

Trabajo Práctico 7 - Servidor de Build (de integración continua)

Desarrollo:

1- Poniendo en funcionamiento Jenkins

- Crear una imagen de Docker que se base en la imagen oficial de Jenkins y que tenga instalado .NET Core. Crear un archivo llamado Dockerfile.jenkins con el siguiente contenido:

```
EXPLORER Welcome Dockerfile.jenkins x
07-SERVIDOR-BUILD
Dockerfile.jenkins
1 FROM jenkins/jenkins:lts
2
3 USER root
4
5 # Instala dependencias necesarias
6 RUN apt-get update && apt-get install -y \
7     apt-transport-https \
8     software-properties-common \
9     wget
10
11 # Agrega el repositorio de Microsoft y actualiza
12 RUN wget -q https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb && \
13     dpkg -i packages-microsoft-prod.deb && \
14     apt-get update
15
16 # Instala el SDK de .NET Core
17 RUN apt-get install -y dotnet-sdk-7.0
18
19 USER jenkins
```

- Desde la misma ubicación donde tengas el archivo Dockerfile personalizado, ejecuta el siguiente comando para construir una nueva imagen de Docker con .NET Core y Jenkins. Esto creará una nueva imagen de Docker llamada jenkins-with-dotnetcore:

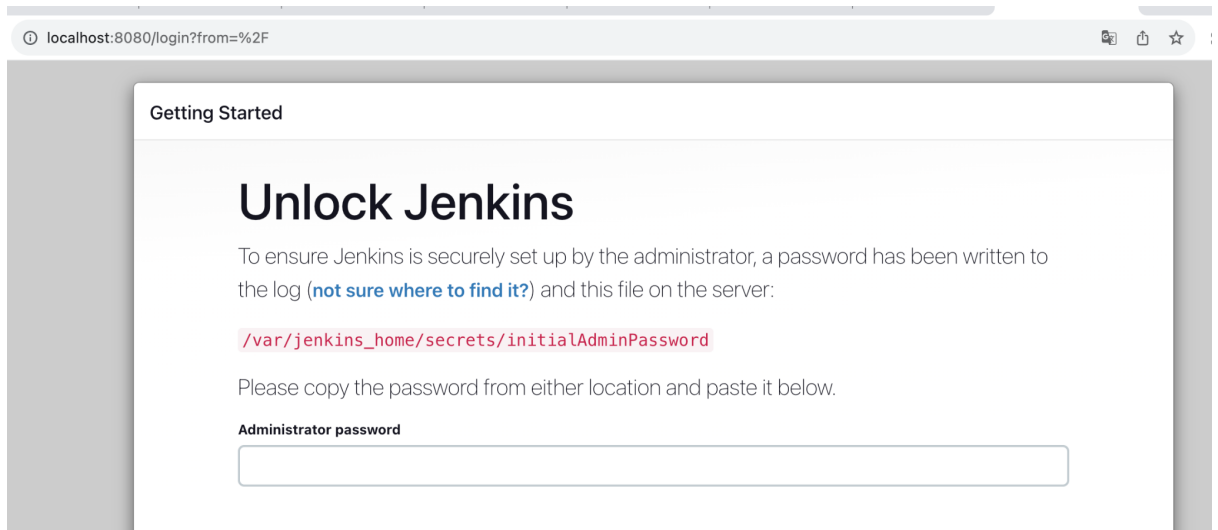
```
pedrofernandez@MacBook-Air-de-Pedro 07-servidor-build % docker build -t jenkins-with-dotnetcore -f Dockerfile.jenkins .
[+] Building 272.2s (5/8)
=> [internal] load .dockerignore
=> [internal] transferring context: 2B
=> [internal] load build definition from Dockerfile.jenkins
=> [internal] transferring dockerfile: 552B
=> [internal] load metadata for docker.io/jenkins/jenkins:lts
=> [auth] jenkins/jenkins:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/jenkins/jenkins:lts@sha256:b705323eaf70a7da4c1eed8b816f33dff2d5c8c3671170a2c17cf77aa4f15432
=> resolve docker.io/jenkins/jenkins:lts@sha256:b705323eaf70a7da4c1eed8b816f33dff2d5c8c3671170a2c17cf77aa4f15432
=> sha256:a62d9df1294b31821a59f46b02c70b7a2c6c5f4e0dcf21d1ec6645103b33a1ba 2.77kB / 2.77kB
=> sha256:1109a203edafbb2150eabe1c8b773146a671d02e96b737c50bc309c118ae201 12.62kB / 12.62kB
=> sha256:3a1383ba127ef9c1b2fef517f9445f101dcc10347eba2be32a018ed88d128064 61.08MB / 61.08MB
=> sha256:fa34261389f3dee2ecffe9bfe38ae7fd05a1908e98a7486354905aee0f648a6e 49.59MB / 49.59MB
```

- Ejecuta un Contenedor con la Nueva Imagen. Ahora, puedes ejecutar un contenedor utilizando la imagen que acabas de crear:

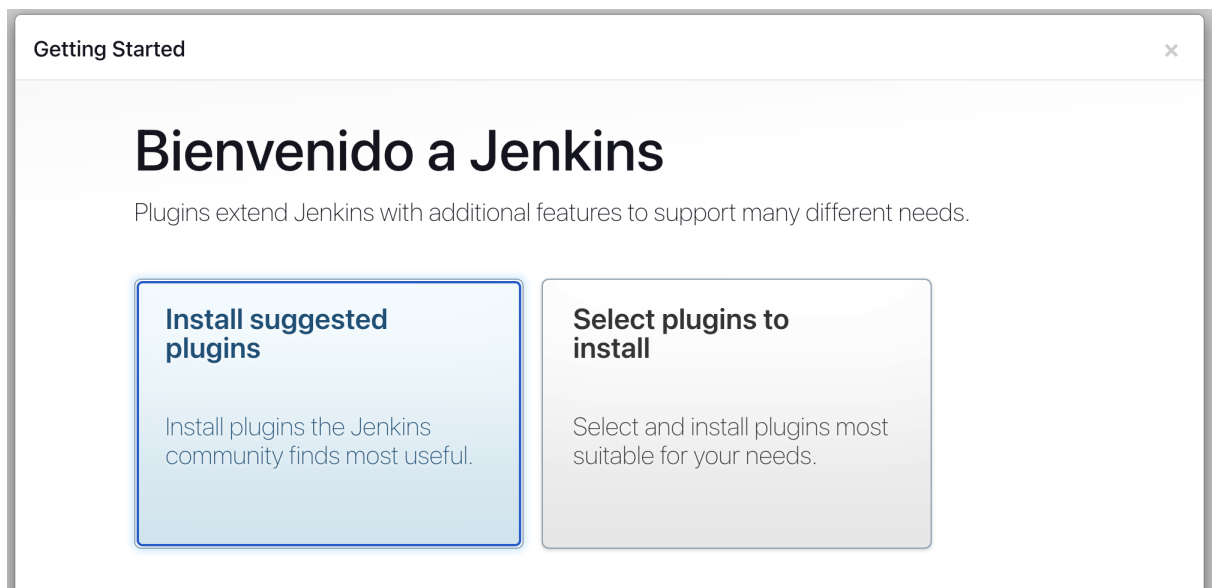
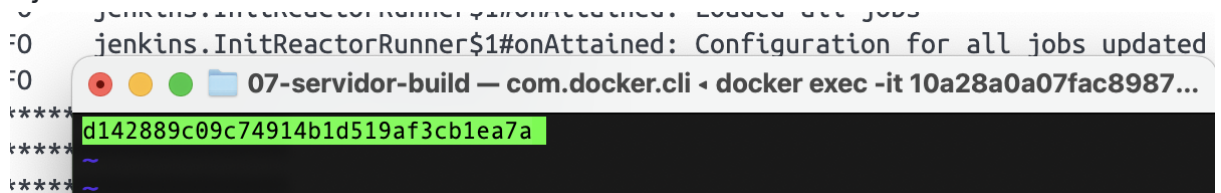
```
What's Next?
View summary of image vulnerabilities and recommendations → docker scout quick view
pedrofernandez@MacBook-Air-de-Pedro 07-servidor-build % mkdir -p ~/jenkins
docker run -d -p 8080:8080 -p 50000:50000 --name jenkins \
-v jenkins_home:/var/jenkins_home \
jenkins-with-dotnetcore

6449243f150713189e6a7eec674a11c49ee8dda8453b15a99e828b0483ca2792
pedrofernandez@MacBook-Air-de-Pedro 07-servidor-build %
```

