

Plan de despliegue de la aplicación MGM

Introducción

El propósito del presente documento es establecer cómo se llevará a cabo la instalación del sistema MGM propuesta como solución a la Miscelanea Ginna Marcela, en su ámbito de producción final.

Propósito

El software de inventario MGM permitirá a sus usuarios realizar el registro y seguimiento de los procesos de venta y compra de la microempresa beneficiada, con lo que se pretende obtener información precisa de las ganancias y de los gastos, así como el detalle de los productos existentes, los proveedores y las facturas que se generan.

Alcance

Este documento es de interés para el cliente y para los involucrados en el desarrollo, pues se refiere a la primera puesta en marcha de la solución en la nube, y posterior uso a través del navegador web. Para ello, se definen los procedimientos de despliegue con el uso de algunas herramientas.

Planificación del despliegue

Se provee un script declarativo de aprovisionamiento de recursos en la nube con la herramienta Terraform y una serie de instrucciones para configurar ciertos recursos importantes que contienen el código del sistema MGM. El aprovisionamiento se hará usando infraestructura como código.

¿Por qué infraestructura como código?

Infraestructura como código (o IaC) es el proceso de administrar y aprovisionar recursos de manera declarativa, en archivos legibles por máquina, en lugar de configuración de hardware físico o de configuración manual. Se usa para simplificar el despliegue y aprovechar las ventajas que permite:

- reutilización,
- consistencia de la configuración,
- versionamiento de la infraestructura y
- declaración de entornos estables y escalables

Por lo general, el procedimiento de la IaC implica los siguientes 3 pasos:

1. Los desarrolladores definen y escriben las especificaciones de la infraestructura en un lenguaje que es específico del dominio.
2. Los archivos que se crean se envían a una API de gestión, a un servidor de nube o a un repositorio de código.
3. La plataforma toma todas las acciones necesarias para crear y configurar los recursos de computación.

Para el caso que nos ocupa, se describe la infraestructura necesaria para el sistema MGM como código con Terraform en el lenguaje de dominio HCL. Pero necesitamos instalar otras herramientas que nos permitirán integrar el código y los servicios necesarios.

Herramientas de despliegue

Las siguientes herramientas se usarán para el despliegue de los componentes necesarios para el funcionamiento del sistema MGM:

- Cuenta en la nube de Microsoft Azure (Azure para estudiantes)

- La interfaz de la línea de comandos de Azure o Azure CLI
- Terraform como herramienta de infraestructura como código
- Azure portal para detalles de configuración
- Git
- Docker

Algunas de estas herramientas son necesarias para apoyar la ejecución del despliegue (como Docker), como se verá más adelante. Por lo demás, Git, Docker, Azure CLI y Terraform son herramientas que deben estar instaladas en el equipo desde donde se lanzará el despliegue, por lo que se menciona a continuación su instalación (también puede optar por ejecutar el script `install-tools` en la terminal de Powershell, asegúrese que tiene permisos de ejecución), específicamente para el sistema operativo Windows 10 o Windows 11 (antes de iniciar a instalar las herramientas necesarias, asegúrese de que su equipo tiene conexión a internet):

Instalar Git

Vaya a la página de git, haga clic en 64-bit y espere a que complete la descarga. Al finalizar esta, ejecute el archivo `.exe` y dé clic en *siguiente* hasta finalizar la instalación.

Instalar Docker

Para instalar Docker en Windows, es necesario tener WSL (Windows Subsystem for Linux) instalado y habilitado. Esto le permitirá tener una distribución con Linux en su equipo Windows.

Para instalar WSL, vaya al menú de inicio de Windows, y escriba **características de windows**, le aparecerá una opción llamada *Activar o desactivar las características de Windows*. De clic en *Abrir* y le saldrá una ventana emergente en donde podrá habilitar o deshabilitar algunas características de su sistema operativo.

En este caso solo necesitamos habilitar dos características: Plataforma de máquina virtual y Subsistema de Windows para Linux. Asegúrese de marcar estas dos características y dar clic en *Aceptar* (tenga en cuenta que el nombre de estas características pueden variar si el idioma de su sistema operativo está en inglés). Debe esperar a que termine de buscar los archivos necesarios para habilitar esta característica y aplique los cambios. Al finalizar podrá dar clic en *Reiniciar ahora*.

Al volver del reinicio del equipo, vaya a la página de Docker, haga clic en *Docker Desktop for Windows*, y, al finalizar la descarga, ejecute el archivo `.exe` que se descarga. En la ventana del instalador, dé clic en Ok, y espere a que finalice. Finalmente le pedirá que cierre su sesión de Windows, dé clic en *Close and logout*. Cuando vuelva a iniciar su sesión de Windows, le pedirá que acepte los términos y condiciones de uso, acepte para que se complete el proceso de inicio de Docker Desktop. Al final de esto, puede dar clic en *Skip tutorial*.

Instalar Azure CLI

Vaya a la página de Azure CLI y haga clic en el botón *Versión más reciente de la CLI de Azure (64 bits)*. Siga los pasos de instalación.

Instalar Terraform

Vaya a la página de Terraform y, en el apartado de Windows, haga clic en el botón de descarga de *AMD64*. Siga los pasos de instalación. Deberá descomprimir el archivo que se descarga siguiendo las instrucciones a continuación:

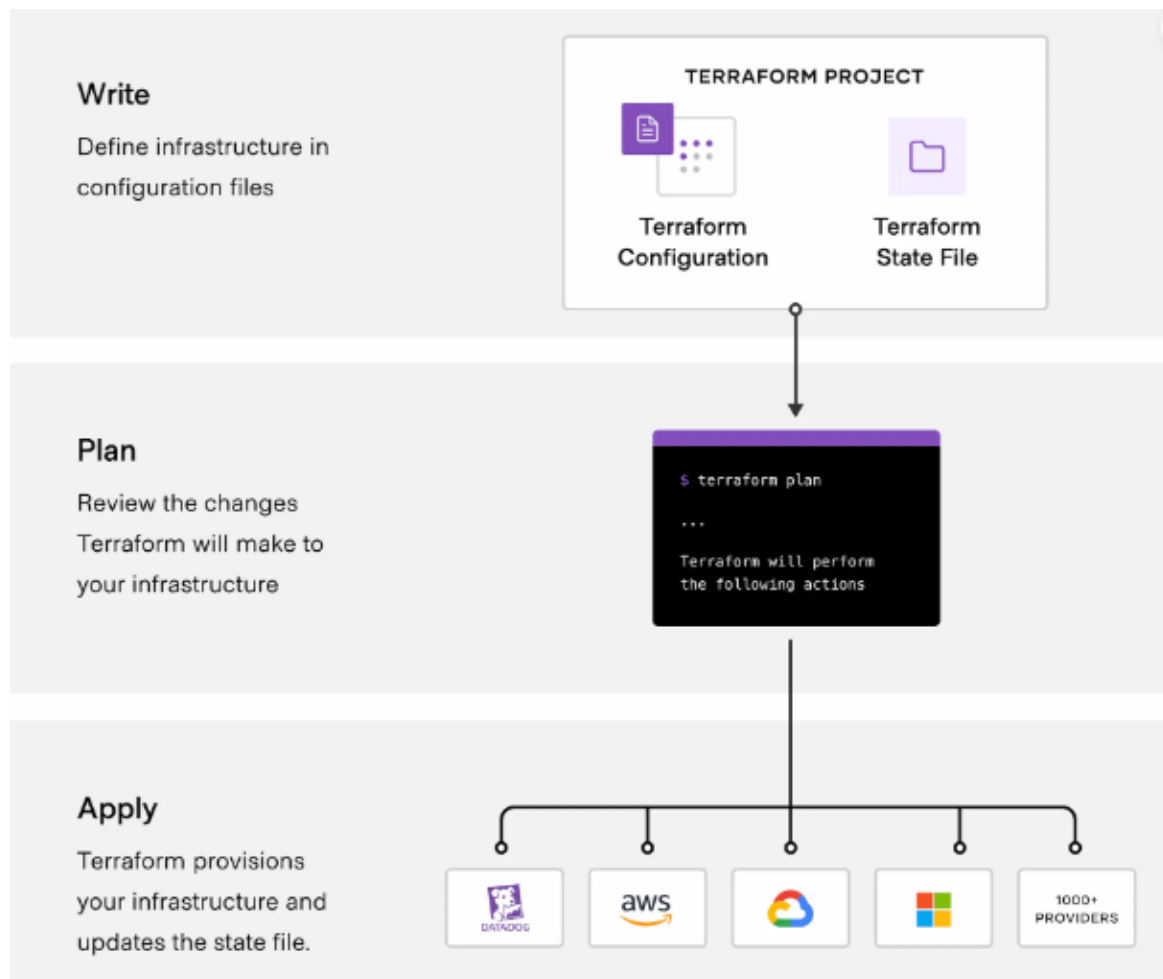
1. Cree una carpeta en `C:/` llamada `terraform`
2. Copie el archivo `.zip` que descargó de terraform y extraiga su contenido en la carpeta creada en el paso anterior.

3. Vaya al menú de inicio de Windows, y escriba **variables de entorno**
4. En la ventana que sale, dé clic en *Variables de entorno*
5. En la sección llamada Variables de sistema, dé clic en Nuevo, agregue el nombre de **terraform** y el valor, la ruta **C:/terraform** (anteriormente creada). Finalmente dé clic en Aceptar
6. En la lista de variables de entorno de sistema, busque la variable Path y dé clic en Editar. En la ventana que surge, dé clic en Nuevo, y pegue nuevamente la ruta **C:/terraform**.
7. Ahora solo haga clic en Aceptar en todas las ventanas que quedan.

Este paso es fundamental para usar IaC y poder desplegar los recursos que necesita el sistema MGM.

Descripción de los recursos definidos con Terraform

Terraform permite maniobrar sobre los proveedores de nube, interactuar con APIs y realizar otras tareas por medio de plugins llamados **providers** o proveedores.



Obtenido de terraform.hashicorp.com Fuente

Tanto los providers como los recursos están definidos en su respectivo archivo de terraform (extensión **.tf**). Los providers necesarios para el despliegue del sistema MGM son:

Providers

Se definen dos providers: **azurerm** para interactuar con la cuenta de nube de Azure y **atlas** para poder correr el script de creación de base de datos dentro del recurso de base de datos de MySQL (esta base de datos se utiliza para normalizar el esquema antes de planificar las migraciones y para simular los cambios a fin de garantizar su aplicabilidad antes de la ejecución).

Nombre	Versión
atlas	0.5.8
azurerm	3.79.0

Recursos

Cada proveedor agrega un conjunto de tipos de recursos y/o fuentes de datos que Terraform puede gestionar. Cada tipo de recurso es implementado por un proveedor; sin proveedores, Terraform no puede gestionar ningún tipo de infraestructura.

Los siguientes son los recursos necesarios para el sistema MGM:

Nombre	Tipo
atlas_schema.mgm-db-schema	esquema de base de datos
azurerm_app_service_connection.mgm-inventory-to-mysql-sc	recurso de conexión para la comunicación entre web app y base de datos
azurerm_linux_web_app.mgm-inventory	recurso web app para servicio de inventario
azurerm_linux_web_app.mgm-users	recurso web app para servicio de usuarios
azurerm_mysql_flexible_database.mgm-db	recurso de base de datos
azurerm_mysql_flexible_server.mgm-db-server	recurso de servidor de base de datos
azurerm_resource_group.mgm-rg	recurso de grupo de recursos en Azure
azurerm_service_plan.mgm-sp	recurso de plan de servicio para web apps
atlas_schema.mgm-db-schema	recurso gestor de esquema de base de datos como código

Inputs

Datos necesarios para la creación de algunos recursos.

Nombre	Descripción	Tipo	Default	Requerido
db_-password	Clave de administrador de base de datos	string	n/a	sí
db_-username	Nombre de usuario administrador de base de datos	string	n/a	sí
prefix	Prefijo del grupo de recursos que es único en la suscripción de Azure	string	"sistema-mgm"	no
resource_-group_-location	Localización (región) para el grupo de recursos	string	"southcentralus"	no

Nombre	Descripción	Tipo	Default	Requerido
jwt_- recovery	Token de recuperación de credenciales	string	n/a	no

Procedimiento de despliegue

Una vez cuente con las herramientas de despliegue instaladas en su pc siga los siguientes pasos:

1. Cree una carpeta
2. Dentro de la carpeta, de clic derecho y abra Git bash (instrucciones válidas solo con git bash)
3. Escriba el comando y ejecútelo oprimiendo la tecla *Enter*:
`git clone https://github.com/dzarkV/TPS_FDS_2671339_PMG6.git`
4. Escriba el comando: `cd TPS_FDS_2671339_PMG6/terraform`
5. Escriba el siguiente comando, reemplazando los valores entre comillas si así lo desea (estos valores serán las credenciales de la base de datos):

```
cat << EOF >> env_dev.auto.tfvars
db_username = "sup3rus3rdatas3"
db_password = "superp4sswordd4t4b4se"
jwt_recovery = "
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI2NDgzYjRlMTM2YmMzY2VjMWFjZWE3MDQiLCJuYW11IjoiaRGluzW5zPyR5GkL4G7D02P01ME4hQjd21ppPtFVU"
EOF
```

6. Inicie sesión en Azure con Azure CLI, para ello, escriba `az login`. Esto le abrirá un navegador en donde deberá colocar sus credenciales de la cuenta de Azure (si no desea iniciar sesión con el navegador, puede optar por el comando `az login -u usuario -p contraseña`, donde deberá reemplazar las palabras `usuario` y `contraseña` por sus respectivas credenciales).
7. Establezca la cuenta "Azure para estudiantes" como la cuenta por defecto con el siguiente comando `az account set --subscription "Azure para estudiantes"`.
8. Ejecute el comando `terraform init`
9. Ejecute el comando `terraform validate`. Este comando validará que la configuración presente en el directorio actual de ejecución no tenga errores de sintaxis.
10. Ejecute el comando `terraform plan -var-file=env_dev.auto.tfvars`. Esto para ejecutar el plan de ejecución de despliegue con las variables definidas en el archivo creado en el paso 5.
11. Finalmente ejecute el comando `terraform apply -var-file=env_dev.auto.tfvars`

El último paso empezará la ejecución de la implementación de los recursos en la nube. El aprovisionamiento de recursos y su configuración tardará unos minutos y el mensaje esperado es **Apply complete!**.