

Azure: Cifrado de secretos con SOPS.

Un saludo, espero se encuentre bien.

Tiempo de lectura: 4 minutos.

Descripción.

A continuación detallaremos, desde **Infraestructura Digicem - CyberSecurity**, la guía del paso a paso para:

- Cifrar Secretos de repositorios o estructuras de archivos en :azure: repos desde una llave:key-vault:

Guía. [↗](#)

- :key-vault: : Herramienta de almacenamiento de secretos dentro del :azure: portal.
- :azure:: Herramienta de gestión de recursos en la nube, a nivel de IaaS, PaaS, SaaS.
- :azure: **Repos**: Herramienta de gestión de pipelines en pasos por el branching strategy declarado de git-flow y trunk, entre los entornos: Dev, STA y Prod, algunos HotFixes y Features de recuperación de aplicaciones comprometidas.

Etapas. [↗](#)

i Requerimientos. [↗](#)

- Tener una :azure: suscripción.
- Tener un equipo local donde administrar cambios de repositorios y subirlos hacia los repositorios en :azure: DevOps, sobre Repos.
- Contar con el archivo, repositorio o estructura a cifrar en AES256, bajo un formato JSON.

✓ Emojis [👉](#) [↗](#)

:azure: **Plataforma del Portal de Azure - PaaS.**

:key-vault: **Key-Vault.**

Publicaciones. [↗](#)

- i** • Paquetes de servicios.
- Entregable formalizado cifrado.



sops_3.7.3_amd64.deb
04 abr 2023, 08:25 pm

- i** Se debe clonar el repositorio, respetar la estructura del contenido, para buen efecto al construir la imagen desde Docker.

dockerfile [↗](#)

```
1 FROM ubuntu
2
3 ADD . .
4
5 ENTRYPOINT tar -xvf azSops.tar; bash ./azSops.sh
6
```

- ⚠** • Se comprimen los **archivos sh** como **scripts** en **tar**, para evitar la conversión que hace **Git** sobre archivos **Windows** en **texto plano** de formato **LF (Linux)** a **CRLF** (el cual se corrompe al ser utilizado en **BASH**) que no es legible en **Unix**.

- i** • El siguiente **Script** automatiza la construcción del contenedor dentro del **dockerfile**.

docker-compose.yml [↗](#)

```
1 version: '3'
2
3 services:
4
5   sops:
6     build:
7       context: .
8       dockerfile: ./dockerfile
9     tty: true
10
```

- El siguiente Script automatiza la instalación del azure CLI dentro del contenedor.

deb_install.sh 

```
1 #!/usr/bin/env bash
2
3 #####
4 # This script does three fundamental things: #
5 # 1. Add Microsoft's GPG Key has a trusted source of apt packages. #
6 # 2. Add Microsoft's repositories as a source for apt packages. #
7 # 3. Installs the Azure CLI from those repositories. #
8 # Given the nature of this script, it must be executed with elevated privileges, i.e. with `sudo`. #
9 # #
10 # Remember, with great power comes great responsibility. #
11 # #
12 # Do not be in the habit of executing scripts from the internet with root-level access to your machine. Only trust #
13 # well-known publishers. #
14 #####
15
16 set -e
17
18 if [[ $# -ge 1 && $1 == "-y" ]]; then
19     global_consent=0
20 else
21     global_consent=1
22 fi
23
24 function assert_consent {
25     if [[ $2 -eq 0 ]]; then
26         return 0
27     fi
28
29     echo -n "$1 [Y/n] "
30     read consent
31     if [[ ! "${consent}" == "y" && ! "${consent}" == "Y" && ! "${consent}" == "" ]]; then
32         echo "'${consent}'"
33         exit 1
34     fi
35 }
36
37 global_consent=0 # Artificially giving global consent after review-feedback. Remove this line to enable interactive mode
38
39 setup() {
40
41     assert_consent "Add packages necessary to modify your apt-package sources?" ${global_consent}
42     set -v
43     export DEBIAN_FRONTEND=noninteractive
44     apt-get update
45     apt-get install -y apt-transport-https lsb-release gnupg curl
46     set +v
47
48     assert_consent "Add Microsoft as a trusted package signer?" ${global_consent}
49     set -v
50     curl -sL https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > /etc/apt/trusted.gpg.d/microsoft.gpg
51     set +v
52
53     assert_consent "Add the Azure CLI Repository to your apt sources?" ${global_consent}
54     set -v
55     # Use env var DIST_CODE for the package dist name if provided
56     if [[ -z $DIST_CODE ]]; then
57         CLI_REPO=$(lsb_release -cs)
58         shopt -s nocasematch
59         ERROR_MSG="Unable to find a package for your system. Please check if an existing package in https://packages.microsoft.com/repos/azure-cli/dists/ can be used"
60         if [[ ! $(curl -sL https://packages.microsoft.com/repos/azure-cli/dists/) =~ $CLI_REPO ]]; then
61             DIST=$(lsb_release -is)
62             if [[ $DIST == "Ubuntu" ]]; then
63                 CLI_REPO="jammy"
64             elif [[ $DIST == "Debian" ]]; then
65                 CLI_REPO="bullseye"
66             elif [[ $DIST == "LinuxMint" ]]; then
67                 CLI_REPO=$(cat /etc/os-release | grep -Po 'UBUNTU_CODENAME=K.*') || true
68             if [[ -z $CLI_REPO ]]; then
69                 echo $ERROR_MSG
70                 exit 1
71             fi
72         else
73             echo $ERROR_MSG
74             exit 1
75         fi
76     fi
77 else
```

```

78     CLI_REPO=$DIST_CODE
79     if [[ ! $(curl -sL https://packages.microsoft.com/repos/azure-cli/dists/) =~ $CLI_REPO ]]; then
80         echo "Unable to find an azure-cli package with DIST_CODE=$CLI_REPO in https://packages.microsoft.com/repos/azure-cli/dists/."
81         exit 1
82     fi
83 fi
84 echo "deb [arch=$(dpkg --print-architecture)] https://packages.microsoft.com/repos/azure-cli/ ${CLI_REPO} main" \
85     > /etc/apt/sources.list.d/azure-cli.list
86 apt-get update
87 set +v
88
89 assert_consent "Install the Azure CLI?" ${global_consent}
90 apt-get install -y azure-cli
91
92 }
93
94 setup # ensure the whole file is downloaded before executing
95

