Azure: Configuración de recursos legados.

Un saludo, espero se encuentre bien.

Tiempo de lectura: Cinco minutos (5m).

Descripción.

A continuación detallaremos, desde Infraestructura Digicem - DevOps, la guía de:

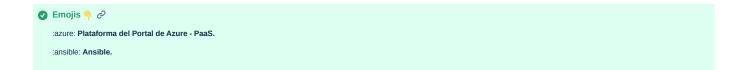
- Automatizar configuraciones, parametrizaciones, de infra legada (Existente) dentro de la nube :azure:.
- · Configuraciones por lotes.

Guía. 🔗

- :ansible:: Herramienta de CaC dentro de los recursos del :azure: portal.
- :azure:: Herramienta de gestión de recursos en la nube, a nivel de laaS, PaaS, SaaS.

Etapas.

- - Tener una :azure: suscripción.
 - Tener un equipo local donde administrar los cambios de archivos ini, cfg, yml, como conectar hacia la infraestructura legada en :azure:, igual al CLI.



Publicaciones. 🔗

• Siéntete libre de hacer uso de las estructuras, manteniendo los cambios posteados.

inventory.ini 🔗

```
1 [Apps]
2 dev-nanaykuna-$RESOURCE
3 sta-nanaykuna-$RESOURCE
4 prd-nanaykuna-$RESOURCE
5
```

Construcción. 🔗

Siéntete libre de hacer uso de las estructuras, manteniendo los cambios posteados.

playbook.yml 🔗

```
---
2 - name: $DESCRIPTION
3 hosts: localhost
4 connection: local
5 tasks:
6 - name: $TASK
7 ansible_azure_module:
8 name: "{{ name }}"
9 location: "eastus"
10 register: "{{ register }}"
11 - debug:
12 var: "{{ var }}"
```

Despliegues. 🔗

• Siéntete libre de hacer uso de las estructuras, manteniendo los cambios posteados.

:azure: CloudShell Sentencias. 🔗

```
1 az ad sp create-for-rbac --name $USER --sdk-auth
2
```

• az : API :azure: Cloud Shell.

• ad : Relaciona la sentencia, como un objeto hacia el directorio activo.

• sp : Servicio principal.

create-for-rbac: Crear un rol basado por control de acceso.
 --name : Asigna el nombre del servicio principal.

• \$USER : Nombre a dar al servicio, ya sea explicito o por variable UNIX.

• --sdk-auth : Output, Formato Clave - Valor (JSON) de cada ID para autorización con SDK, CLI, :azure: y :ansible:, respectivos al usuario creado para el servicio principal.

```
1 export AZURE_CLIENT_ID=""
2 export AZURE_SECRET=""
3 export AZURE_SUBSCRIPTION_ID=""
4 export AZURE_TENANT=""
5
```

• export : Setea, una variable de entorno global, dentro de un sistema operativo Unix.

• AZURE_CLIENT_ID: Variable, declarada según nombre a relacionar.

= : Operador lógico. "" : String.

Sentencias Python. ${\mathscr O}$

```
pip install --upgrade --user ansible[azure]
```

• pip : Packet manager.

• install : Acción.

• --upgrade: Actualizaciones disponibles.

--user : Usuarios.ansible : Paquete.[azure] : Dependencia.

:ansible: Sentencias. $\mathscr O$

```
1 ansible-galaxy collection install azure.azcollection --force
2 ANSIBLE_INVENTORY_ANY_UNPARSED_IS_FAILED=True ansible-playbook -i inventory.ini playbook.yml
3
```

• ansible-galaxy : API del repositorio actual de :ansible: .

collection : Ultimas actualizaciones.
 install : instala actualizaciones.
 azure.azcollection: Colección de paquetes de azure.
 --force : Forzar las actualizaciones.

• ANSIBLE_INVENTORY_ANY_UNPARSED_IS_FAILED: Clave - Valor en True, para parsear por completo el archivo de inventario de recursos a impactar.

• ansible-playbook: API para correr la receta (Cook) de tareas.

-i : Indicar Path del inventario.inventory.ini : Inventario de recursos.

• playbook.yml : Manifiesto de configuraciones y tareas a correr.

Ahora se puede dimensionar el nivel de ejecución, como tiempo a la hora de gestionar este requerimiento.

¡Mil gracias por la atención prestada!

Cualquier duda me puedes contactar...
:WhatsApp: +573058288031
como mi usuario :slack: