Guía de acondicionamiento y despliegue de la infraestructura Azure del ETL

Proyecto: Contrato de prestación de servicios n° 202500136 de 2025 celebrado entre la corporación Ruta N - Medellín y Guane Enterprises S.A.S

Fecha de elaboración: 2026-08-12

Tabla de contenido

Parte I - Acondicionamiento de la plataforma 2
Qué se hace en Azure 2
Glosario 3
Requisitos 3
- **Paso 1. Abrir Azure Cloud Shell** 3
- **Paso 2. Obtener el subscription_id y el tenant_id** 4
- **Paso 3. Editar el script setup_infra.ps1** 5
- **Paso 4. Subir el script y ejecutarlo (una sola vez)** 6
- **Paso 5. Verificar que todo quedó listo** 6

Validación por el portal (comprobación visual) 7
- **Roles asignados** 7
- **Proveedores de recursos registrados** 8
- **Almacenamiento del state** 9

Parte II - Despliegue Terraform y configuración en el portal 10
- **Paso 6. Confirmar el puente entre Parte I y Parte II** 10
- **Paso 7. Ubicar el código en la carpeta infra/** 10
- **Paso 8. Preparar backend.hcl (state remoto de la Parte I)** 11
- **Paso 9. Ejecutar el despliegue Terraform** 12
- **Paso 10. Revisar los outputs de Terraform** 13
- **Paso 11. Validar en el portal el shell desplegado** 13
- **Paso 12. Configurar en el portal los valores del Distrito** 17
- **Glosario de la Parte II** 17

Parte I - Acondicionamiento de la plataforma

Toda la preparación se resume en cuatro acciones sencillas:

- **Paso 1:** abrir la consola en línea de Azure (Cloud Shell).
- **Paso 2:** obtener dos identificadores con un comando: el de la suscripción y el del tenant.
- **Paso 3:** editar tres líneas del script «setup_infra.ps1».
- **Paso 4:** subir el script a la consola y ejecutarlo una sola vez.

Al terminar la Parte I, continúe en esta misma guía con la Parte II: despliegue Terraform desde infra/ y configuración en el portal de los valores del Distrito.

Qué se hace en Azure

El script automatiza, de forma segura, todas las tareas de preparación que de otro modo habría que hacer una por una en el portal. En una sola ejecución realiza lo siguiente:

Acción del script	Para qué sirve
Activa la suscripción indicada	Asegura que todo se cree en el entorno correcto (staging o el del cliente).
Asigna permisos a su usuario	Le otorga los roles «Contributor» y «User Access Administrator», necesarios para que Terraform pueda crear recursos y asignar permisos.
Habilita los proveedores de recursos	Enciende los servicios de Azure que el sistema usa (Synapse, Data Factory, PostgreSQL, almacenamiento, caja fuerte y red).
Verifica los nombres globales	Comprueba que los nombres de almacenamiento estén disponibles, para que el despliegue no falle.
Crea el almacenamiento del state	Prepara el lugar seguro donde Terraform guardará su registro interno (el «state»).
(Opcional) Crea el usuario técnico	Si se activa, genera un «Service Principal» para despliegues automatizados.

Nota. El script es idempotente: puede ejecutarse más de una vez sin causar problemas. Si algo falla, corríjalo y vuelva a ejecutarlo.

Nota. El script no crea los recursos del sistema (base de datos, Synapse, etc.). Esos los crea Terraform después, en la guía de despliegue de la infraestructura.

Glosario

Términos que aparecen en esta guía, explicados de forma sencilla:

Término	Qué significa
Suscripción	La 'cuenta' de Azure donde se cobran y agrupan los recursos. Cada entorno usa una.
Tenant	La organización o identidad digital que agrupa a los usuarios y sus permisos.
Azure Cloud Shell	Una consola de comandos incluida en el portal, ya conectada a su cuenta. No hay que instalar nada.
Terraform	La herramienta que crea la infraestructura automáticamente a partir de archivos de texto.
State	El archivo donde Terraform recuerda qué recursos ha creado. Debe guardarse en un lugar seguro.
Rol / permiso	Define qué puede hacer una persona o un programa sobre la suscripción.
setup_infra.ps1	El script de preparación incluido en la carpeta infra/ de dap-pipelines. Es el protagonista de la Parte I.

Requisitos

- Una cuenta de Azure en el entorno correcto (la nuestra para staging; la del cliente para producción).
- Permisos suficientes: el rol «Owner», o bien «Contributor» + «User Access Administrator».
- El archivo «setup_infra.ps1», que está en la carpeta «infra» del repositorio «dap-pipelines».

Paso 1. Abrir Azure Cloud Shell

Azure Cloud Shell es una consola que ya viene dentro del portal, conectada a su cuenta, y que no requiere instalar nada. Es donde editaremos y ejecutaremos el script.

- Entre a portal.azure.com e inicie sesión con la cuenta del entorno correcto.
- En la barra superior del portal, haga clic en el ícono de consola (el símbolo «>_»). También puede entrar directamente a shell.azure.com.
- Cuando pregunte, elija la opción «PowerShell».

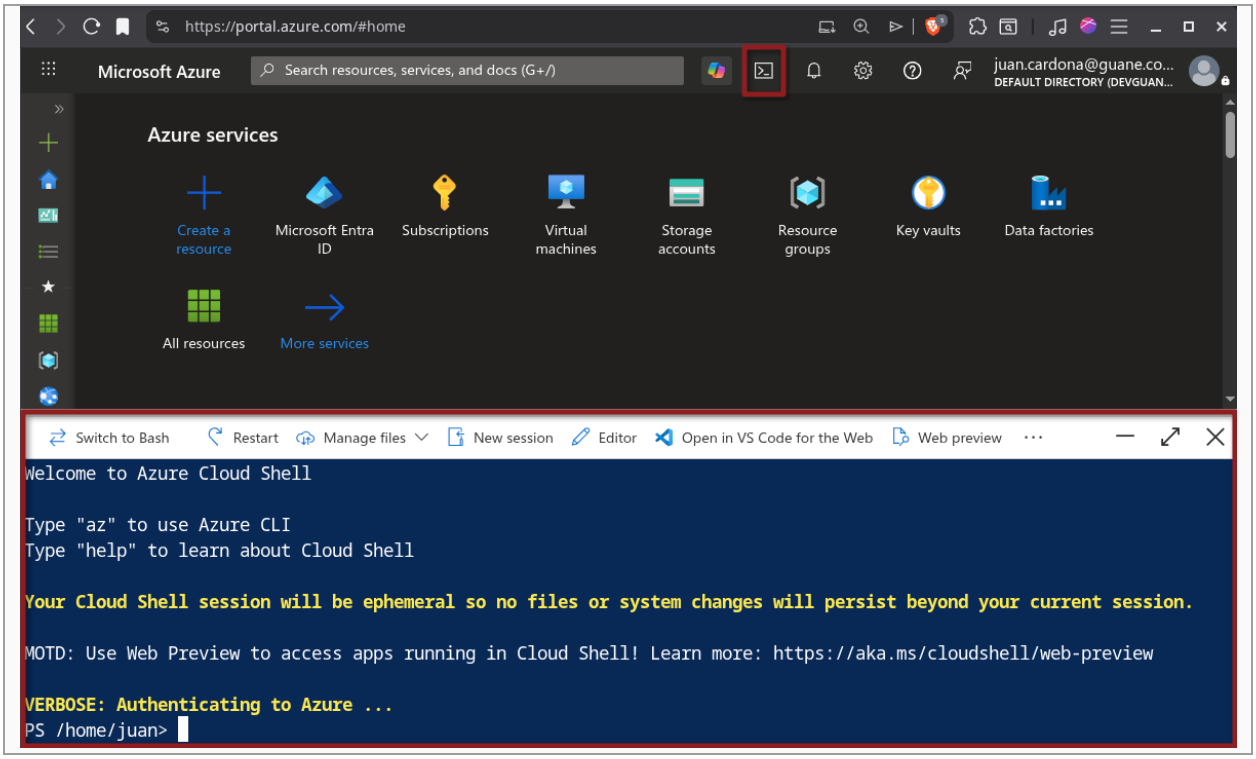


Figura 1. La consola Azure Cloud Shell abierta en modo PowerShell.

