



OSS in the Cloud – Laboratorio 3

Nombre del Laboratorio:
Containers on Azure

Preparado para:
Microsoft

Fecha:
[02/28/2018]

Versión: **[1.0]**

Preparado por:
Julio Cesar Zepeda Alvarez
oss@bside.com.mx

bSide
Expertos en Soluciones TI, S.C.



Contenido

Metas y objetivos del Laboratorio	3
Alcance del Laboratorio	3
Desarrollo del Laboratorio.....	4
Despliegue de Ubuntu en Windows 10	4
Despliegue de Azure CLI 2.0	5
Despliegue de Docker CE en Ubuntu	7
Contenedores de Docker Hub.....	10
ACR despliegue de imágenes de contenedores	11
AKS Crear clúster	14
Eliminar Ambiente.....	15
Conclusiones	15
Contacto	15



Metas y objetivos del Laboratorio

Esta sección describe las metas y los objetivos que se pretende lograr con este laboratorio.

A continuación, se enlistan las metas

- Microsoft Loves Linux.
- Windows 10 + Linux.
- Conocer la facilidad de despliegue de Servicios en Azure por medio de Azure CLI 2.0.
- Crear instancias de contenedores por medio de Azure Cli
- Creación y uso de ACR con imágenes de contenedores
- Creación y uso de un clúster con AKS
- Conocer la facilidad de despliegue de imágenes de contenedores en un registro privado y dentro de un clúster de Kubernetes.

A continuación, se enlistan los objetivos:

- Que el cliente conozca las bondades y facilidad de despliegue un Docker container en un recurso PaaS como ACR y la flexibilidad de la liberación de imágenes de contenedores en un servicio IaaS como AKS.

Alcance del Laboratorio

Como parte de esta demo solo se cubre:

1. Creación de imágenes de contenedores provenientes de Docker Hub.
2. Creación de recursos de Azure como ACR y AKS desde línea de comandos.
3. Despliegue de contenedores dentro de ACR y AKS.
4. Ejemplo de la creación de ACR y AKS desde el portal de Azure.

Desarrollo del Laboratorio

Despliegue de Ubuntu en Windows 10

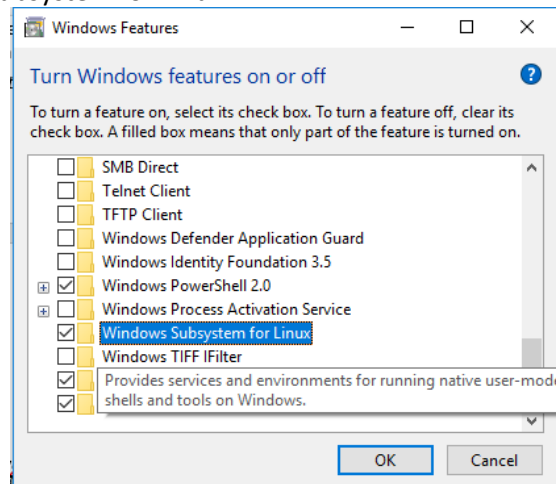
- Script:

En este momento se instalan los **Pre-requisitos** para poder ejecutar el script de despliegue de elementos de Azure por medio de Azure Cli 2.0.

La primera aportación (2009) de Microsoft al mundo Open Source Software fue la contribución de 20,000 líneas de código al Kernel de Linux.

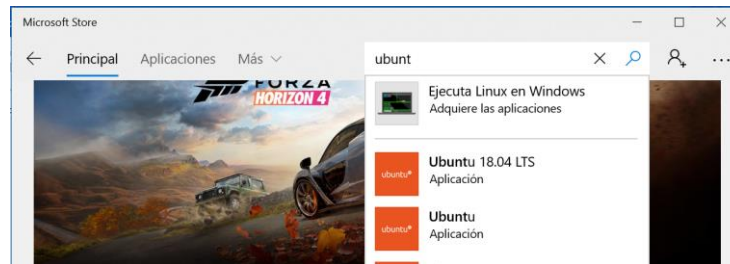
En esta sección se muestra la facilidad de instalar un ambiente Linux dentro de Windows 10 para poder gestionar centralmente dos grandes mundos Windows / Linux.

1. Ingresar a Control Panel -> Programs -> Programs and Features -> Turn Windows features on or off.
2. Habilitar "Windows Subsystem for Linux".



4

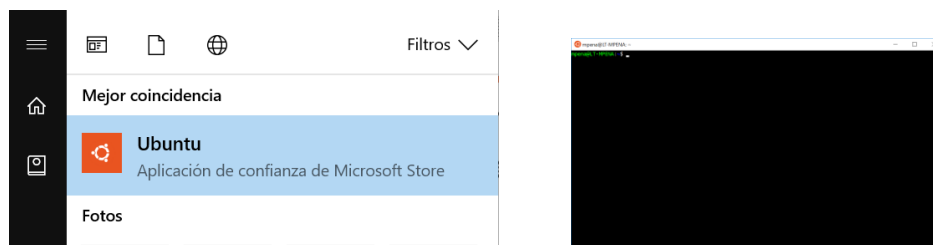
3. Solicitará el reinicio del equipo.
4. Después de Reiniciar el equipo hay que abrir el Microsoft Store y buscar Ubuntu



5. Dar clic en obtener:



6. Una vez instalado ir a menú de inicio y buscar Ubuntu para abrir la terminal:



5

Despliegue de Azure CLI 2.0

Es importante mencionar que gran parte de la base de Azure, no como todo mundo piensa, esta basado en Open Source, uno de los ejemplos es Azure CLI. Este CLI sirve para gestionar todo el mundo Azure.

- Ejecución:

En este punto realizaremos el despliegue de Azure CLI 2.0 dentro del bash instalado en Windows 10.

1. Agregar los repositorios de Microsoft para Azure Cli:

```
# echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ wheezy main"
| \ sudo tee /etc/apt/sources.list.d/azure-cli.list
```

```
# sudo apt-key adv --keyserver packages.microsoft.com --recv-keys
52E16F86FEE04B979B07E28DB02C46DF417A0893
```

2. Instalar soporte de repositorios por medio de HTTPS. (Mencionar punto de seguridad).

```
# sudo apt-get install -y apt-transport-https
```

```
psmapad@LTMPEA:/mnt/c/Users/mpena$ sudo apt-get install apt-transport-https
[sudo] password for psmad:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

3. Actualizar listado de repositorios e instalar azure-cli.

```
# sudo apt-get update && sudo apt-get install -y azure-cli
```

```
psmapad@LTMPEA:/mnt/c/Users/mpena$ sudo apt-get update && sudo apt-get install azure-cli
Obj:1 http://archive.ubuntu.com/ubuntu xenial InRelease
Des:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Des:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Des:4 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [384 kB]
Des:5 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [657 kB]
Des:6 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [170 kB]
Des:7 http://security.ubuntu.com/ubuntu xenial-security/restricted amd64 Packages [7 472 B]
Des:8 http://security.ubuntu.com/ubuntu xenial-security/restricted Translation-en [2 412 B]
```

```
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
snap-confine
Utilice «sudo apt autoremove» para eliminarlo.
Se actualizarán los siguientes paquetes:
azure-cli
1 actualizados, 0 nuevos se instalarán, 0 para eliminar y 80 no actualizados.
```

4. Después de esto se puede utilizar Azure CLI 2.0:

```
psmapad@LTMPEA:/mnt/c/Users/mpena$ az login --username mpena@bside.com.mx
Password:
{
  "cloudName": "AzureCloud",
  "id": "...",
  "isDefault": true,
  "name": "...",
  "state": "Enabled",
  "tenantId": "...",
  "user": {
    "name": "...",
    "type": "user"
  }
}
```

5. Ejemplo de Listado de Resource Group

```
# az group list -o table
```

```
mpena@LT-MPEA:~$ az group list -o table
Name                                Location    Status
-----
eastus                             Succeeded
eastus                             Succeeded
southcentralus                    Succeeded
eastus                             Succeeded
southcentralus                    Succeeded
southcentralus                    Succeeded
southcentralus                    Succeeded
```

Despliegue de Docker CE en Ubuntu

Se mostrará la instalación de Docker CE para un ambiente Ubuntu, en caso de tener hacer el laboratorio desde una máquina con Windows realizar las configuraciones para el sistema que se muestran en este documento.

Para Windows

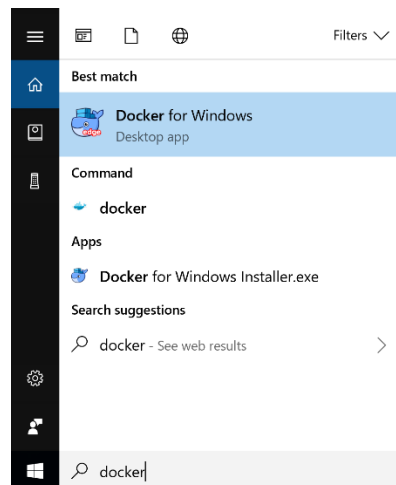
1. Instalar Docker CE para Windows

Se mostrará la instalación de Docker CE para un ambiente Ubuntu, en caso de tener alguna otra distribución de Linux, algunos comandos podrían cambiar.

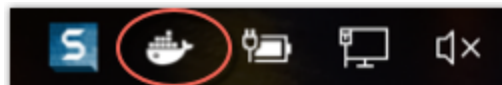
En caso de ambiente Windows se debe tener instalado el cliente de Docker para Windows, el cual se puede descargar de la siguiente liga:

<https://download.docker.com/win/stable/Docker%20for%20Windows%20Installer.exe>

2. Asegurarse que el proceso de Docker se haya instalado correctamente



3. Validar en la barra de tareas que el icono del logo de Docker aparezca, si es así el Docker Engine está funcionando correctamente.



Para Ubuntu

A continuación, se muestran los pasos para la instalación de Docker Ce desde la línea de comandos de Linux.

1. Desinstalar distribuciones anteriores en caso de tenerlas

```
# $ sudo apt-get remove docker docker-engine docker.io
```

2. Instalar los paquetes que permiten el uso de repositorios sobre Https

```
# sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  software-properties-common
```

```
juliodevz@LT-JZEPEDA:~$ sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
The following package was automatically installed and is no longer required:
  libfreetype6
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libcurl4 python3-software-properties
The following NEW packages will be installed:
```

3. Agregar la llave gPG:

```
# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
juliodevz@LT-JZEPEDA:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
juliodevz@LT-JZEPEDA:~$ sudo apt-key fingerprint 0EBFCD88
pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid   [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

4. Asignar repositorio estable

```
# sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

5. Actualizar paquetes de repositorio de información

```
# sudo apt-get update
```

6. Instalar la última versión de Docker CE

sudo apt-get install docker-ce

```
juliodevz@LT-JZEPEDA:~$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:3 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Hit:4 https://packages.microsoft.com/repos/azure-cli bionic InRelease
Hit:5 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:6 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:7 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [3161 B]
Fetched 67.6 kB in 7s (10.3 kB/s)
Reading package lists... Done
juliodevz@LT-JZEPEDA:~$ sudo apt-get update
Hit:1 https://packages.microsoft.com/repos/azure-cli bionic InRelease
Hit:2 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:5 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:6 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
juliodevz@LT-JZEPEDA:~$ sudo apt-get install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfreetype6
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
```

7. Validar instalación ejecutando un hola mundo

sudo docker run hello-world

```
juliodevz@LT-JZEPEDA:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

juliodevz@LT-JZEPEDA:~$
```

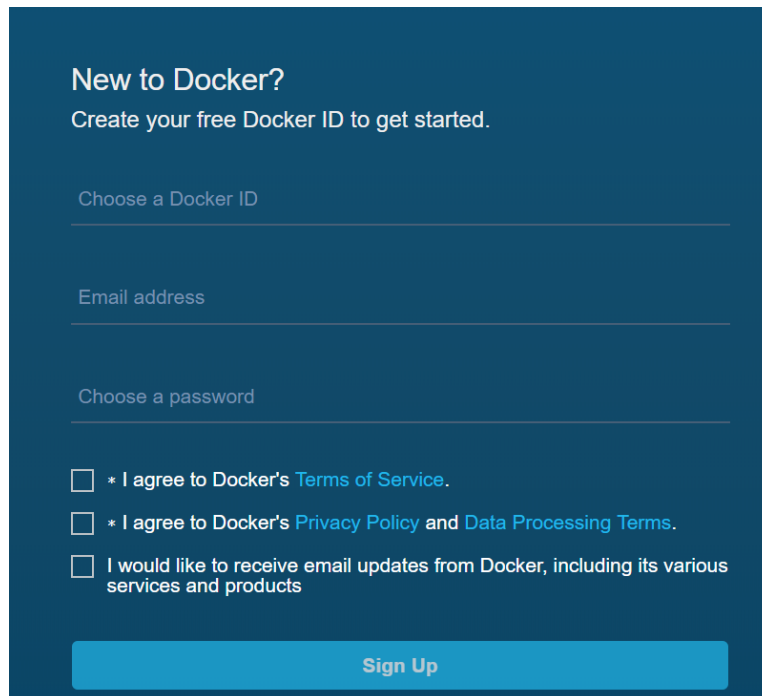
Contenedores de Docker Hub

En este punto se obtendrá el registro en Docker Hub para poder hacer uso de imágenes creadas por la comunidad y se encuentran dentro de un registro público.

1. Obtener una cuenta de Docker Hub ingresando en la siguiente liga y completando el formulario

<https://hub.docker.com>

Nota: Almacenar usuario y password creados pues se utilizarán más adelante.



10

2. Firmarse con usuario y password de Docker Hub con el siguiente comando
docker login

```
juliodevz@LT-JZEPEDA:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://
ub.docker.com to create one.
Username: juliodevz
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
```

3. Obtener las imágenes de nginx y TomCat de Docker Registry
docker search nginx
docker search TomCat

```
juliodevz@LT-JZEPEDA:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
a5a6f2f73cd8: Pull complete
1ba02017c4b2: Pull complete
33b176c904de: Pull complete
Digest: sha256:5d32f60db294b5deb55d078cd4feb410ad88e6fe77500c87d3970eca97f54dba
Status: Downloaded newer image for nginx:latest
```

```
juliodevz@LT-JZEPEDA:~$ docker pull tomcat
Using default tag: latest
latest: Pulling from library/tomcat
54f7e8ac135a: Pull complete
d6341e30912f: Pull complete
087a57faf949: Pull complete
95065f220961: Pull complete
0887630ce576: Pull complete
c375d1959fab: Pull complete
e00a5e6055cc: Pull complete
8319f5fb56cf: Pull complete
258c74eb25ab: Pull complete
6eb86d11d371: Pull complete
adb5a4a6adf5: Pull complete
4685b434f297: Pull complete
Digest: sha256:e394c2f94eee344300e68d7183f3f19d272316f9691b280ac0e3911ea610e590
Status: Downloaded newer image for tomcat:latest
```

4. Validar funcionamiento de las imágenes con el siguiente comando
`#docker run -it -p 8888:8080 tomcat`
`#docker run --name some-nginx -d -p 8080:80 nginx`

ACR despliegue de imágenes de contenedores

En este apartado se realizará un despliegue de los contenedores que se obtuvieron en la sección anterior dentro de el servicio de Azure Container Registry

1. Firmarse con sus credenciales que contengan la suscripción de Azure para poder generar recursos
`# az login`

```
juliodevz@LT-JZEPEDA:~$ az login
Note, we have launched a browser for you to login. For old experience with device code, use "az login --use-device-code"
You have logged in. Now let us find all subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
```

Una vez firmado en el portal de azure se desplegará la información de la suscripción.

Nota: En caso de tener varias suscripciones con la misma cuenta, se debe indicar la suscripción con la que se trabajará con el siguiente comando

`# az account set -s <Nombre de la suscripción>`

2. Crear un grupo de recursos para los productos del laboratorio a utilizar
`# az group create --name myResourceGroup --location EastUS`

```
juliodevz@LT-JZEPEDA:~$ az group create --name containerscli0 --location EastUS
{
  "id": "/subscriptions/
  "location": "eastus",
  "managedBy": null,
  "name": "containerscli0",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

3. Creación del recurso ACR

```
#az acr create --resource-group myResourceGroup --name <acrName> --sku Basic
```

```
juliodevz@LT-JZEPEDA:~$ az acr create --resource-group containerscli0 --name azregistry0 --sku Basic
{
  "adminUserEnabled": false,
  "creationDate": "2018-12-05T10:13:04.451613+00:00",
  "id": "/subscriptions/containerscli0/resourceGroups/containerscli0/providers/Microsoft.ContainerRegistry/registries/azregistry0",
  "location": "eastus",
  "loginServer": "azregistry0.azurecr.io",
  "name": "azregistry0",
  "provisioningState": "Succeeded",
  "resourceGroup": "containerscli0",
  "sku": {
    "name": "Basic",
    "tier": "Basic"
  },
  "status": null,
  "storageAccount": null,
  "tags": {},
  "type": "Microsoft.ContainerRegistry/registries"
}
```