```

- Por favor, puedes hacer uso de la siguientes estructuras como bases para lograr el fin solicitado.

sops.yaml [🔗](#)

```

1 creation_rules:
2   - azure_keyvault: https://dev-kv-devops-compañia.vault.azure.net/keys/sops-key/L1av3DeC1fr4d0
3

```

- Se debe establecer el archivo, bajo extensión JSON, debido a la estructura de salida que relaciona SOPS con el cifrado AES256, el cual puede alterar según la extensión la estructura del cifrado cómo data o sin llaves.
- *El siguiente es un ejemplo, no dudes en modificarlo para cifrar o descifrar (sops), solo respeta la extensión JSON.*

env.json [🔗](#)

```

1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "",
5       "Microsoft.AspNetCore": ""
6     }
7   },
8   "Cache": [
9     {
10      "Name": "",
11      "Hours": "",
12      "Minutes": "",
13      "seconds": ""
14    }
15  ],
16 }
17

```

env.enc.dev2 [🔗](#)

```

1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "ENC[AES256_GCM,data:pcOXU6/mRmhi4CY=,iv:W1urE1wsbICW25DKYjHD640coRKzhevvCKeZnVMGBK8=,tag:Gst537NfKyyRIWT3jiBzDA==,type:str]",
5       "Microsoft.AspNetCore": "ENC[AES256_GCM,data:V5W0dx18dw==,iv:SAHY8IHkz4U05MCYT2a3fQgh4+x7zLQ3oWqy4tGJyzE=,tag:DPI5eN8EVxRONTFOwakcow==,type:str]"
6     }
7   },
8   "Cache": [
9     {
10      "Name": "ENC[AES256_GCM,data:stBipS3udA==,iv:Mr8470TFYn0hpQ0m7ZnhyybKxRzpJacbfD80fKwyOyM=,tag:0lsXADxGQx905M6fHFR0tg==,type:str]",
11      "Hours": "ENC[AES256_GCM,data:Bw==,iv:PdlXAyeEr3vSQ36vG0PFrcu0AFHcgGRRqIxvE4/Yi+o=,tag:z0Tp62n0fFDb1GQxMgYTNQ==,type:str]",
12      "Minutes": "ENC[AES256_GCM,data:SQ==,iv:X/6lBr5p0ghVx4614f6Zc7n43jz2yJMo7/301xEoCgw=,tag:g11g0tgd87T2Moe8E033ng==,type:str]",
13      "seconds": "ENC[AES256_GCM,data:Xw==,iv:rV7fH4xyFfyzeRgfCjvAkrX2G0KAx1Rd2t37V74mbDs=,tag:c0iUDjYm1w/Y8ddcv4PbZw==,type:str]"
14    }
15  ],
16   "sops": {
17     "kms": null,
18     "gcp_kms": null,
19     "azure_kv": [
20       {
21         "vault_url": "https://dev-kv-devops-compañia.vault.azure.net",
22         "name": "sops-key",
23         "version": "L1av3DeC1fr4d0",
24         "created_at": "2023-04-04T02:38:51Z",
25         "enc": "S3cR3t0C1fr4d0"
26       }
27     ]
28   }
29 }