Solo en producción. Antes de continuar, confirme dos veces que está en la suscripción del cliente y no en la de staging.

Paso 2. Obtener el subscription_id y el tenant_id

El script necesita el identificador de la suscripción. Además, aprovechamos para obtener el del tenant, que usará más adelante al llenar el archivo de configuración de Terraform. Ejecute este comando en Cloud Shell:

```
az account show --query "{subscription_id:id, tenant_id:tenantId, nombre:name}" -o table
```

El resultado muestra ambos identificadores. Cópielos y guárdelos:

Dato que aparece	Para qué lo usará
subscription_id	Lo pegará en el script (variable $SUB) en el paso 3 y en el archivo terraform.tfvars.
tenant_id	Lo usará más adelante en el archivo terraform.tfvars (guía de despliegue de infraestructura).

Nota. Si administra varias suscripciones, primero seleccione la correcta con: «az account set --subscription <ID>» y luego ejecute el comando anterior.

Paso 3. Editar el script setup_infra.ps1

Abra el archivo «setup_infra.ps1» (de la carpeta «infra» del repositorio «dap-pipelines») con cualquier editor de texto. Solo debe cambiar las líneas de la sección «AJUSTAR» que aparecen al inicio:

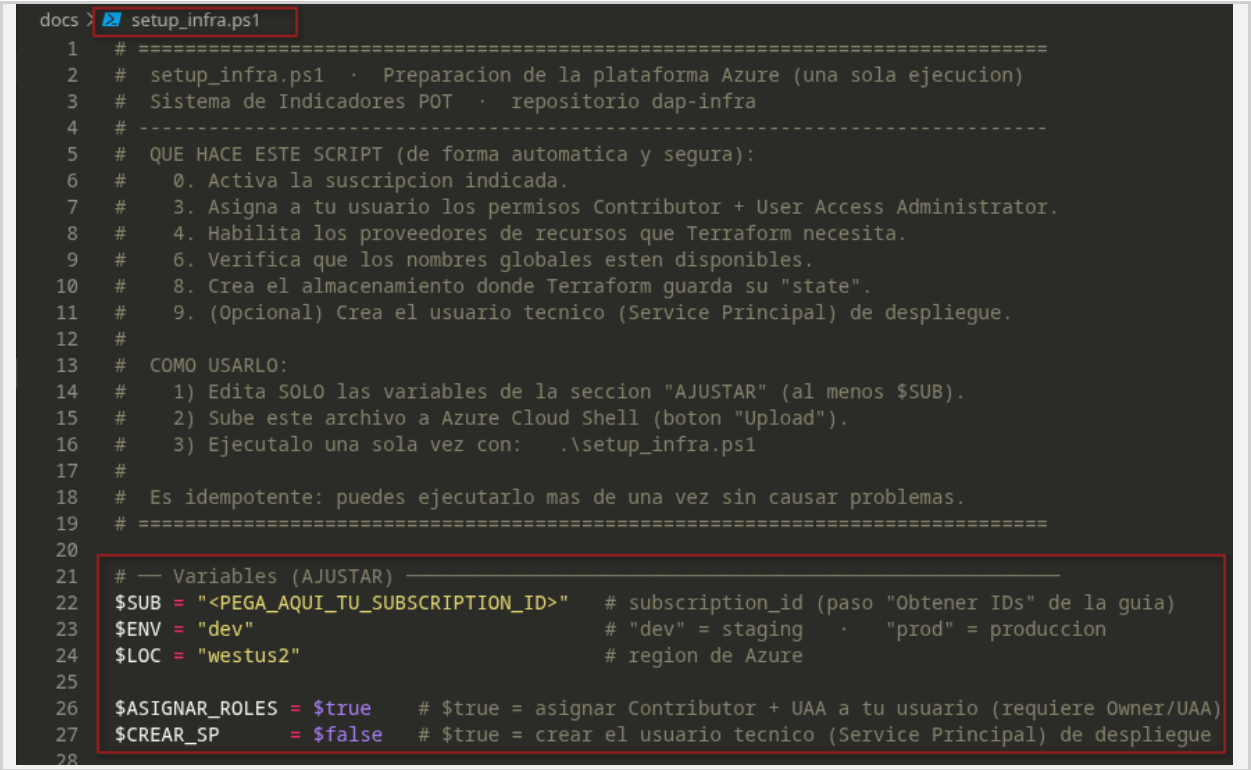


Figura 2. Archivo setup_infra.ps1 en modo edición.

Guía rápida de qué poner en cada línea:

- **$SUB:** el subscription_id que obtuvo en el paso 2. Es lo único obligatorio de cambiar.
- **$ENV:** «dev» si prepara staging; «prod» si prepara producción.
- **$LOC:** la región de Azure (déjela en «westus2» salvo indicación distinta).
- **$ASIGNAR_ROLES:** déjelo en «$true» para que el script le asigne los permisos automáticamente.

$CREAR_SP: déjelo en «$false», salvo que necesite un usuario técnico para automatización.

Guarde el archivo después de editarlo.

Paso 4. Subir el script y ejecutarlo (una sola vez)

Ahora suba el archivo editado a Cloud Shell y ejecútelo. Todo el trabajo de preparación ocurre en esta única ejecución.

Subir el archivo:

- En la barra de herramientas de Cloud Shell, haga clic en el ícono «Upload/Download files» (subir o descargar archivos) y elija «Upload».
- Seleccione el archivo «setup_infra.ps1» que acaba de editar.
- El archivo queda en su carpeta personal de Cloud Shell.

