >

rg-acervantes

Directorio predeterminado (alaincervantes@gmail.com)

Crear

No agrupar

Buscar

Crear

Administrar vista

Eliminar grupo de recursos

Actualizar

Exportar a CSV

Abrir consulta

Asignar etiquetas

Mover

Está viendo una nueva versión de la experiencia de exploración. Puede que fallen algunas características. Haga clic aquí para acceder a la experiencia anterior.

Nombre 1

NetworkWatcherRG

rg-acervantes

Información general

Registro de actividad

Control de acceso (IAM)

Etiquetas

Visualizador de recursos

Eventos

Configuración

Administración de costos

Supervisión

Automation

Ayuda

Información esencial

Suscripción (mover) : Azure subscription 1

Implementaciones : 1 Conecta

Id. de suscripción : 30968ef3-435a-4ee2-8c06-7bb66267b87d

Ubicación : Brazil South

Etiquetas (editar) : Agregar etiquetas

Recursos

Recomendaciones

Filtrar por cualquier campo

Tipo es igual a todo

Ubicación es igual a todo

Agregar filtro

Mostrando de 1 a 6 de 6 registros

Mostrar tipos ocultos

Sin agrupar

Vista de lista

Nombre	Tipo	Ubicación
acervantes.mysql.database.azure.com	Zona DNS privada	Global
mysql-acervantes-db	Servidor flexible de Azure Database for MySQL	Brazil South
nic-acervantes-web	Interfaz de red	Brazil South
vm-acervantes-web01	Máquina virtual	Brazil South
vm-acervantes-web01_disk1_8d5d43aedc74aa2bb221c0b0c0fa3b6b	Disco	Brazil South
vnet-acervantes	Red virtual	Brazil South

Showing 1 - 2 of 2. Display count: 10

Anterior

1

Siguiente

Excluir imágenes

Máquina Virtual: vm-acervantes-web01

vm-acervantes-web01

Máquina virtual

Buscar

Conectar

Iniciar

Reiniciar

Detener

Hibernar

Captura

Eliminar

Actualizar

Abrir en dispositivos móviles

Comentarios

Información general

Registro de actividad

Control de acceso (IAM)

Etiquetas

Diagnosticar y solucionar problemas

Conectar

Redes

Configuración

Disponibilidad y escala

Seguridad

Copia de seguridad y recuperación ante desastres

Operaciones

Supervisión

Automation

Ayuda

Información esencial

Grupo de recursos (mover) : rg-acervantes

Estado : En ejecución

Ubicación : Brazil South

Suscripción (mover) : Azure subscription 1

Id. de suscripción : 30968ef3-435a-4ee2-8c06-7bb66267b87d

Etiquetas (editar) : Agregar etiquetas

Sistema operativo : Linux (ubuntu 18.04)

Tamaño : Standard D51 v2 (1 vcpu, 3.5 GiB de memoria)

Dirección IP pública : -

Red virtual/subred : vnet-acervantes/subnet-acervantes-app

Nombre DNS : -

Estado de mantenimiento : -

Hora de creación : 9/10/2024, 13:58 UTC

Propiedades

Supervisión

Funcionalidades (7)

Recomendaciones

Tutoriales

Máquina virtual

Nombre del equipo : vm-acervantes-web01

Sistema operativo : Linux (ubuntu 18.04)

Generación de VM : V1

Arquitectura de VM : x64

Estado del agente : Ready

Versión del agente : 2.11.1.12

Hibernación : Deshabilitado

Grupo host : -

Host : -

Redes

Dirección IP pública : -

Dirección IP pública (IPv6) : -

Dirección IP privada : 10.0.1.4

Dirección IP privada (IPv6) : -

Red virtual/subred : vnet-acervantes/subnet-acervantes-app

Nombre DNS : -

Tamaño

Tamaño : Standard D51 v2

vCPU : 1

Base de Datos MySQL: mysql-acervantes-db

mysql-acervantes-db

Servidor flexible de Azure Database for MySQL

Buscar

ConectarVer lista de procesosEliminarRestablecer contraseñaRestaurarReiniciarDetenerActualizarComentarios

Información general

Registro de actividadControl de acceso (IAM)EtiquetasDiagnosticar y solucionar problemasLearning centerConfiguraciónPower PlatformSeguridad

Información esencial

Suscripción (mover) : Azure subscription 1

Id. de suscripción : 30968ef3-435a-4ee2-8c06-7bb66267b87d

Grupo de recursos (mover) : rg-acervantes

Estado : Disponible

Ubicación : Brazil South

Etiquetas (editar) : Agregar etiquetas

Nombre del servidor : mysql-acervantes-db.mysql.database.azure.com

Nombre de inicio de sesi... : uaxfqhrl

Configuración : Uso general D2ds v4 2 núcleos virtuales 8 GiB de RAM...

Versión de MySQL : 8.0

Zona de disponibilidad : 3

Se creó el : 2024-10-09 13:59:54.2904246 UTC

Introducción

PropiedadesRecomendacionesSupervisiónTutoriales

Vista JSON

Balanceador de Carga: lb-acervantes-web

lb-acervantes-web

Equilibrador de carga

Buscar

MoverEliminarActualizarEnviar comentarios

Información general

Registro de actividadControl de acceso (IAM)EtiquetasDiagnosticar y solucionar problemasConfiguraciónSupervisión

Información esencial

Grupo de recursos (mover) : rg-acervantes

Ubicación : Brazil South

Suscripción (mover) : Azure subscription 1

Id. de suscripción : 30968ef3-435a-4ee2-8c06-7bb66267b87d

SKU : Estándar

Etiquetas (editar) : Agregar etiquetas

Ver más

Grupo de back-end : backend-pool

Regla de equilibrio de ca... : http-lb-rule (Tcp/80)

Sondeo de estado : tcp-probe (Tcp/80)

Reglas NAT : 0 de entrada

Nivel : Regional

Vista JSON

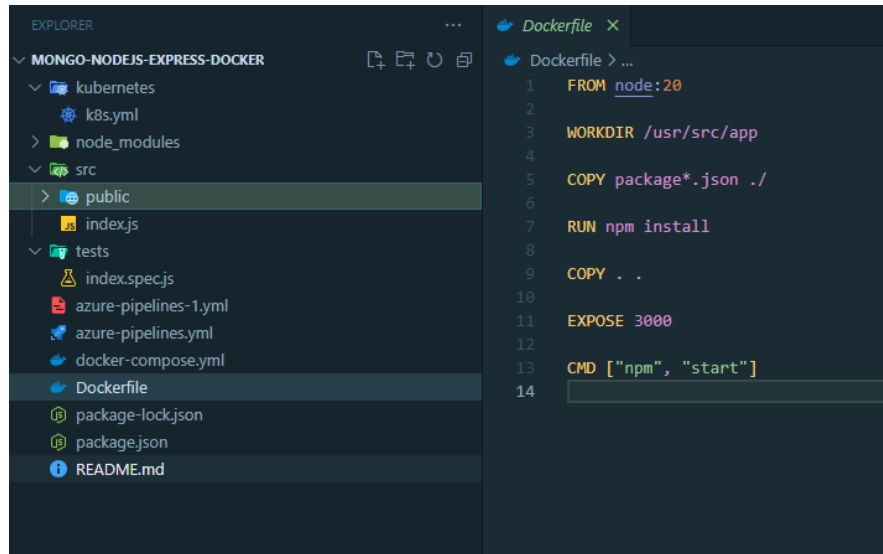
Docker y Contenedores

1. Objetivo

Crear un **Dockerfile** y un archivo **docker-compose.yml** para orquestar una aplicación web y su base de datos en contenedores separados.

Para fines del ejercicio se ha creado una pequeña app utilizando Node.js, Express y MongoDB.

2. Creación de archivo Dockerfile



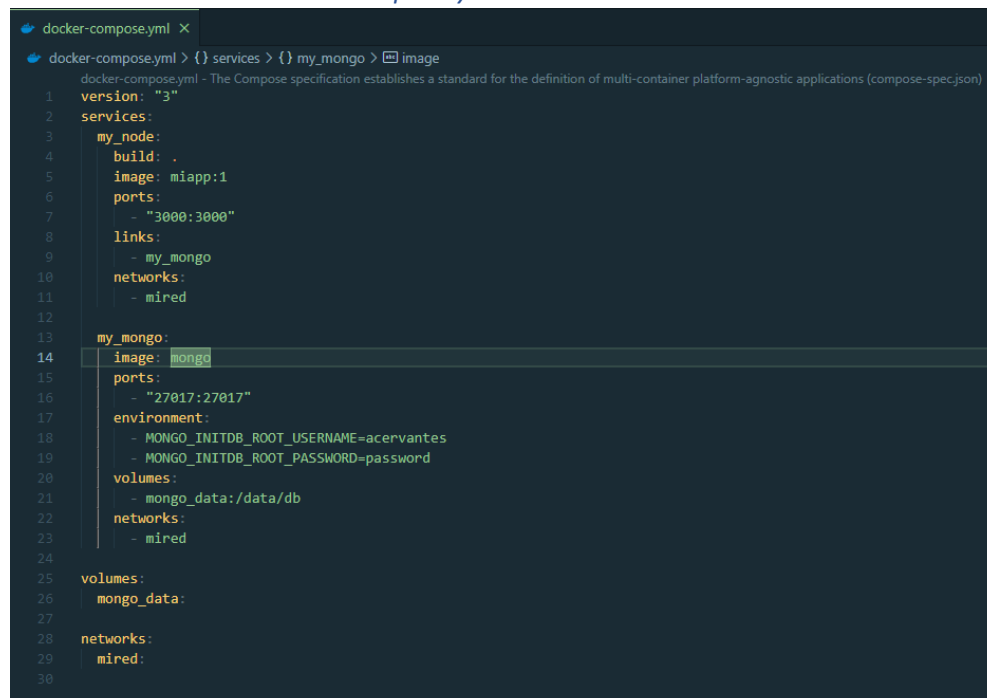
The screenshot shows the VS Code interface. On the left, the Explorer pane displays the project structure for 'MONGO-NODEJS-EXPRESS-DOCKER'. The 'src' directory is expanded, showing 'public' (selected), 'index.js', 'tests', and various configuration files. On the right, the Dockerfile is open, showing the following content:

```

1 FROM node:20
2
3 WORKDIR /usr/src/app
4
5 COPY package*.json ./
6
7 RUN npm install
8
9 COPY . .
10
11 EXPOSE 3000
12
13 CMD ["npm", "start"]
14

```

3. Creación de archivo Docker-compose.yml



The screenshot shows the VS Code interface with the 'docker-compose.yml' file open. The file content is as follows:

```

1 version: "3"
2 services:
3   my_node:
4     build: .
5     image: miapp:1
6     ports:
7       - "3000:3000"
8     links:
9       - my_mongo
10    networks:
11      - mired
12
13   my_mongo:
14     image: mongo
15     ports:
16       - "27017:27017"
17     environment:
18       - MONGO_INITDB_ROOT_USERNAME=acervantes
19       - MONGO_INITDB_ROOT_PASSWORD=password
20     volumes:
21       - mongo_data:/data/db
22     networks:
23       - mired
24
25 volumes:
26   mongo_data:
27
28 networks:
29   mired:
30

```


4. Creación y Ejecución de Contenedores

```
alaincervantes:~/DevOps/repos/Mongo-NodeJs-Express-Docker$ docker compose up -d
[+] Running 1/1
  my_node Warning pull access denied for miapp, repository does not exist or may require 'docker login': denied: requested access to the resource is denied
[+] Building 1.8s (11/11) FINISHED
=> [my_node internal] load build definition from Dockerfile
=> == transferring dockerfile: 172B
=> [my_node internal] load metadata for docker.io/library/node:20
=> [my_node auth] library/node:pull token for registry-1.docker.io
=> [my_node internal] load .dockerignore
=> == transferring context: 2B
=> [my_node 1/5] FROM docker.io/library/node:20@sha256:ffffa89e023a3351904c04284029105d9e2ac7020886d683775a298569591e5bb
=> [my_node internal] load build context
=> == transferring context: 324.72kB
=> CACHED [my_node 2/5] WORKDIR /usr/src/app
=> CACHED [my_node 3/5] COPY package*.json ./
=> CACHED [my_node 4/5] RUN npm install
=> CACHED [my_node 5/5] COPY . .
=> [my_node] exporting to image
=> == exporting layers
=> == writing image sha256:7425f0430808bc3f6e2ff6683a91070fcc831a21922a6aecf6a344a94afc3db6e
=> == naming to docker.io/library/miapp:1
[+] Running 3/3
  ✓ Network mongo-nodejs-express-docker_mired      Created
  ✓ Container mongo-nodejs-express-docker-my_mongo-1 Started
  ✓ Container mongo-nodejs-express-docker-my_node-1 Started
alaincervantes:~/DevOps/repos/Mongo-NodeJs-Express-Docker$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
308e837af446   miapp:1        "docker-entrypoint.s..." 31 seconds ago Up 29 seconds 0.0.0.0:3000->3000/tcp             mongo-nodejs-express-docker-my_node-1
c3709f65c68b   mongo         "docker-entrypoint.s..." 31 seconds ago Up 29 seconds 0.0.0.0:27017->27017/tcp          mongo-nodejs-express-docker-my_mongo-1
alaincervantes:~/DevOps/repos/Mongo-NodeJs-Express-Docker$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
miapp         1         7425f0430808b 43 minutes ago 1.3GB
python3service latest    82bc8e9c26c8  13 days ago   128MB
mongo         latest    72576a3db032  2 weeks ago   855MB
alaincervantes:~/DevOps/repos/Mongo-NodeJs-Express-Docker$
```

Con el comando *docker compose up -d* iniciamos y ejecutamos los servicios definidos en un archivo *docker-compose.yml*

docker ps muestra los contenedores creados y que están actualmente en ejecución.

docker images nos muestra las imágenes de docker creadas

Descripción de la Aplicación

La aplicación implementada es una **API RESTful** en **Node.js** que permite a los usuarios participar en una encuesta sobre sus series favoritas. Los usuarios pueden votar por su serie preferida y proporcionar información adicional, como su nombre, correo electrónico, edad y razones por las que aman la serie, información que se almacena en una base de datos de MongoDB.

Endpoints de la API

La aplicación incluye los siguientes endpoints:

- **GET /encuesta:** Muestra el formulario de la encuesta.
- **POST /encuesta:** Envía las respuestas de la encuesta.

MEJOR SERIE DE LA HISTORIA

Vota por tu serie favorita:

Nombre:

Email:

Edad:

Serie Favorita:

¿La ves en VOS?

☐ No

☒ Sí, no puedo esperar a que la traduzcan

¿Desde dónde ves la serie?

☒ Desde la TV

☒ Desde el ordenador

☒ Desde la tablet

☒ Desde el Celular

¿Por qué amas esa serie?

© Copyright 2024 | Tus Series Favoritas & Co

- **GET /encuestados:** Lista todos los participantes de la encuesta de acuerdo a la información almacenada en el contenedor de MongoDB.

```
{
  "nombre": "Pepito Perez",
  "email": "pepito@gmail.com",
  "edad": "25",
  "serieFavorita": "Game of Thrones",
  "vos": "Sí",
  "dispositivo": ["TV", "Celular"],
  "razonAmor": "Es la mejor serie de la historia."
}
```

Despliegue Continuo con Azure DevOps

1. Objetivo

Configurar un pipeline de Azure DevOps en YAML que realice pruebas automatizadas y despliegue la aplicación en un entorno de prueba.

Solución

azure-pipelines.yml

```
trigger:
  branches:
    include:
      - main

pool:
  name: AgentsDevops
  vmImage: 'ubuntu-latest'

steps:
- checkout: self

- script: |
  docker version
  minikube version
  kubectl version --client
  displayName: 'Check Docker, Minikube and kubectl Versions'

# Iniciar Minikube
- script: |
  minikube start
  displayName: 'Start Minikube'

# Verificar que Minikube esté corriendo
- script: |
  minikube status
  displayName: 'Verify Minikube is running'

# Construir la imagen de Docker
- script: |
  if [[ "$(docker images -q miapp:1 2> /dev/null)" != "" ]]; then
    echo "Image miapp:1 exists. Deleting it..."
    docker rmi -f miapp:1
  else
    echo "Image miapp:1 does not exist. Skipping deletion."
  fi
```

```
    echo "Building new image miapp:1 ..."
    docker build -t miapp:1 .
  displayName: 'Build Docker Image'

# Validar que la imagen de Docker se haya creado correctamente
- script: |
    if [[ "$(docker images -q miapp:1 2> /dev/null)" != "" ]]; then
        echo "Image miapp:1 was created successfully."
    else
        echo "Error: Image miapp:1 was not created."
        exit 1
    fi
  displayName: 'Validate Docker Image Creation'

# Subir la imagen de Docker a Minikube
- script: |
    eval $(minikube docker-env)
    docker build -t miapp:1 .
  displayName: 'Build Docker Image for Minikube'

# Aplicar el archivo de Kubernetes (Deployment y Service) en Minikube
- script: |
    kubectl apply -f kubernetes/k8s.yml
  displayName: 'Deploy to Minikube'

# Verificar el estado del pod y servicio en Minikube
- script: |
    kubectl get pods
    kubectl get svc
  displayName: 'Check Pods and Services'
```

Este pipeline está diseñado para integrarse con un entorno de CI/CD en Azure DevOps y automatiza varias tareas clave, desde verificar las versiones de herramientas, iniciar Minikube, construir imágenes de Docker, desplegar en Kubernetes, y verificar que todo esté funcionando correctamente.

1. Trigger de la Rama Principal

Este pipeline se ejecuta automáticamente cada vez que hay un cambio (commit o pull request) en la rama main del repositorio.

2. Definir el Pool de Agentes

El pipeline se ejecuta en una máquina virtual de Ubuntu. AgentsDevops hace referencia a un pool de agentes predefinidos en Azure DevOps.

3. Clonar el Repositorio

Este paso se encarga de clonar el código fuente del repositorio en el agente de CI para ejecutar los siguientes pasos.

4. Verificación de Versiones

Este script revisa que Docker, Minikube y kubectl estén instalados y funcionando correctamente en el entorno del agente, mostrando las versiones de cada herramienta.

5. Iniciar Minikube

Este paso inicia Minikube, que es el clúster de Kubernetes local utilizado para ejecutar los contenedores.

6. Verificar Estado de Minikube

Este paso verifica si Minikube está corriendo correctamente antes de proceder a los siguientes pasos.

7. Construir la Imagen de Docker

Este paso revisa si ya existe una imagen de Docker llamada miapp:1. Si existe, la elimina, y luego procede a construir una nueva imagen llamada miapp:1 usando el Dockerfile del repositorio.

8. Validar que la Imagen se Haya Creado Correctamente

Una vez creada la imagen, este paso verifica que la imagen miapp:1 realmente exista. Si no, se detiene el pipeline con un mensaje de error.

9. Construir la Imagen para Minikube

Este paso configura el entorno Docker de Minikube para asegurarse de que la imagen miapp:1 esté disponible en Minikube, y luego vuelve a construir la imagen dentro del entorno de Minikube.

10. Desplegar en Minikube con Kubernetes

Aquí, se utiliza kubectl para desplegar los recursos definidos en el archivo k8s.yml en el clúster de Minikube. Este archivo probablemente define los deployments y services necesarios para la aplicación.

11. Verificar el Estado de los Pods y Servicios

Este paso verifica que los pods y servicios de Kubernetes se estén ejecutando correctamente tras el despliegue. Utiliza los comandos `kubectl get pods` y `kubectl get svc` para listar el estado de los pods y servicios.

Pipeline Finalizado

The screenshot displays the Azure DevOps interface for a pipeline named 'Mongo-NodeJs-Express-Docker' in the 'App-Backend' project. The pipeline is in a 'Completed' state, with a build ID of 20241009.3. The left sidebar shows the project structure, including Overview, Boards, Repos, Pipelines, Environments, Library, Test Plans, and Artifacts. The main area shows the pipeline jobs, which are all completed successfully. The 'Post-job: Checkout Mongo-NodeJs-Express-Docker@main to s' job is highlighted, showing its logs.

Job	Duration
Initialize job	<1s
Checkout Mongo-NodeJs-Express...	6s
Check Docker, Minikube and ku...	<1s
Start Minikube	56s
Verify Minikube is running	1s
Build Docker Image	14s
Validate Docker Image Creation	<1s
Build Docker Image for Minikube	19s
Deploy to Minikube	1s
Check Pods and Services	<1s
Post-job: Checkout Mongo-Nod...	<1s
Finalize Job	<1s
Report build status	<1s

```
1 Starting: Checkout Mongo-NodeJs-Express-Docker@main to s
2 =====
3 Task : Get sources
4 Description : Get sources from a repository. Supports Git, TFSVC, and SVN repositories.
5 Version : 1.8.0
6 Author : Microsoft
7 Help : [More Information](https://go.microsoft.com/fwlink/?linkid=798199)
8 =====
9 Cleaning any cached credential from repository: Mongo-NodeJs-Express-Docker (git)
10 Finishing: Checkout Mongo-NodeJs-Express-Docker@main to s
```

Validacion de los Pods y Servicios creados

```

✓ Check Pods and Services

1 Starting: Check Pods and Services
2 =====
3 Task      : Command line
4 Description : Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
5 Version    : 2.244.3
6 Author     : Microsoft Corporation
7 Help      : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/command-line
8 =====
9 Generating script.
10 ===== Starting Command Output =====
11 /usr/bin/bash --noprofile --norc /home/alaincervantes/devopstech-agents/_work/_temp/a1bb65df-d3b9-4f7f-b704-aa31466fd459.sh
12 NAME          READY   STATUS    RESTARTS   AGE
13 my-mongo-6896db4d8-dz7zw    1/1     Running   3 (84s ago)  5h5m
14 my-node-5ff9c5c7bd-zsbrp    1/1     Running   10 (85s ago) 5h5m
15 node-movie-6db5d48bb7-21j4c 1/1     Running   3 (76s ago)  6h3m
16 NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
17 kubernetes     ClusterIP   10.96.0.1    <none>        443/TCP        6h48m
18 my-mongo       ClusterIP   10.96.5.184  <none>        27017/TCP      5h5m
19 my-node        NodePort    10.107.43.123 <none>        3000:30002/TCP 5h5m
20 node-movie     NodePort    10.107.93.186 <none>        3000:30001/TCP 6h15m
21
22 Finishing: Check Pods and Services

```

Repositorio

```

1 trigger:
2   branches:
3     - main
4   - dev
5   pool:
6     name: Agents-DevOps
7     vmImage: 'ubuntu-latest'
8   steps:
9     - checkout: self
10    - docker-compose.yml
11    - Dockerfile
12    - package-lock.json
13    - package.json
14    - README.md
15  # Iniciar Minikube
16  - script: |
17    minikube start
18    minikube status
19    minikube addons enable ingress
20    minikube addons enable dashboard
21    minikube addons enable metrics-server
22    minikube addons enable kubelet
23    minikube addons enable kube-proxy
24    minikube addons enable kube-dns
25    minikube addons enable kube-scheduler
26    minikube addons enable kube-controller-manager
27    minikube addons enable kube-apiserver
28    minikube addons enable kubelet
29    minikube addons enable kube-proxy
30    minikube addons enable kube-dns
31    minikube addons enable kube-scheduler
32    minikube addons enable kube-controller-manager
33    minikube addons enable kube-apiserver
34    minikube addons enable kubelet
35    minikube addons enable kube-proxy
36    minikube addons enable kube-dns
37    minikube addons enable kube-scheduler
38    minikube addons enable kube-controller-manager
39    minikube addons enable kube-apiserver
40    minikube addons enable kubelet
41    minikube addons enable kube-proxy
42    minikube addons enable kube-dns
43    minikube addons enable kube-scheduler
44    minikube addons enable kube-controller-manager
45    minikube addons enable kube-apiserver
46    minikube addons enable kubelet
47    minikube addons enable kube-proxy
48    minikube addons enable kube-dns
49    minikube addons enable kube-scheduler
50    minikube addons enable kube-controller-manager
51    minikube addons enable kube-apiserver
52    minikube addons enable kubelet
53    minikube addons enable kube-proxy
54    minikube addons enable kube-dns
55    minikube addons enable kube-scheduler
56    minikube addons enable kube-controller-manager
57    minikube addons enable kube-apiserver
58    minikube addons enable kubelet
59    minikube addons enable kube-proxy
60    minikube addons enable kube-dns
61    minikube addons enable kube-scheduler
62    minikube addons enable kube-controller-manager
63    minikube addons enable kube-apiserver
64    minikube addons enable kubelet
65    minikube addons enable kube-proxy
66    minikube addons enable kube-dns
67    minikube addons enable kube-scheduler
68    minikube addons enable kube-controller-manager
69    minikube addons enable kube-apiserver
70    minikube addons enable kubelet
71    minikube addons enable kube-proxy
72    minikube addons enable kube-dns
73    minikube addons enable kube-scheduler
74    minikube addons enable kube-controller-manager
75    minikube addons enable kube-apiserver
76    minikube addons enable kubelet
77    minikube addons enable kube-proxy
78    minikube addons enable kube-dns
79    minikube addons enable kube-scheduler
80    minikube addons enable kube-controller-manager
81    minikube addons enable kube-apiserver
82    minikube addons enable kubelet
83    minikube addons enable kube-proxy
84    minikube addons enable kube-dns
85    minikube addons enable kube-scheduler
86    minikube addons enable kube-controller-manager
87    minikube addons enable kube-apiserver
88    minikube addons enable kubelet
89    minikube addons enable kube-proxy
90    minikube addons enable kube-dns
91    minikube addons enable kube-scheduler
92    minikube addons enable kube-controller-manager
93    minikube addons enable kube-apiserver
94    minikube addons enable kubelet
95    minikube addons enable kube-proxy
96    minikube addons enable kube-dns
97    minikube addons enable kube-scheduler
98    minikube addons enable kube-controller-manager
99    minikube addons enable kube-apiserver
100   minikube addons enable kubelet

```