```

```

26     }
27     },
28     "hc_vault": null,
29     "age": null,
30     "lastmodified": "2023-04-04T02:38:53Z",
31     "mac": "ENC[AES256_GCM,data:6Qkv30pqpXKNymWbP+arhTMHfx22HRIukBTQdTJB1c1WaQC/KjUp+JX1NR16mPnE0uIBiibwrRIu+1nzQrheGqcQuKN3WzVzVQ0U7B7tBtoaDnHQ999eIiZArf",
32     "pgp": null,
33     "unencrypted_suffix": "_unencrypted",
34     "version": "3.7.3"
35 }
36 }
37

```

• El siguiente Script automatiza el inicio de sesión dentro del portal, desde :azure: CLI dentro del contenedor.

azure.sh

```

1 #!/usr/bin/env bash
2
3 az login --tenant ""
4

```

☐ Corre una tarea interactiva con la imagen creada dentro del contenedor.

```

$ Ejecutando tarea: docker run --rm -it sops:latest

To sign in, use a web browser to open the page https://microsoft.com/devicelogin, and enter the code EG7F4343I to authenticate.

```

Inicio del docker, desde las sentencias declaradas en el Entrypoint.

☐ Autentica dentro de la API :azure: con la llave temporal.



Especificar el código

Escriba el código que se muestra en su aplicación o dispositivo.

EG7F4343I

Siguiente

Autenticación de Azure CLI.

Microsoft Azure



Selección de la cuenta

Está iniciando sesión en **Microsoft Azure Cross-platform Command Line Interface** en otro dispositivo que se encuentra en **Colombia**. Si no es usted, cierre esta página.



Dev Tool Progresol
dev.tools@progresol.com
Conectado

:

Seleccionamos cuenta de acceso.



dev.tools@progresol.com

¿Está intentando iniciar sesión en Microsoft Azure CLI?

Continúe solo si ha descargado la aplicación de una tienda de aplicaciones o un sitio web de confianza.

Cancelar

Continuar

Finalizamos la autenticación.

Construcción. [🔗](#)

- Por favor, puedes hacer uso de la siguiente sentencia para construir tu proyecto.

Comandos Docker. [🔗](#)

```
1 docker compose -f "PATH\docker-compose.yml" up -d --build
```

- **docker:** API.
- **compose:** Orquestador de contenedores.
- **-f:** Establece un argumento para la ruta de un archivo..
- **path:** Ruta relativa del archivo docker compose.
- **docker-compose.yml:** Script de construcción del dockerfile.
- **up:** Levanta el servicio desde la construcción.
- **-d:** activa una sesion detach o permanente, ante cortes de conexión o salidas idle, sin hacer exit o logout.
- **--build:** Construye el proyecto del contenedor como una imagen para docker.

- Por favor, puedes hacer uso de las siguientes sentencias para validar accesos a los recursos y la herramienta.

Comandos Docker Compose. [🔗](#)

```
1 docker run --rm -it name-tag -d
```

- **docker:** API.
- **run:** Corre la imagen desde la API.
- **--rm:** Elimina la imagen actual cacheada o predeterminada.
- **-i:** Inicia una tarea o sesion interactiva.
- **-t:** Etiqueta para la imagen.
- **name-tag:** Nombre de la imagen: versión, como ejemplo **"sops:latest"**.
- **-d:** activa una sesion detach o permanente, ante cortes de conexión o salidas idle, sin hacer exit o logout.

Pruebas. [🔗](#)

- Por favor, puedes hacer uso de las siguientes sentencias para validar accesos a los recursos y la herramienta.

:azure: Comandos. [🔗](#)

```
1 az login --tenant ""
```

- **az** : API.
- **login** : Inicio de sesión sobre la plataforma :azure:, en una ventana aparte bajo un código de validación.
- **--tenant:** Indica, que hará el login, sobre una cuenta tenant :azure: especifica. Ideal si haces multi - tenant.
- **" "** : el ID del tenant de :azure:.

```
1 export AZURE_CLIENT_ID="AzUr3-C1eNt-1D"
```

- **export** : Declara la variable en sistemas operativos Unix igual Linux.
- **AZURE_CLIENT_ID** : Nombre de variable a setear.
- **=** : Operador lógico del valor de la variable.
- **"AzUr3-C1eNt-1D"**: Valor de la variable a retornar, al ser llamada.

```
1 export AZURE_CLIENT_SECRET="AzUr3-C1eNt-5E{r3T";
2 export AZURE_TENANT_ID="AzUr3-T3n4Nt-1D"
```

- **::** Operador de salto de sentencia en línea.