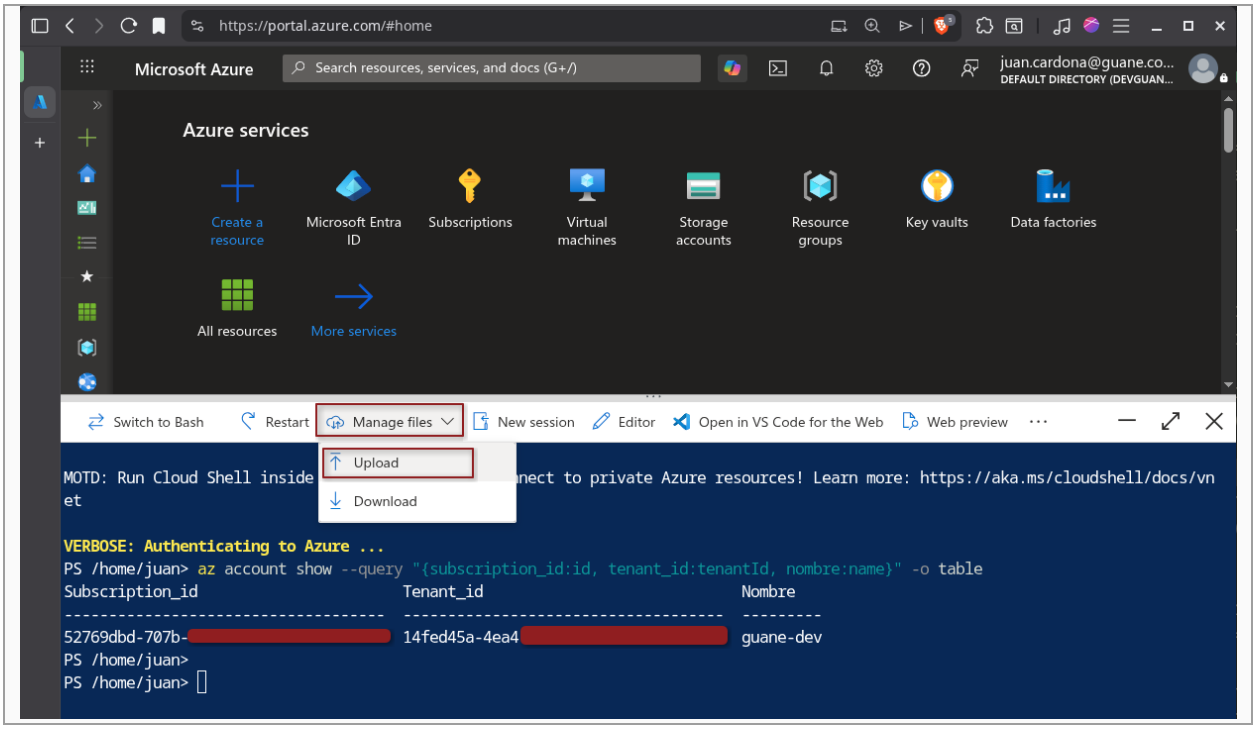


Figura 3. Botón para subir archivos a Cloud Shell.

Ejecutar el script:

```
.\setup_infra.ps1
```

Paso 5. Verificar que todo quedó listo

Al finalizar, el script muestra mensajes de confirmación. Debería ver, entre otros:

- «Suscripción activa: ...» con el nombre correcto del entorno.
- «Roles asignados al usuario actual.»
- «Proveedores de recursos registrándose ...»
- La disponibilidad de los nombres globales (deben aparecer como disponibles).
- «Backend del state listo: rg-tfstate-pot / sttfstatepot{env}001 / tfstate».
- «Acondicionamiento completado. Continúa con la Parte II de esta guía (despliegue Terraform).»

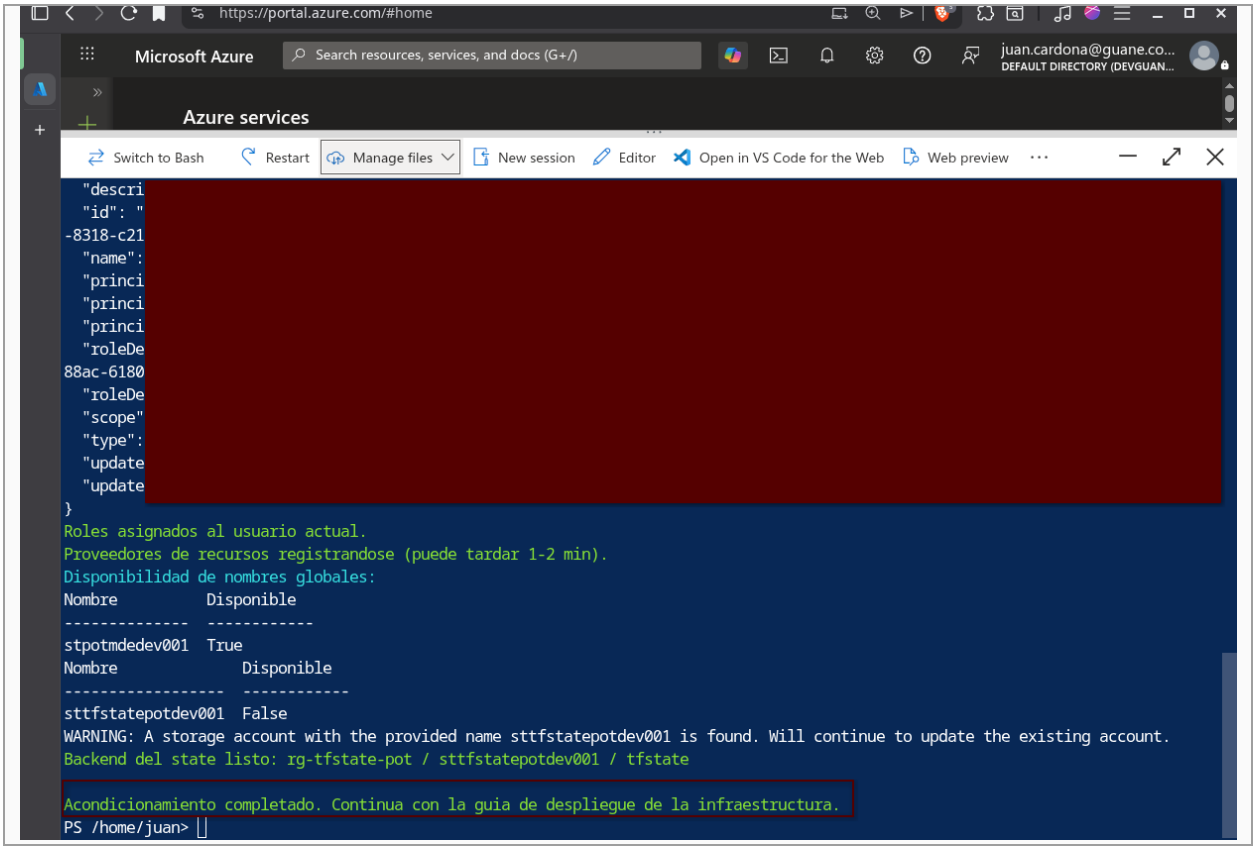


Figura 4. Mensajes de confirmación al final de la ejecución.

Nota. Si algún paso falló, corrija la causa y vuelva a ejecutar el script. Al ser idempotente, puede ejecutarse las veces que necesite.

Validación por el portal (comprobación visual)

Después de ejecutar el script consolidado, confirme visualmente en el portal que cada elemento quedó configurado.

Roles asignados

- Abra «Subscriptions», la suscripción, «Access control (IAM)», pestaña «Role assignments».
- Debe aparecer con dos roles: «Owner» o «Contributor» y «User Access Administrator».

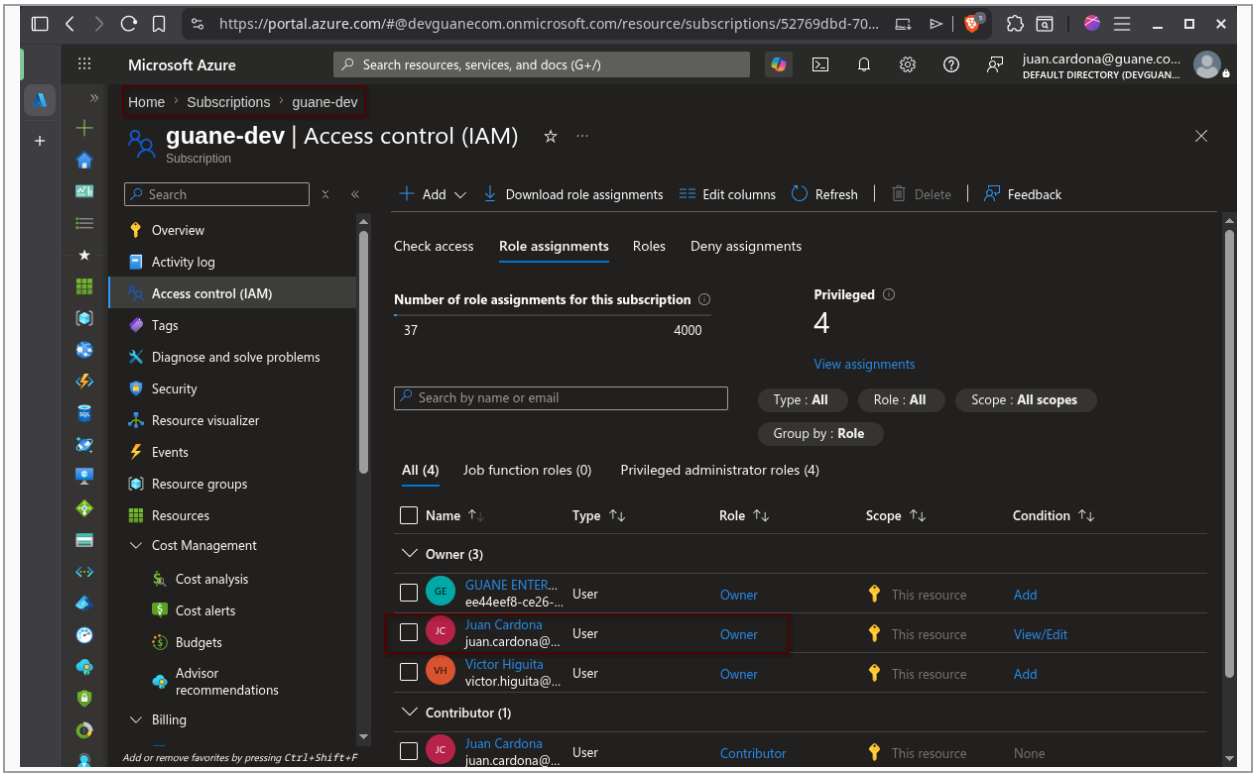


Figura 5. Panel de asignación de roles en la suscripción de Azure para el usuario.

Proveedores de recursos registrados

- Abra «Subscriptions», la suscripción, «Resource providers».
- Verifique que los proveedores ("Microsoft.Synapse", "Microsoft.DataFactory", "Microsoft.DBforPostgreSQL", "Microsoft.Storage", "Microsoft.KeyVault", "Microsoft.Network") estén en estado «Registered». Si alguno sigue en «Registering», espere 1 o 2 minutos y actualiza.

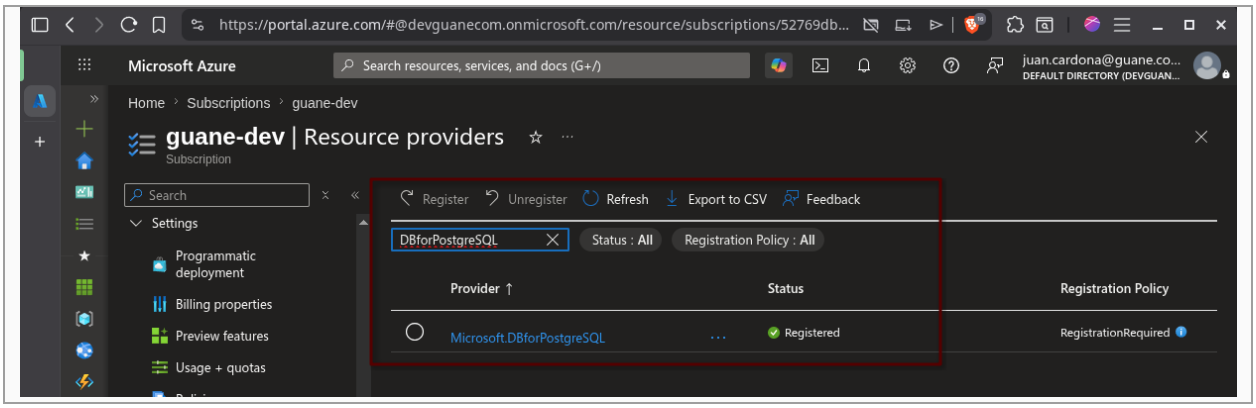


Figura 6. Estado de registro para uno de los 6 proveedores.

Almacenamiento del state

- Abra «Subscriptions», la suscripción, «Resource groups», confirme que existe «rg-tfstate-pot».
- Ábralo y confirme la cuenta de almacenamiento «sttfstatepot{env}001».
- Abra la cuenta, vaya a «Containers» y confirme el contenedor «tfstate».

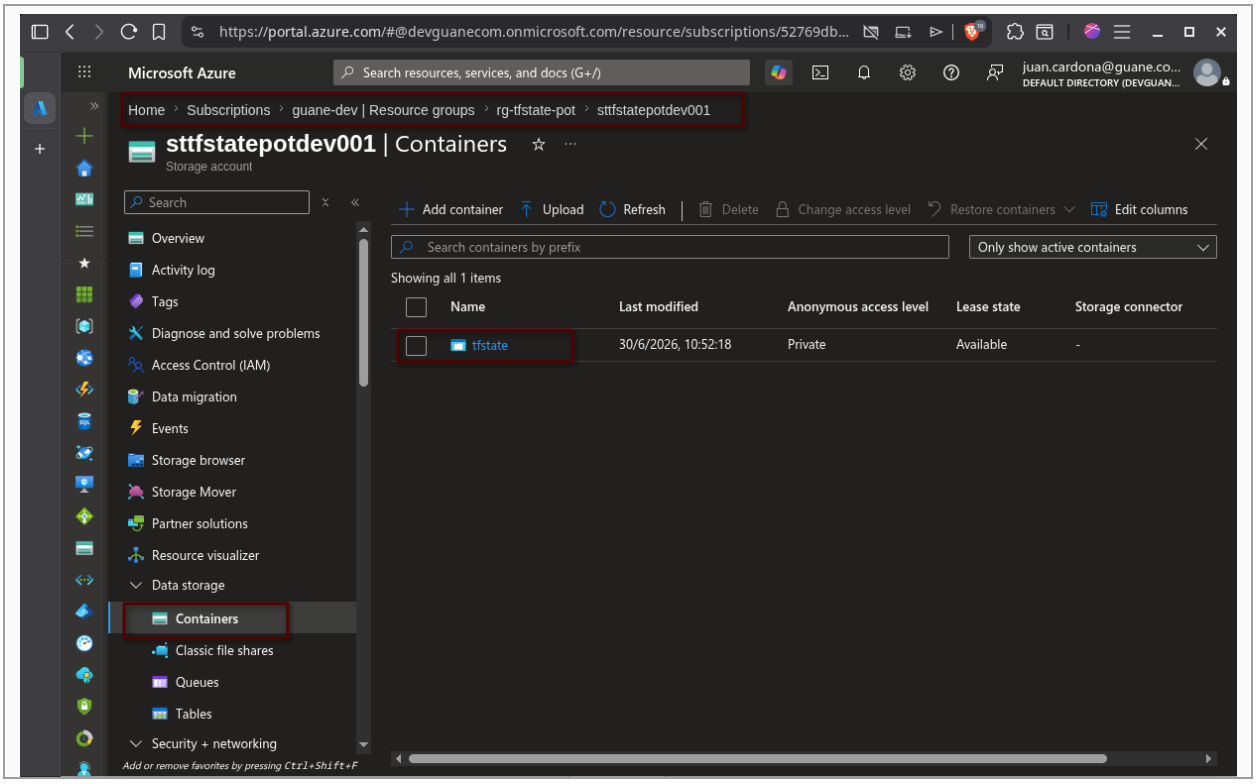


Figura 7. Comprobación de cuenta de almacenamiento para el despliegue.

Parte II - Despliegue Terraform y configuración en el portal

Esta parte comienza cuando la Parte I terminó con el mensaje de acondicionamiento completado. Desde la misma carpeta infra/ (donde está setup_infra.ps1) se despliega el nodo Azure con Terraform y, solo después de validar los recursos, se configuran en el portal Azure los valores del Distrito. No se escriben contraseñas ni hosts on-premise en terraform.tfvars.

Paso 6. Confirmar el puente entre Parte I y Parte II

Antes de ejecutar Terraform, confirme que el acondicionamiento quedó usable:

- Tiene a la mano el subscription_id y el tenant_id del Paso 2.
- En el portal existe «rg-tfstate-pot», la cuenta «sttfstatepot{env}001» y el contenedor «tfstate».
- Los proveedores Synapse, Storage, KeyVault, DBforPostgreSQL y Network están Registered.

Si algo falló, vuelva a ejecutar «.\setup_infra.ps1» en Cloud Shell (es idempotente) y repita la validación de la Parte I.

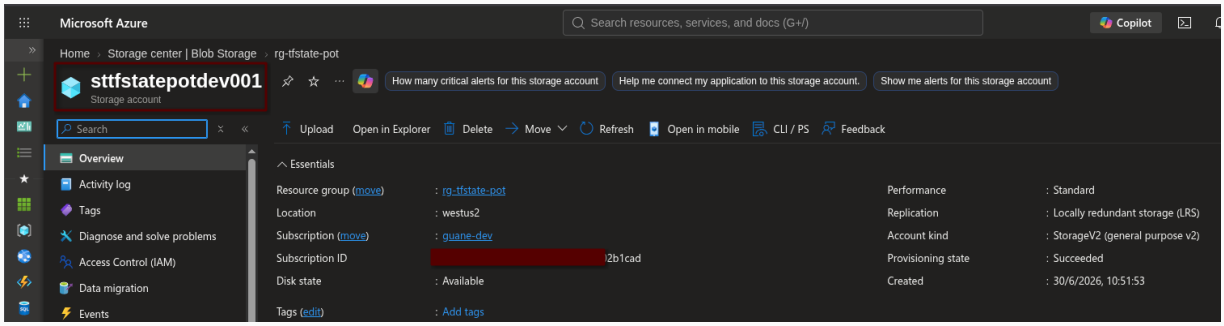


Figura 8. Backend de state rg-tfstate-pot listo tras la Parte I.

Paso 7. Ubicar el código en la carpeta infra/

En el equipo local (Linux, Windows o WSL), clone o actualice dap-pipelines y entre a infra/. Allí están setup_infra.ps1, los módulos Terraform y deploy.sh / deploy.ps1.

```
cd dap-pipelines/infra
```

Debe ver main.tf, variables.tf, outputs.tf, deploy.sh, deploy.ps1, setup_infra.ps1, terraform.tfvars.example y modules/.

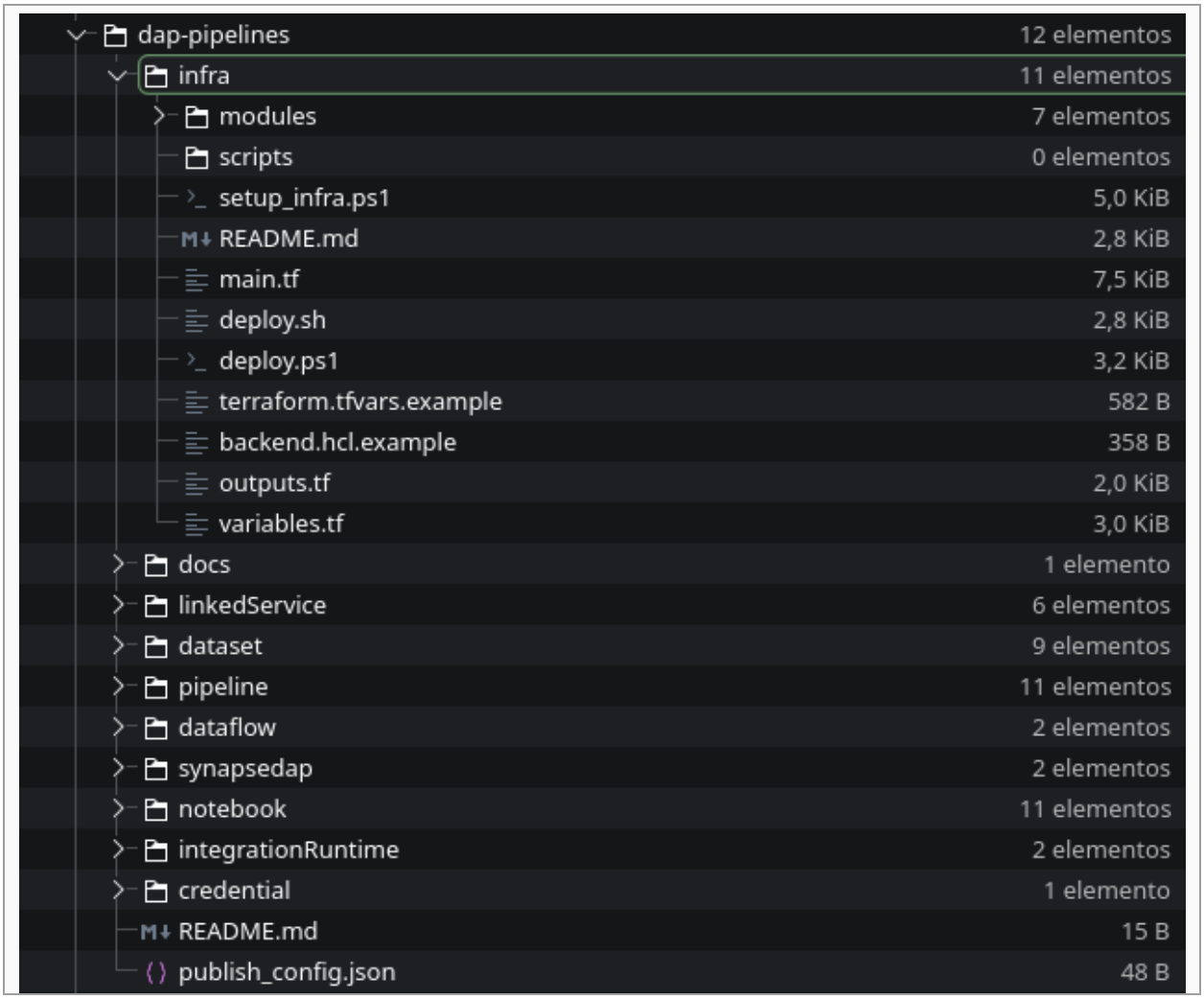


Figura 9. Contenido de la carpeta infra/ en dap-pipelines.

Paso 8. Preparar backend.hcl (state remoto de la Parte I)

La Parte I ya creó el almacenamiento del state. Conéctelo a Terraform:

```
cp backend.hcl.example backend.hcl
```

Edite backend.hcl: resource_group_name = rg-tfstate-pot; storage_account_name = sttfstatepotdev001 (o …prod001 según $ENV de setup_infra.ps1); container_name = tfstate; key = dap-pipelines/infra.tfstate.

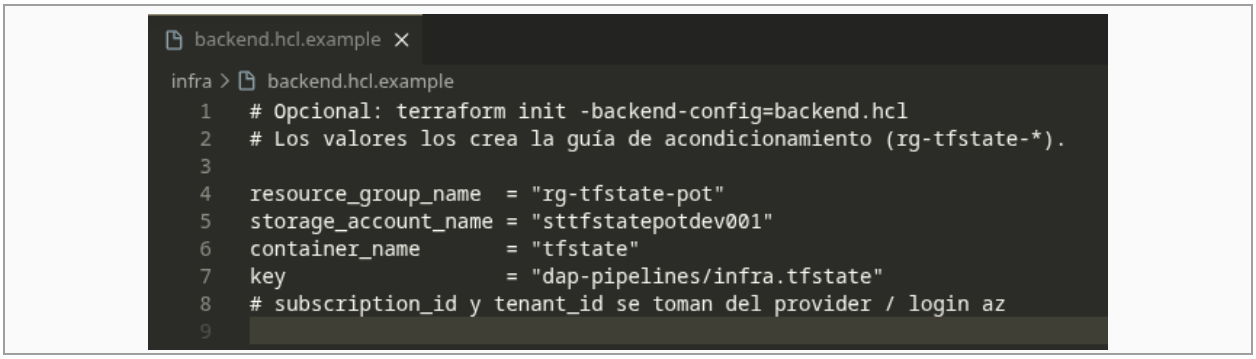


Figura 10. backend.hcl apuntando al storage de la Parte I.

Paso 9. Ejecutar el despliegue Terraform

Requisitos: Terraform ≥ 1.5 y az login en la suscripción de la Parte I. Los scripts detectan subscription_id, tenant_id e IP pública; crean terraform.tfvars si falta (solo plataforma).

Linux, macOS, WSL o Git Bash:

```
./deploy.sh apply
```

Windows PowerShell:

```
.\deploy.ps1 -Action apply
```

Confirme el plan con yes y espere Apply complete. No interrumpa Synapse ni PostgreSQL a mitad de creación.

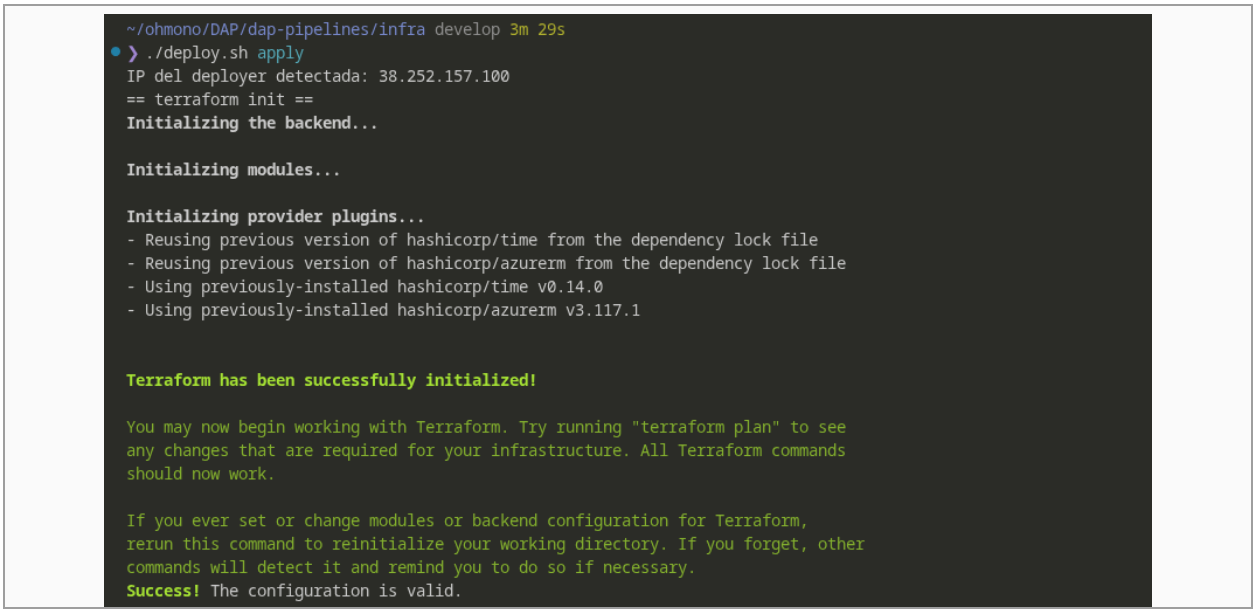


Figura 11. terraform apply completado en consola.

Paso 10. Revisar los outputs de Terraform

Consulte los nombres generados:
terraform output

- Anote resource_group, synapse_workspace_name, key_vault_name, adls_name, postgres_fqdn.
- Revise portal_secrets_to_configure y post_deploy_checklist.

La clave del SHIR es sensible; en el Paso 13 use solo:
terraform output -raw shir_auth_key

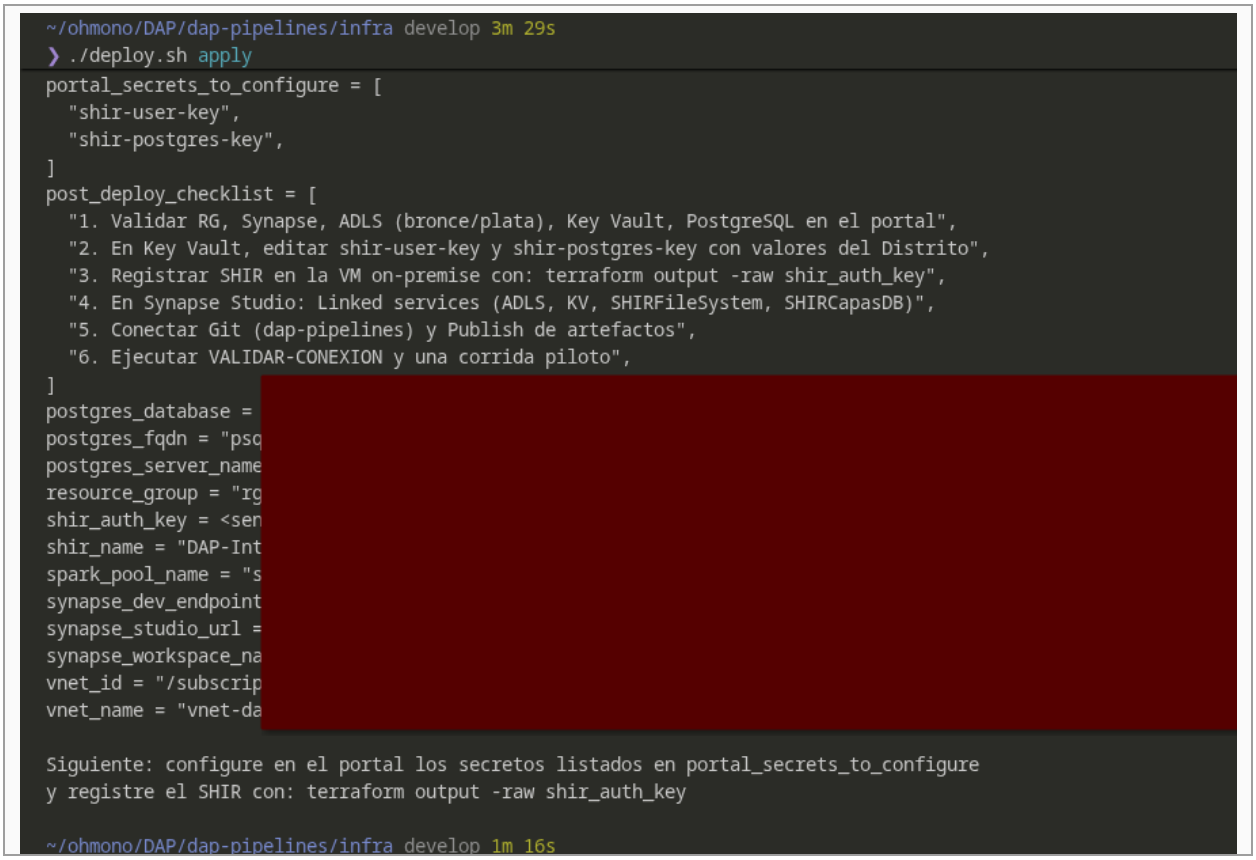


Figura 12. Salida de terraform output (sin exponer shir_auth_key).

Paso 11. Validar en el portal el shell desplegado

Antes de secretos del Distrito, abra el resource group (por defecto rg-dap-mde-dev) y verifique:

- VNet con subnets snet-psql, snet-pe y snet-shir.