12

4. Firmarse en el ACR y listar las imágenes existentes (hasta este punto no hay imágenes en el servicio de Azure).

```
#az acr login --name <acrName>
```

```
# az acr repository list --name <acrName> --output table
```

```
juliodevz@LT-JZEPEDA:~$ az acr login --name azregistry0
Login Succeeded
juliodevz@LT-JZEPEDA:~$ az acr login --name azregistry0
Login Succeeded
```

5. Obtener la dirección pública del ACR

```
# az acr list --resource-group <myResourceGroup> --query "[].{acrLoginServer:loginServer}" --output table
```

```
juliodevz@LT-JZEPEDA:~$ az acr list --resource-group containerscli0 --query "[].{acrLoginServer:loginServer}" --output table
acrLoginServer
-----
azregistry0.azurecr.io
```

Nota: Almacenar el valor de AcrLoginServer pues se utilizará en los posteriores pasos



- Colocar tags a las imágenes de los contenedores previo a colocarlas en el ACR.
docker tag nombrecontenedor <acrLoginServer>/nombrecontenedor:<tag>

Listar las imágenes existentes

#docker images

```
juliodevz@LT-JZEPEDA:~$ docker images
REPOSITORY                                TAG                IMAGE ID           CREATED            SIZE
azregistry0.azurecr.io/nginx              c2                 568c4670fa80      7 days ago        109MB
nginx                                       latest            568c4670fa80      7 days ago        109MB
azregistry0.azurecr.io/tomcat              c1                 78b258e36eed      2 weeks ago       463MB
tomcat                                      latest            78b258e36eed      2 weeks ago       463MB
juliodevz@LT-JZEPEDA:~$
```

- Insertar las imágenes dentro del ACR
docker push <acrLoginServer>/<contenedor>:<tag>

```
juliodevz@LT-JZEPEDA:~$ docker push azregistry0.azurecr.io/tomcat:c1
The push refers to repository [azregistry0.azurecr.io/tomcat]
99460a7016ad: Pushed
1c3c62be78b6: Pushed
a0a80c0efe37: Pushed
d4397eab7439: Pushed
e8995be06405: Pushed
586032a40815: Pushed
86becce36874: Pushed
360cf37035a0: Pushed
1850621c23b2: Pushed
8f7ee6d76fd9: Pushed
c23711a84ad4: Pushed
90d1009ce6fe: Pushed
c1: digest: sha256:6878215065a0bf6aec78ca133f0d1be6ff80f4ca52dd4fd53e7a246327223945 size: 2836
juliodevz@LT-JZEPEDA:~$ docker push azregistry0.azurecr.io/nginx:c2
The push refers to repository [azregistry0.azurecr.io/nginx]
ece4f9fdef59: Pushed
ad5345cbb119: Pushed
ef68f6734aa4: Pushed
c2: digest: sha256:87e9b6904b4286b8d41bba4461c0b736835fcc218f7ecbe5544b53fdd467189f size: 948
juliodevz@LT-JZEPEDA:~$
```

13

Repetir la instrucción para cada imagen

Visualizar el contenido del recurso

az acr repository list --name <acrName> --output table

```
^[[Bjuliodevz@LT-JZEPEDA:~$ az acr repository list --name azregistry0 --output table
Result
-----
nginx
tomcat
```

AKS Crear clúster

En este apartado se realizará la creación de un cluster de Kubernetes mediante Aks

1. Dar permisos de creación de aks

```
#az ad sp create-for-rbac --skip-assignment
```

```
azure-voting-app-redis>az ad sp create-for-rbac --skip-assignment

{
  "appId": "555e5bd0-63ef-4e21-8000-000000000000",
  "displayName": "azure-cli-2018-10-25-15-54-38",
  "name": "http://azure-cli-2018-10-25-15-54-38/",
  "password": "f122bdca-b163-41e8-8b6a-c7971e55837f",
  "tenant": "555e5bd0-63ef-4e21-8000-000000000000"
}
```

2. Obtener las credenciales del ACR para permitir que el AKS use las imágenes del ACR

```
# az acr show --resource-group myResourceGroup --name <acrName> --query "id" --output tsv
```

```
# az role assignment create --assignee <appld> --scope <acrId> --role Reader
```

```
/subscriptions/containers\azure-voting-app-redis>az acr show --resource-group containerscli0 --name azregistry0 --query "id" --output tsv
/resourceGroups/containerscli0/providers/Microsoft.ContainerRegistry/registries/azregistry0

C:\Users\Julio Zepeda\Documents\containers\azure-voting-app-redis>az role assignment create --assignee 5dbbb0aa-d83f-4571-a87a-33600e1e98f1 --scope /subscriptions
/resourceGroups/containerscli0/providers/Microsoft.ContainerRegistry/registries/azregistry0 --role Reader

{
  "canDelegate": null,
  "id": "/subscriptions/containerscli0/resourceGroups/containerscli0/providers/Microsoft.ContainerRegistry/registries/azregistry0/providers/Microsoft.Authorization/roleAssignments/6e02",
  "name": "6e0242b5-e4ec-43e9-8663-0cde66ddffed",
  "principalId": "6cf840ef-b920-4bb6-b046",
  "resourceGroup": "containerscli0",
  "roleDefinitionId": "/subscriptions/containerscli0/resourceGroups/containerscli0/providers/Microsoft.Authorization/roleDefinitions/acdd72a7-3385-48ef-bd42-f606fba81ae7",
  "scope": "/subscriptions/containerscli0/resourceGroups/containerscli0/providers/Microsoft.ContainerRegistry/registries/azregistry0",
  "type": "Microsoft.Authorization/roleAssignments"
}
```

14

3. Crear el clúster de Kubernetes

```
# az aks create \
  --resource-group myResourceGroup \
  --name myAKSCluster \
  --node-count 1 \
  --service-principal <appld> \
  --client-secret <password> \
  --generate-ssh-keys
```

```
Bash
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI 2.0
Type "help" to learn about Cloud Shell

julio@Azure:~$ az aks create --resource-group kubertessamplez0east --name myFirstAKSCluster20 --node-count 1 --service-principal 5dbbb0aa-d83f-4571-a87a-33600e1e98f1 --client-secret f122bdca-b163-41e8-8b6a-c7971e55837f --generate-ssh-keys
SSH key files '/home/julio/.ssh/id_rsa' and '/home/julio/.ssh/id_rsa.pub' have been generated under ~/.ssh to allow SSH access to the VM. If using machines without permanent storage like Azure Cloud Shell without an attached file share, back up your keys to a safe location.
- Running ...
```

- Conectarse al clúster usando kubectl y despliegue de aplicativo
az aks get-credentials --resource-group <grupo de recursos> --name "nombre-aks"

```
(\containers\azure-voting-app-redis>az aks get-credentials --resource-group kubertessamplez0east --name myFirstAKSClusterZ0
Merged "myFirstAKSClusterZ0" as current context in C:\Users\Julio Zepeda\.kube\config

(\containers\azure-voting-app-redis>kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
aks-nodepool1-51936561-0            Ready    agent    13m    v1.9.11

(\containers\azure-voting-app-redis>az acr list --resource-group containerscli0 --query "[].{acrLoginServer:loginServer}" --output table
acrLoginServer
-----
azregistry0.azurecr.io

-redis>vi azure-vote-all-in-one-redis.yaml

C:\Users\Julio Zepeda\Documents\containers\azure-voting-app-redis>kubectl apply -f azure-vote-all-in-one-redis.yaml
deployment.apps "azure-vote-back" created
service "azure-vote-back" created
deployment.apps "azure-vote-front" created
service "azure-vote-front" created
```

- Visualizar dashboard de Kubernetes
az aks browse --resource-group kubertessamplez0east --name myFirstAKSClusterZ0

Eliminar Ambiente

Al terminar la ejecución del laboratorio es importante eliminar todos los recursos, hay que recordar que parte de los beneficios de la nube es que se paga por el tiempo que se usa.

```
# az group delete --name <myResourceGroup>
```

Conclusiones

En esta documentación técnica se incluye los detalles que se realizaron para cubrir los requerimientos de Microsoft. En esto se incluye:

- Microsoft Loves Linux.
- Integración de Linux en Windows 10.
- Obtención y despliegue de imágenes de contenedores.
- Creación y gestión de recursos ACR y AKS desde la consola.

Contacto