✓ Emojis 🗨️ [🔗](#)

:azure: **Plataforma del Portal de Azure - PaaS.**

Desplegar. [🔗](#)

Comandos Docker. [🔗](#)

```
1 docker run --rm -it name:tag -d
```

- **docker** : API.
- **run** : Corre la imagen desde la API.
- **--rm** : Elimina la imagen actual cacheada o predeterminada.
- **-i** : Inicia una tarea o sesion interactiva.
- **-t** : Etiqueta para la imagen.

- **name:tag**: Nombre de la imagen: versión, como ejemplo **"sops:lastest"**.
- **-d** : activa una sesión detach o permanente, ante cortes de conexión o salidas idle, sin hacer exit o logout.

:key-vault:Comandos para cifrado. [🔗](#)

```
1 sops -e env.json > env.enc.dev2; cat env.enc.dev2
```

- **sops** : Herramienta de cifrado.
- **-e** : Parámetro o argumento de cifrar secretos.
- **env.json** : Archivo o estructura a cifrar.
- **>** : Redireccionamiento de salida según error, de la sentencia anterior.
- **env.enc.dev2**: Documento receptor de salida.
- **cat** : Concatena los valores desde consola sin requerir abrir un binario sobre el archivo.

☐ Copia la salida de la concatenación en Linux con el archivo que contendrá los secretos.

```
GetClientsInformationFunctionKey": "ENC[AES256_GCM,data:c5+EpHlo9u0VVG09c
8AYD+u7tygk4SiiYk27/fot7g=,iv:uogTkdcg6Bd5Qa7FzF51if6mJemAcxkh1Z0KXpXAR/4
pe:str]",
GetProductCategoriesFunctionKey": "ENC[AES256_GCM,data:tvSmoY6ISURIdYFRiv
2hnq5vT6cjNNarSPLee/YtDw=,iv:E0opTwnjSWGhyCD4kp4DHSe26/1S/dwJaGVMwJqWvag=
e:str]",
UpdateProductCategoryFunctionKey": "ENC[AES256_GCM,data:rQ2M4ytr50yEWgUul
meUwDmvKSUHVGMp+7DCu3bQPQ=,iv:/Yqp79ZUzhPb/9EAS7QN88v50yq7RE3xHMcBQiZj5HE
pe:str]"

{
  "Domain": "ENC[AES256_GCM,data:zH+k4jAh7p0AuJvj+Tf7gMmknFANryYmbmKYQKw9qlw
6cGC8f4SHP/Zb2I=,tag:zAZ4fqCLafsvnsKB6l5M6Q==,type:str]",
  "ClientId": "ENC[AES256_GCM,data:EEeAjq53srmFNuFR05kviIdagAsY4Z5e0VcX0lr0l
y0BEx1Gwkc9hbCbAU=,tag:Yyg+jRsmW96G7bi3cBZiXw==,type:str]"

  "kms": null,
  "gcp_kms": null,
  "azure_kv": [
    {
      "vault_url": "https://dev-kv-devops-nanaykuna.vault.azure.com",
      "name": "sops-key",
      "version": "dc7611e069b64c1ea46a9dbb98452d72",
      "lastmodified": "2023-04-26T20:44:00Z",
      "mac": "ENC[AES256_GCM,data:DKKLPUbVNu17e1oLw2EVHaSfMfsh5/DuJyAeCdIIg3VvDt
rmiQJYZoQbLc8Q3yPPJS1lE8ceIiU1VLF28M1lC1nY+5/nRXkszl1gsbjPraJ30bdtvR70ras
NLY70prJUBHhrXvoj1+uaZmhXc5wiamEFs=,tag:IpCUj6V1lWnqTorQf0Tawg==,type:str]
      "pgp": null,
      "unencrypted_suffix": "_unencrypted",
      "version": "3.7.3"
```

Output del binario UNIX "Cat".

✖ Cat como binario en Unix será deprecado, como buena práctica de shell code se debe usar <, sin necesidad del pipeline como tubería de filtrado en salida (|).

:key-vault:Comandos de descifrado. [🔗](#)

```
1 sops --output-type json -d env.enc.dev2
```

- **sops** : Herramienta de cifrado.
- **--output-type**: Predetermina una salida de comando.
- **json** : Formato de archivo a mantener en la salida.
- **-d** : Parámetro o argumento para descifrar secretos.
- **env.enc.dev2** : Archivo o estructura a descifrar.

👉 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: