

# Memoria Docker | Informática Como Servicio

---

Autor - Lucas José Lara García

**Opción 1** - Desplegar un contenedor con wordpress:latest y un contenedor con mysql:5.7. El contenedor wordpress consume el puerto 3306 del contenedor mysql usando la url pública.

---

En esta memoria se describe como ha sido el proceso para realizar la práctica de PaaS de Azure paso a paso. Para poder empezar a trabajar se ha configurado el entorno descargando docker en el sistema y creando la imagen *bulletinboard:1.0* con la que se ha trabajado en la práctica. Además, se ha instalado la consola de azure en ubuntu con la orden:

```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
```

De igual manera es necesario instalar *Helm*, la dependencia para el despliegue de contenedores, ejecutando el comando:

```
curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3  
| bash
```

---

## Login en azure

---

En primer lugar se ha creado una cuenta de Azure asociada a la de la udc para poder obtener una serie de servicios gratuitos. A continuación se ha iniciado sesión.

```
az login
```



```
mlpc@lucas: ~  
mlpc@lucas:~$ az login  
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in  
the web browser. If no web browser is available or if the web browser fails to open, use device code flow with 'az login --use-devi  
ce-code'.  
[  
  {  
    "cloudName": "AzureCloud",  
    "homeTenantId": "defae9d9-59fe-4fa9-b151-12ed7438e099",  
    "id": "bab762ce-fe04-485f-a47b-7603cdfae631",  
    "isDefault": true,  
    "managedByTenants": [],  
    "name": "Azure for Students",  
    "state": "Enabled",  
    "tenantId": "defae9d9-59fe-4fa9-b151-12ed7438e099",  
    "user": {  
      "name": "thehaias@outlook.es",  
      "type": "user"  
    }  
  }  
]
```

## Creación del Container Registry

---

Seguidamente se ha creado un nuevo recurso *"gruporecursoslucas"*. Este recurso es el registro de imágenes Docker de Azure:

```
az group create --name gruporecursoslucas --location "West Europe"
```

```
mipc@lucas:~$ az group create --name gruporecursoslucas --location "West Europe"
{
  "id": "/subscriptions/bab762ce-fe04-485f-a47b-7603cdfae631/resourceGroups/gruporecursoslucas",
  "location": "westeurope",
  "managedBy": null,
  "name": "gruporecursoslucas",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
mipc@lucas:~$
```

Y a continuación se ha creado el *Container Registry* *"planserviciolucas"* sobre el grupo de recursos creado en el paso anterior:

```
az acr create --name planserviciolucas --resource-group gruporecursoslucas --sku Basic
```

```
mipc@lucas:~$ az acr create --name planserviciolucas --resource-group gruporecursoslucas --sku Basic
{
  "adminUserEnabled": false,
  "anonymousPullEnabled": false,
  "creationDate": "2022-11-10T17:45:45.174030+00:00",
  "dataEndpointEnabled": false,
  "dataEndpointHostNames": [],
  "encryption": {
    "keyVaultProperties": null,
    "status": "disabled"
  },
  "id": "/subscriptions/bab762ce-fe04-485f-a47b-7603cdfae631/resourceGroups/gruporecursoslucas/providers/Microsoft.ContainerRegistry/registries/planserviciolucas",
  "identity": null,
  "location": "westeurope",
  "loginServer": "planserviciolucas.azurecr.io",
  "name": "planserviciolucas",
  "networkRuleBypassOptions": "AzureServices",
  "networkRuleSet": null,
  "policies": {
    "azureAdAuthenticationAsArmPolicy": {
      "status": "enabled"
    },
    "exportPolicy": {
      "status": "enabled"
    },
    "quarantinePolicy": {
      "status": "disabled"
    },
    "retentionPolicy": {
      "days": 7,
      "lastUpdatedTime": "2022-11-10T17:45:53.855413+00:00",
      "status": "disabled"
    },
    "softDeletePolicy": {
      "lastUpdatedTime": "2022-11-10T17:45:53.855413+00:00",
      "retentionDays": 7,
      "status": "disabled"
    },
    "trustPolicy": {
      "status": "disabled",
      "type": "Notary"
    }
  }
}
```

---

## Push de la imagen al Container registry de Azure

---

Primeramente, para hacer el push de la imagen hay que crear un **tag** y además debe tener un formato específico, en concreto:

```
<nombreRegistro>.azurecr.io/<nombre>:<version>
```

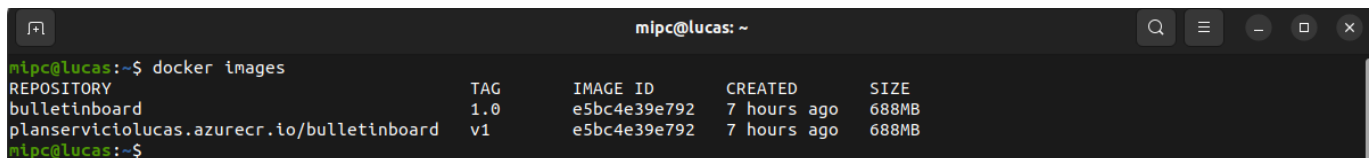
A continuación creamos el tag para la imagen *bulletinboard* originada localmente al configurar el entorno.

```
docker tag bulletinboard:1.0 planserviciolucas.azurecr.io/bulletinboard:v1
```

Y seguidamente para comprobarlo listamos las imágenes Docker del sistema:

```
docker images
```

El resultado es el siguiente:



```
mipc@lucas: ~  
mipc@lucas:~$ docker images  
REPOSITORY                                TAG      IMAGE ID       CREATED        SIZE  
bulletinboard                             1.0      e5bc4e39e792   7 hours ago    688MB  
planserviciolucas.azurecr.io/bulletinboard v1       e5bc4e39e792   7 hours ago    688MB  
mipc@lucas:~$
```

Seguidamente, al tratarse de un repositorio privado, hay que habilitar el login ejecutando el comando:

```
az acr update -n planserviciolucas --admin-enabled true
```

```
mlpc@lucas: ~  
mlpc@lucas:~$ az acr update -n planserviciolucas --admin-enabled true  
{  
  "adminUserEnabled": true,  
  "anonymousPullEnabled": false,  
  "creationDate": "2022-11-10T17:45:45.174030+00:00",  
  "dataEndpointEnabled": false,  
  "dataEndpointHostNames": [],  
  "encryption": {  
    "keyVaultProperties": null,  
    "status": "disabled"  
  },  
  "id": "/subscriptions/bab762ce-fe04-485f-a47b-7603cdfae631/resourceGroups/gruporecursoslucas/providers/Microsoft.ContainerRegistry/registries/planserviciolucas",  
  "identity": null,  
  "location": "westeurope",  
  "loginServer": "planserviciolucas.azurecr.io",  
  "name": "planserviciolucas",  
  "networkRuleBypassOptions": "AzureServices",  
  "networkRuleSet": null,  
  "policies": {  
    "azureAdAuthenticationAsArmPolicy": {  
      "status": "enabled"  
    },  
    "exportPolicy": {  
      "status": "enabled"  
    },  
    "quarantinePolicy": {  
      "status": "disabled"  
    },  
    "retentionPolicy": {  
      "days": 7,  
      "lastUpdatedTime": "2022-11-10T18:05:42.289744+00:00",  
      "status": "disabled"  
    },  
    "softDeletePolicy": {  
      "lastUpdatedTime": "2022-11-10T18:05:42.289744+00:00",  
      "retentionDays": 7,  
      "status": "disabled"  
    },  
    "trustPolicy": {  
      "status": "disabled",  
      "type": "Notary"  
    }  
  }  
}
```

Y si accedemos al portal de azure a la sección *registro de contenedor* se puede ver el contenedor creado:

The screenshot shows the Azure portal interface for the 'planserviciolucas' container registry. The left sidebar contains navigation options like 'Inicio', 'Registro de actividad', 'Control de acceso (IAM)', 'Etiquetas', 'Inicio rápido', 'Eventos', 'Configuración', 'Claves de acceso', 'Cifrado', 'Identidad', 'Redes', 'Microsoft Defender for Cloud', 'Bloqueos', 'Servicios', 'Repositorios', 'Webhooks', 'Replicaciones', 'Tareas', 'Registros conectados (vista previa)', 'Permisos para el repositorio', and 'Directivas'. The main content area is titled 'planserviciolucas' and includes a search bar and action buttons (Mover, Eliminar, Actualizar). The 'Información esencial' section shows the resource group 'gruporecursoslucas', location 'West Europe', subscription 'Azure for Students', and ID 'bab762ce-fe04-485f-a47b-7603cdfae631'. The 'Uso' section displays a bar chart with values: 10.0 GIB (Incluido en la SKU), 0.25 GIB (Usado), and 0.00 GIB (Almacenamiento a.). The 'ACR Tasks' section provides instructions on compiling, executing, inserting, and revising containers. The 'Integraciones de seguridad del contenedor' section highlights 'Microsoft Defender for Cloud' for vulnerability management. The 'Métricas de registro' section shows a time range selector (1 hora, 6 horas, 12 horas, 1 día, 7 días, 30 días) and three charts: 'Recuento de extracción de imágenes', 'Recuento de inserción de imágenes', and 'Duración de la ejecución de la tarea'.

Posteriormente hacemos login por consola con el siguiente comando:

```
docker login planserviciolucas.azurecr.io
```

El resultado se muestra en la siguiente imagen:

```
mipc@lucas:~$ docker login planserviciolucas.azurecr.io
Username (00000000-0000-0000-0000-000000000000): planserviciolucas
Password:
WARNING! Your password will be stored unencrypted in /home/mipc/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
mipc@lucas:~$
```

Finalmente ya podemos hacer el push de nuestra imagen Docker local para su despliegue en Azure con el siguiente comando:

```
docker push miPlanServicio.azurecr.io/bulletinboard:v1
```

El resultado se muestra en la siguiente imagen:

```
mipc@lucas: ~
mipc@lucas:~$ docker push planserviciolucas.azurecr.io/bulletinboard:v1
The push refers to repository [planserviciolucas.azurecr.io/bulletinboard]
1dde6c8e1282: Pushed
a1baa19f8873: Pushed
dc0b74d1bd6f: Pushed
555d454d1ec3: Pushed
dd883cbb0cf7: Pushed
a3a3f69aeb8: Pushed
0da372da714b: Pushed
bf3841becf9d: Pushed
63866df00998: Pushed
2f9128310b77: Pushed
d9a5f9b8d5c2: Pushed
c01c63c6823d: Pushed
v1: digest: sha256:5f9ef5bd9d932357bb592c28c86138f62fe2121345541e61d09f2082ec56ede8 size: 2841
mipc@lucas:~$
```

---

## Creación del contenedor con la imagen recién subida a Azure

---

Una vez que se ha subido la imagen, hay que crear el contenedor que contendrá la imagen junto a las opciones necesarias. Para ello utilizamos el siguiente comando:

```
az container create --resource-group gruporecursoslucas --name test-
bulletinboard --image planserviciolucas.azurecr.io/bulletinboard:v1 --dns-
name-label test-bulletinboard --ports 8080
```

El resultado es el siguiente:

```
mipc@lucas: ~  
mipc@lucas:~$ az container create --resource-group gruporecursoslucas --name test-bulletinboard --image planserviciolucas.azurecr.io/bulletinboard:v1 --dns-name-label lucascontainer --ports 8080  
Image registry username: planserviciolucas  
Image registry password:  
{  
  "containers": [  
    {  
      "command": null,  
      "environmentVariables": [],  
      "image": "planserviciolucas.azurecr.io/bulletinboard:v1",  
      "instanceView": {  
        "currentState": {  
          "detailStatus": "",  
          "exitCode": null,  
          "finishTime": null,  
          "startTime": "2022-11-10T18:43:27.762000+00:00",  
          "state": "Running"  
        },  
        "events": [  
          {  
            "count": 1,  
            "firstTimestamp": "2022-11-10T18:42:49+00:00",  
            "lastTimestamp": "2022-11-10T18:42:49+00:00",  
            "message": "pulling image \"planserviciolucas.azurecr.io/bulletinboard@sha256:5f9ef5bd9d932357bb592c28c86138f62fe2121345541e61d09f2082ec56ede8\"",  
            "name": "Pulling",  
            "type": "Normal"  
          },  
          {  
            "count": 1,  
            "firstTimestamp": "2022-11-10T18:43:16+00:00",  
            "lastTimestamp": "2022-11-10T18:43:16+00:00",  
            "message": "Successfully pulled image \"planserviciolucas.azurecr.io/bulletinboard@sha256:5f9ef5bd9d932357bb592c28c86138f62fe2121345541e61d09f2082ec56ede8\"",  
            "name": "Pulled",  
            "type": "Normal"  
          },  
          {  
            "count": 1,  
            "firstTimestamp": "2022-11-10T18:43:27+00:00",  
            "lastTimestamp": "2022-11-10T18:43:27+00:00",  
            "message": "Started container",  
            "name": "Started",  
            "type": "Normal"  
          }  
        ]  
      }  
    }  
  ]  
}
```

---

## Web con la imagen corporativa

---

Finalmente para comprobar que todo se ha realizado correctamente se puede visitar la imagen desplegada en el siguiente enlace:

```
http://lucascontainer.westeurope.azurecontainer.io:8080/
```

El resultado se muestra en la siguiente imagen:

