

CS2032 - Cloud Computing (Ciclo 2024-2)

Virtualización con contenedores

Semana 4 - Taller 4: Contenedores

ELABORADO POR: GERALDO COLCHADO

Con apoyo de Asistente de Cátedra y Laboratorio:

- Sofía García (sofia.garcia@utec.edu.pe)

Contenido

Contenedores

1. **Objetivo del taller 4**
2. Ejercicio 1: Subir imágenes de Aplicación Multi Contenedor a hub.docker.com
3. Ejercicio 2: Desplegar Aplicación Multi Contenedor en otras computadoras
4. Ejercicio 3: Diagramar Arquitectura de Solución de Aplicación Multi Contenedor
5. Ejercicio 4: Catálogo de Apis
6. Cierre

Objetivo del taller 4:

Contenedores

- Aprender a subir imágenes de Aplicación Multi Contenedor a hub.docker.com
- Aprender a desplegar Aplicación Multi Contenedor en otras computadoras
- Aprender a diagramar Arquitectura de Solución de Aplicación Multi Contenedor
- Aprender a documentar Apis en un Catálogo de Apis

Contenido

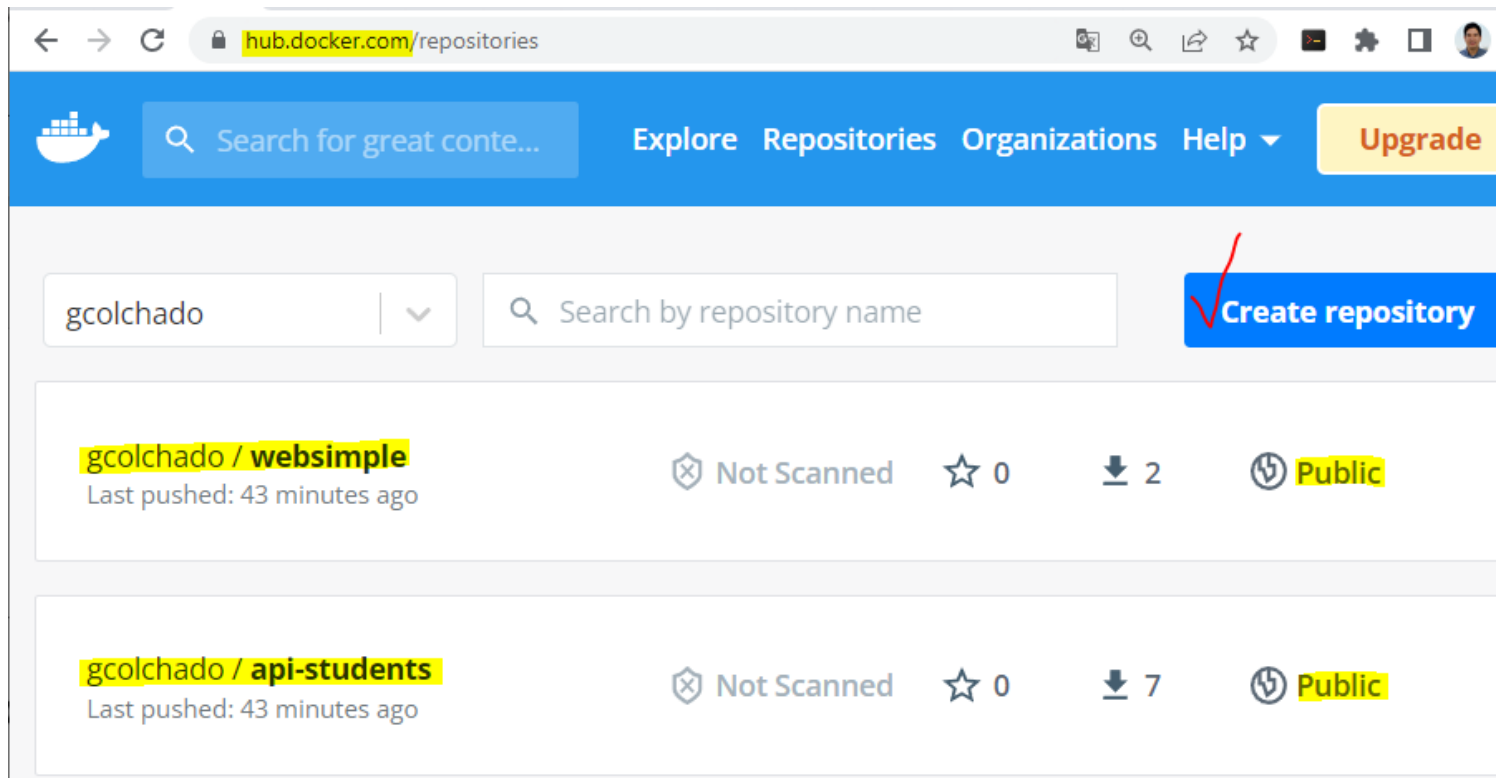
Contenedores

1. Objetivo del taller 4
2. **Ejercicio 1: Subir imágenes de Aplicación Multi Contenedor a hub.docker.com**
3. Ejercicio 2: Desplegar Aplicación Multi Contenedor en otras computadoras
4. Ejercicio 3: Diagramar Arquitectura de Solución de Aplicación Multi Contenedor
5. Ejercicio 4: Catálogo de Apis
6. Cierre

Ejercicio 1:

Subir imágenes de Aplicación Multi Contenedor a hub.docker.com

- Paso 1:** Cree dos repositorios públicos en hub.docker.com para las imágenes (Opcional)



Ejercicio 1:

Subir imágenes de Aplicación Multi Contenedor a
hub.docker.com

- **Paso 2:** Modifique el archivo compose.yml en “MV Desarrollo”

```
:~/compose $ cat compose.yml

services:
  api-students:
    build: ./api-students
    ports:
      - 8000:8000
  websimple:
    build: ./websimple
    ports:
      - 8080:80
```



```
version: "3.3"

services:
  api-students:
    build: ./api-students
    image: gcolchado/api-students
    ports:
      - 8000:8000
  websimple:
    build: ./websimple
    image: gcolchado/websimple
    ports:
      - 8080:80
```

Ejercicio 1:

Subir imágenes de Aplicación Multi Contenedor a hub.docker.com

- **Paso 3:** Ejecute la aplicación multi contenedor
`$ docker compose up -d` (Ejecuta en segundo plano)
- **Paso 4:** Analice las imágenes creadas y los contenedores y los logs de ejecución
`$ docker images`
`$ docker ps -a`
`$ docker compose logs`
- **Paso 5:** Ingrese a hub docker (Docker Registry) con este comando:
`$ docker login -u gcolchado` (Reemplace amarillo)
- **Paso 6:** Suba sus imágenes a los repositorios públicos creados:
`$ docker compose push`
- **Paso 7:** Salga del hub docker (Docker Registry)
`$ docker logout`
- **Paso 8:** Detenga la aplicación multi contenedor
`$ docker compose down`

Contenido

Contenedores

1. Objetivo del taller 4
2. Ejercicio 1: Subir imágenes de Aplicación Multi Contenedor a hub.docker.com
3. **Ejercicio 2: Desplegar Aplicación Multi Contenedor en otras computadoras**
4. Ejercicio 3: Diagramar Arquitectura de Solución de Aplicación Multi Contenedor
5. Ejercicio 4: Catálogo de Apis
6. Cierre

Ejercicio 2:


Desplegar Aplicación Multi Contenedor en otras computadoras

- **Paso 1:** Ingrese por ssh a la [máquina virtual](#) “MV Pruebas” y abra puerto 8080
- **Paso 2:** Cree el directorio [/home/ubuntu/compose](#) e ingrese
- **Paso 3:** Cree el archivo compose.yml

```
services:
  api-students:
    build: ./api-students
    image: gcolchado/api-students
    ports:
      - 8000:8000
  websimple:
    build: ./websimple
    image: gcolchado/websimple
    ports:
      - 8080:80
```



```
services:
  api-students:
    image: gcolchado/api-students
    ports:
      - 8000:8000
  websimple:
    image: gcolchado/websimple
    ports:
      - 8080:80
```



Ejercicio 2:

Desplegar Aplicación Multi Contenedor en otras computadoras

- **Paso 4:** Ejecute la aplicación Multi Contenedor:
\$ **docker compose up -d**
- **Paso 5:** Analice las imágenes creadas y los contenedores y logs de ejecución
\$ docker images
\$ docker ps -a
\$ **docker compose logs**
- **Paso 6:** Pruebe en navegador y con postman
- **Paso 7:** Detenga la aplicación multi contenedor
\$ **docker compose down**

Contenido

Contenedores

1. Objetivo del taller 4
2. Ejercicio 1: Subir imágenes de Aplicación Multi Contenedor a hub.docker.com
3. Ejercicio 2: Desplegar Aplicación Multi Contenedor en otras computadoras
4. **Ejercicio 3: Diagramar Arquitectura de Solución de Aplicación Multi Contenedor**
5. Ejercicio 4: Catálogo de Apis
6. Cierre

Ejercicio 3:

Diagramar Arquitectura de Solución de Aplicación Multi Contenedor

- Paso 1: Ingrese a <https://draw.io/> y siga las indicaciones del docente para elaborar el Diagrama de Arquitectura de Solución de Aplicación Multi Contenedor

Contenido

Contenedores

1. Objetivo del taller 4
2. Ejercicio 1: Subir imágenes de Aplicación Multi Contenedor a hub.docker.com
3. Ejercicio 2: Desplegar Aplicación Multi Contenedor en otras computadoras
4. Ejercicio 3: Diagramar Arquitectura de Solución de Aplicación Multi Contenedor
5. **Ejercicio 4: Catálogo de Apis**
6. Cierre

Ejercicio 4:

Catálogo de Apis

- Paso 1: Cree un repositorio en github con los 2 archivos del directorio catalog

The screenshot shows the GitHub interface for a repository named 'catalog' owned by 'gcolchado'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows three commits: an initial commit for 'README.md' 32 minutes ago, and two subsequent commits for 'api-students.yaml' and 'apis-pet-store.yaml' 30 minutes ago, both added via upload.

gcolchado / catalog

Issues Pull requests Actions Projects Wiki Security Insights Settings

catalog Public Pin Unwatch 1

main 1 Branch 0 Tags Go to file Add file Code

gcolchado Add files via upload		5f16eb4 · 30 minutes ago 3 Commits
README.md	Initial commit	32 minutes ago
api-students.yaml	Add files via upload	30 minutes ago
apis-pet-store.yaml	Add files via upload	30 minutes ago

Ejercicio 4:

Catálogo de Apis

- Paso 2: Descargue el repositorio de github en la MV Pruebas con git clone

```
/home/ubuntu/catalog
~/catalog (main) $ ls -l
total 28
-rw-rw-r-- 1 ubuntu ubuntu    9 Apr 21 23:39 README.md
-rw-rw-r-- 1 ubuntu ubuntu 2972 Apr 21 23:39 api-students.yaml
-rw-rw-r-- 1 ubuntu ubuntu 17761 Apr 21 23:39 apis-pet-store.yaml
~/catalog (main) $
```

- Paso 3: Ejecute el contenedor de swagger ui (Analice el parámetro **-e** y **-v** que explique el docente)

```
$ docker run -d -p 8080:8080 -e SWAGGER_JSON=/catalog/api-students.yaml -v /home/ubuntu/catalog:/catalog swaggerapi/swagger-ui
```

Ejercicio 4:

Catálogo de Apis

- Paso 4: Visualice el Catálogo de Apis para api-students

The screenshot shows the Swagger UI interface for an API. At the top, the URL `54.147.227.204:8080` is displayed. Below this, the Swagger logo is visible, along with the text `Supported by SMARTBEAR`. A search bar contains the file `/api-students.yaml`, and an `Explore` button is to its right. The main heading is **API de Estudiantes**, with version tags `1.0.0` and `OAS 3.0`. Below the heading, the file `/api-students.yaml` is listed, followed by the description: `API para gestionar registros de estudiantes.`

A `Servers` dropdown menu is set to `http://localhost:8000 - Servidor de desarrollo`. Under the `default` server section, five API endpoints are listed:

- GET** `/students`: Obtener todos los estudiantes
- POST** `/students`: Crear un nuevo estudiante
- GET** `/student/{id}`: Obtener un estudiante por ID
- PUT** `/student/{id}`: Actualizar un estudiante por ID
- DELETE** `/student/{id}`: Eliminar un estudiante por ID

At the bottom right, there is a `VALID` button and a `{-}` icon.

Ejercicio 4:

Catálogo de Apis

- Paso 5: Visualice el Catálogo de Apis para apis-pet-store

The screenshot shows the Swagger UI for the Petstore API. At the top, the address bar displays '54.147.227.204:8080'. The Swagger logo is on the left, and the URL '/apis-pet-store.yaml' is entered in the search bar, with an 'Explore' button to its right. Below the search bar, the title 'Swagger Petstore' is followed by version tags '1.0.7' and 'OAS 2.0'. A note indicates the base URL is 'petstore.swagger.io/v2'. A paragraph explains that this is a sample server and provides links to Swagger documentation and a special key 'special-key' for testing. Links for 'Terms of service', 'Contact the developer', 'Apache 2.0', and 'Find out more about Swagger' are listed. A 'Schemes' dropdown is set to 'HTTPS', and an 'Authorize' button is present. The main section, titled 'pet Everything about your Pets', lists three endpoints: a POST endpoint for uploading an image, a POST endpoint for adding a new pet, and a PUT endpoint for updating an existing pet. Each endpoint is shown with its method, path, description, and a lock icon.

54.147.227.204:8080

Swagger
Supported by SMARTBEAR

/apis-pet-store.yaml Explore

Swagger Petstore 1.0.7 OAS 2.0

[Base URL: petstore.swagger.io/v2]
/apis-pet-store.yaml

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [irc.freenode.net, #swagger](irc://freenode.net/#swagger). For this sample, you can use the api key `special-key` to test the authorization filters.

[Terms of service](#)
[Contact the developer](#)
[Apache 2.0](#)
[Find out more about Swagger](#)

Schemes
HTTPS

Authorize

pet Everything about your Pets Find out more

- POST** /pet/{petId}/uploadImage uploads an image
- POST** /pet Add a new pet to the store
- PUT** /pet Update an existing pet

Contenido

Contenedores

1. Objetivo del taller 4
2. Ejercicio 1: Subir imágenes de Aplicación Multi Contenedor a hub.docker.com
3. Ejercicio 2: Desplegar Aplicación Multi Contenedor en otras computadoras
4. Ejercicio 3: Diagramar Arquitectura de Solución de Aplicación Multi Contenedor
5. Ejercicio 4: Catálogo de Apis
6. **Cierre**

Cierre:

Contenedores - Qué aprendimos?

- Subir imágenes de Aplicación Multi Contenedor a hub.docker.com
- Desplegar Aplicación Multi Contenedor en otras computadoras
- Diagramar Arquitectura de Solución de Aplicación Multi Contenedor
- Documentar Apis en un Catálogo de Apis

Gracias

Elaborado por docente: Geraldo Colchado