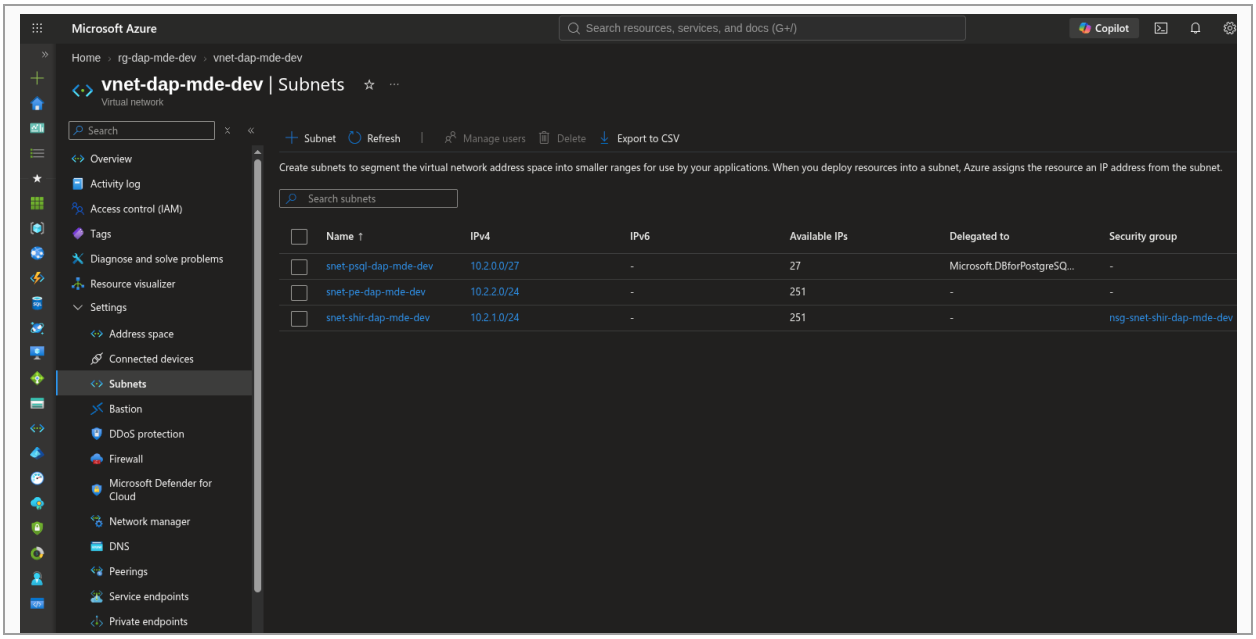


Figura 13. VNet del ETL y subnets.

- Key Vault con secretos listados (marcadores del Distrito aún presentes).

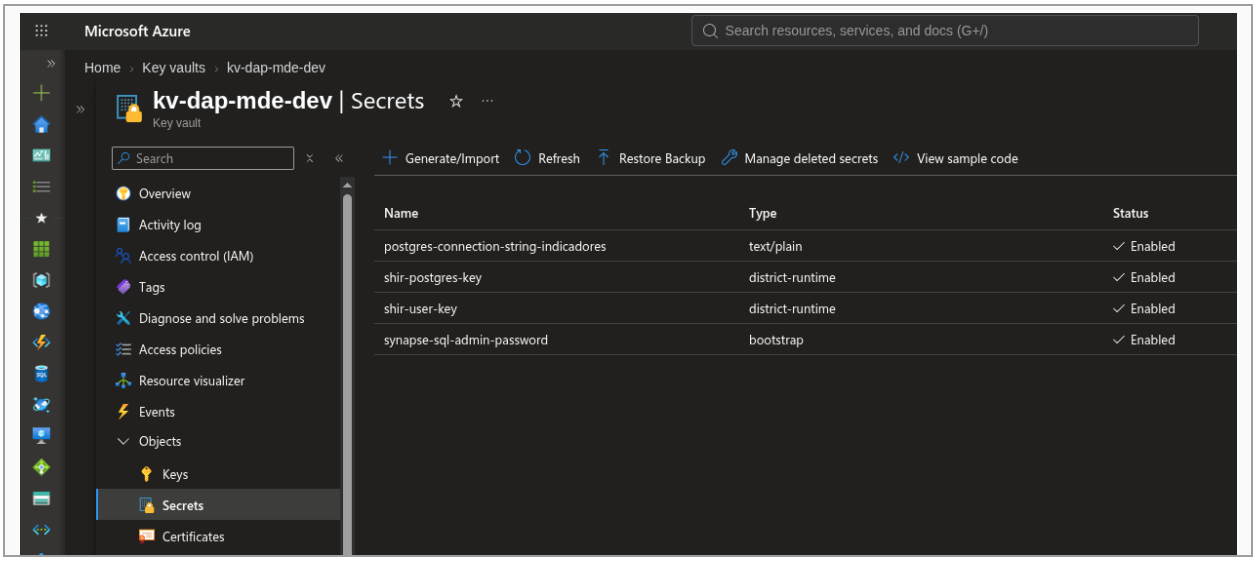


Figura 14. Secrets del Key Vault tras el apply.

- ADLS con filesystems bronce, plata y synapsews.

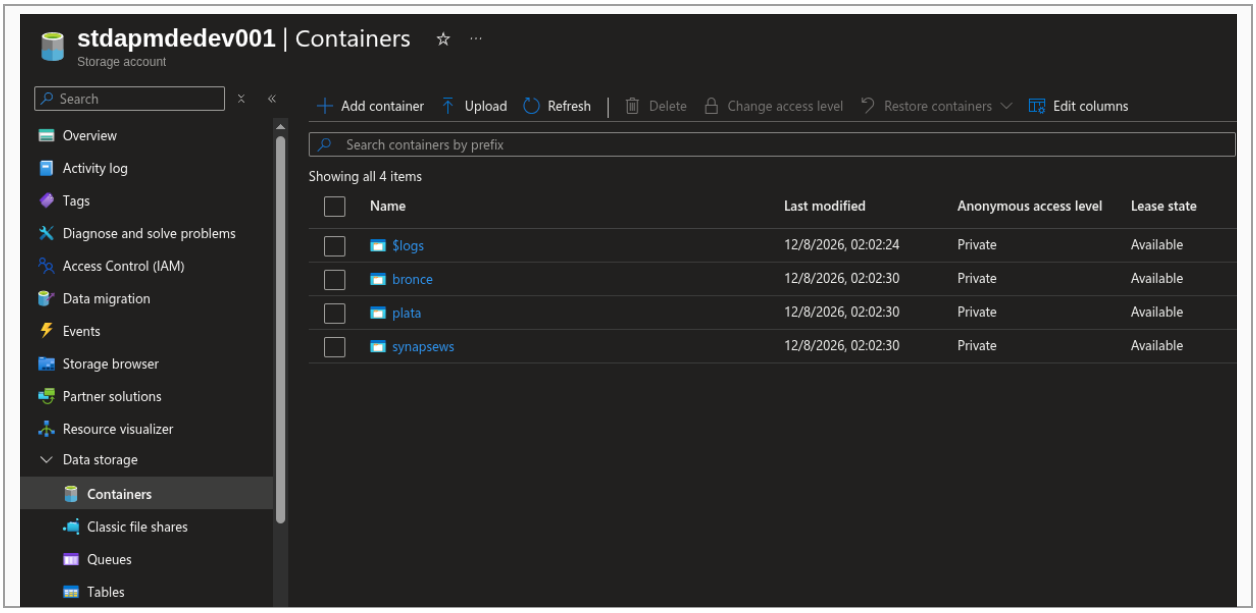


Figura 15. Storage account bronce / plata / synapsews.

- PostgreSQL Flexible Server y base indicadores.

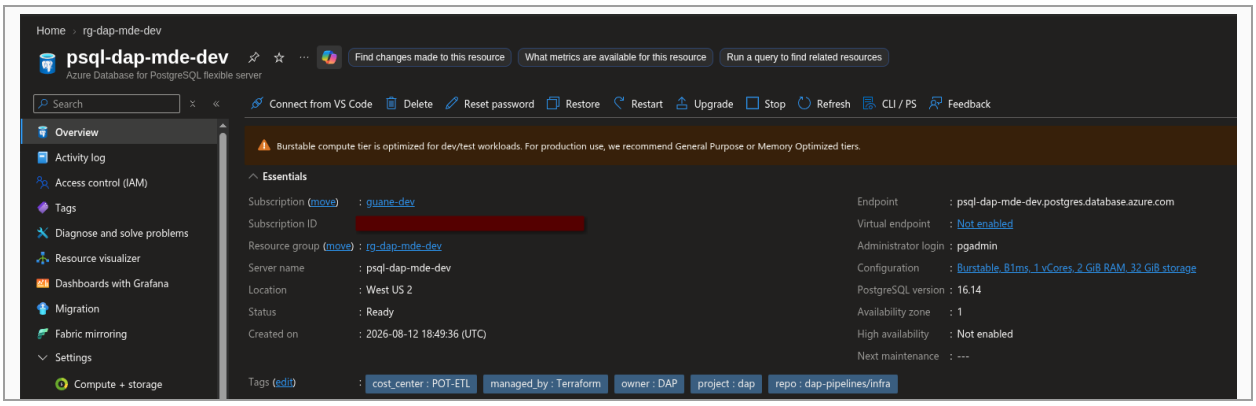


Figura 16. PostgreSQL Flexible Server.

- Synapse: Spark pool e IR DAP-IntegrationRuntime1 pendiente de registro.

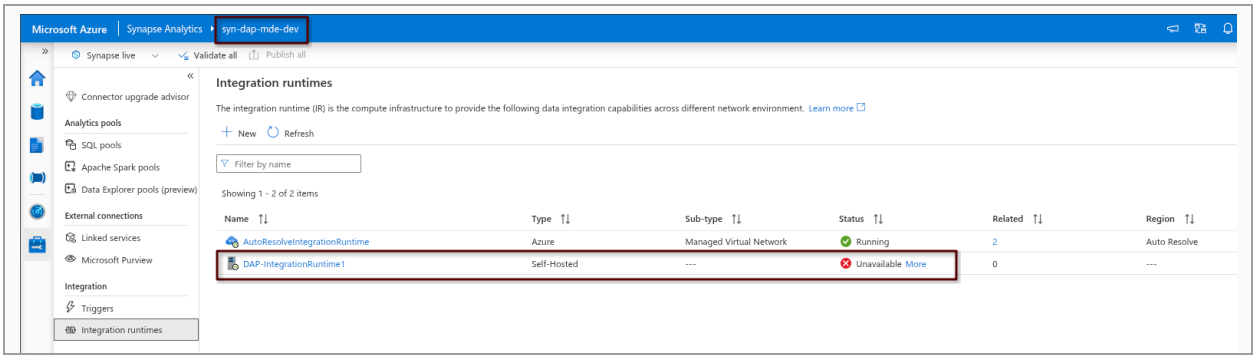


Figura 17. IR DAP-IntegrationRuntime1 sin registrar.

- Managed private endpoints a ADLS, Key Vault y PostgreSQL (aprobar si aplica).

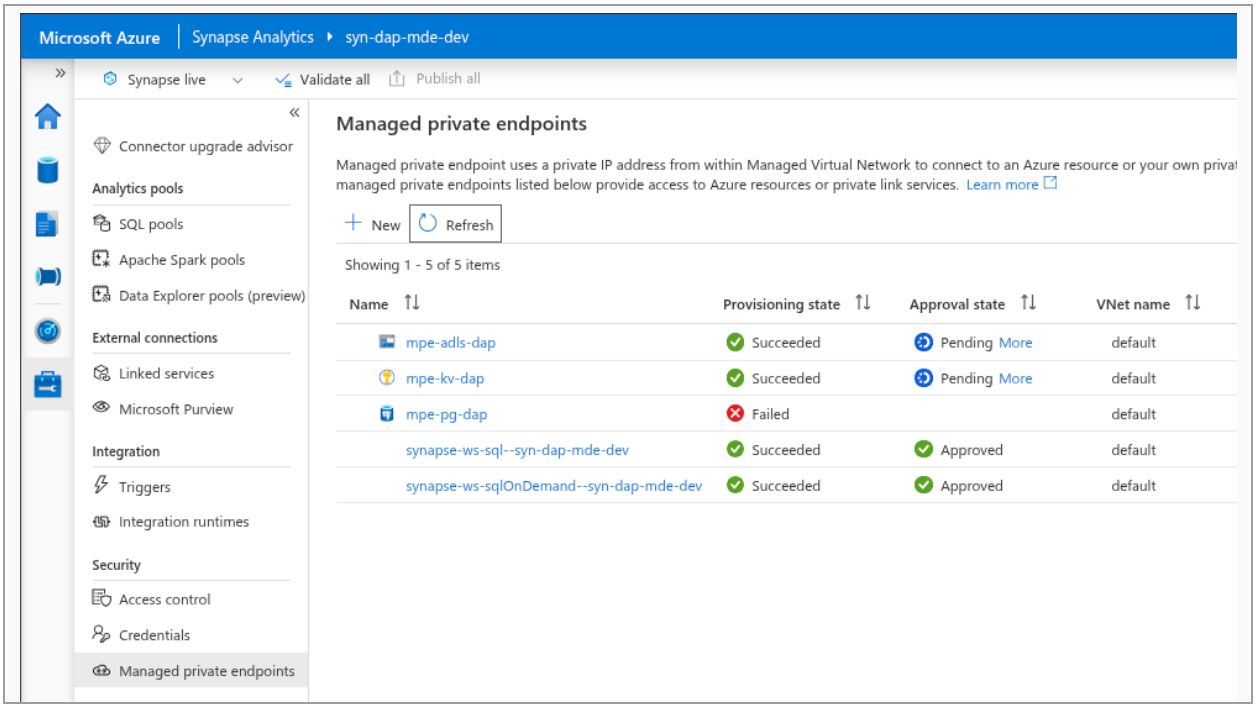


Figura 18. Managed private endpoints Approved.

Paso 12. Configurar en el portal los valores del Distrito

Key Vault → Secrets → secreto → New version → pegue el valor → Create. Un re-apply de Terraform no sobrescribe esos valores.

Secreto	Qué debe pegar	Quién lo provee
shir-user-key	Password de la cuenta de dominio con lectura al share	Operación / AD del Distrito
shir-postgres-key	Cadena de conexión a GeoDB (PostgreSQL/PostGIS)	Administrador GeoDB
postgres-connection-string-indicadores	Bootstrap; rotar solo si cambia el admin Oro	Terraform
synapse-sql-admin-password	Password SQL de arranque del workspace	Terraform

Tabla 1 (Parte II). Secretos: edición en portal vs bootstrap.

Glosario de la Parte II

- infra/: carpeta de dap-pipelines con setup_infra.ps1, Terraform y scripts deploy.
- Shell: recursos Azure del apply, antes de secretos del Distrito.
- Marcador CONFIGURE_IN_AZURE_PORTAL: valor temporal a reemplazar en el Paso 12.
- SHIR: Integration Runtime en la VM on-premise (Paso 13).
- Publish: despliegue de artefactos del repo al entorno vivo de Synapse.