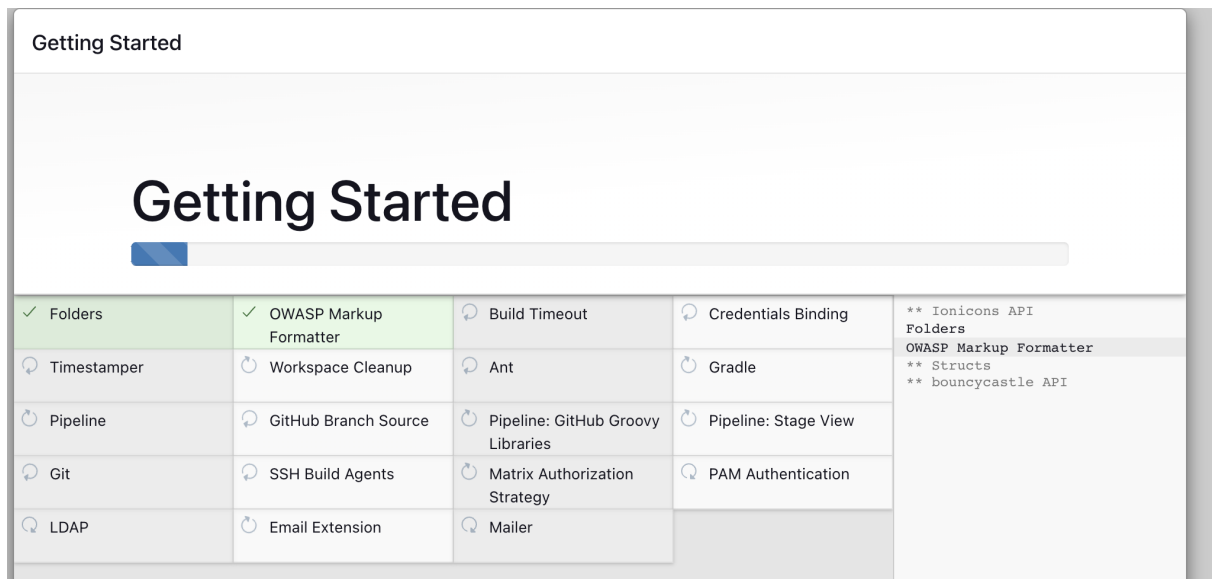
- Una vez en ejecución, abrir <http://localhost:8080>



- Inicialmente deberá especificar el texto que se encuentra dentro del archivo `~/.jenkins/secrets/initialAdminPassword` en el contenedor



- Instalar los plugins por defecto



- Crear el usuario admin inicial. Colocar cualquier valor que considere adecuado.

Getting Started

Usuario

piterfmz

Contraseña

.....

Confirma la contraseña

.....

Nombre completo

Pedro Fernandez

Dirección de email

2009636@ucc.edu.ar

2- Instalando Plugins y configurando herramientas

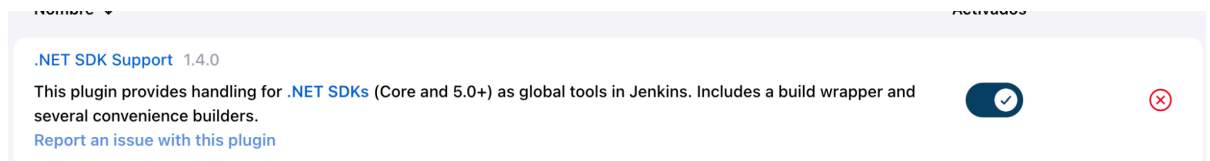
- En Administrar Jenkins vamos a la sección de Administrar Plugins
- De la lista de plugins disponibles instalamos .NET SDK Support

Plugins

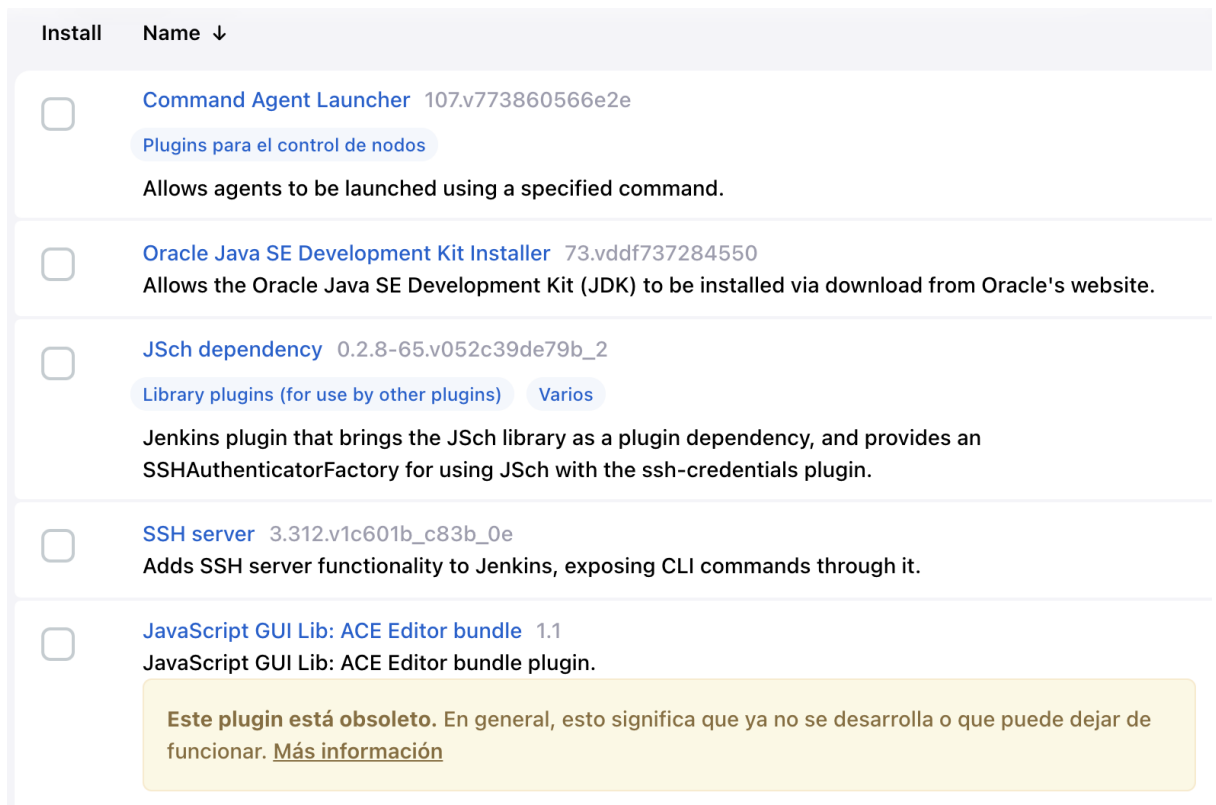
Q .NET SDK

Install	Name ↓	Relea
<input type="checkbox"/>	.NET SDK Support 1.4.0 .NET Development This plugin provides handling for .NET SDKs (Core and 5.0+) as global tools in Jenkins. Includes a build wrapper and several convenience builders.	Hace

- Reiniciamos el servidor

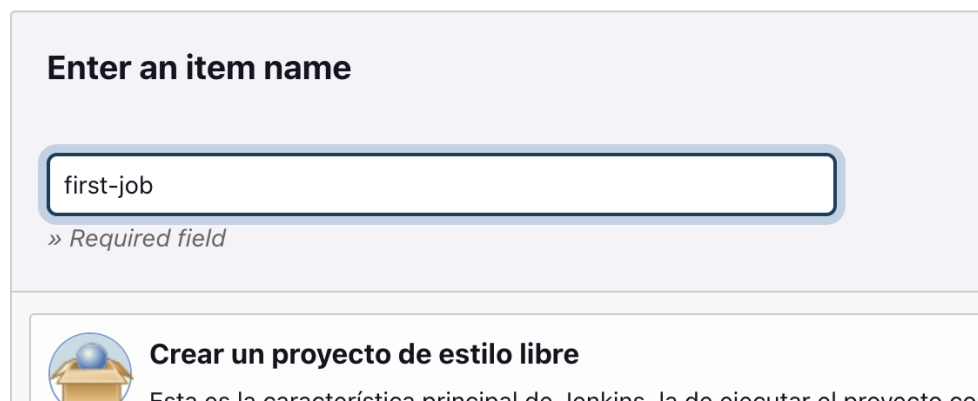


- Abrir nuevamente página de Plugins y explorar la lista, para familiarizarse qué tipo de plugins hay disponibles.



3- Creando el primer Job de estilo libre

- Crear un nuevo item, del tipo estilo libre con nombre first-job



- Una vez creado el job, en la sección Build Steps seleccionamos Ejecutar línea de comandos (shell) y escribimos:

Build Steps

Añadir un nuevo paso ▲

Filter

Ejecutar Ant

Ejecutar línea de comandos (shell)

Ejecutar tareas 'maven' de nivel superior

Ejecutar un comando de Windows

Invoke Gradle script

Run with timeout

Set build status to "pending" on GitHub commit

Build Steps

Ejecutar línea de comandos (shell) ?

Comando

Visualizar la lista de variables de entorno disponibles

```
# Obtén la fecha y hora actual en formato personalizado
current_datetime=$(date +"%Y-%m-%d %H:%M:%S")

# Imprime la fecha y hora actual utilizando el comando echo
echo "La fecha y hora actual es: $current_datetime"
```

Avanzado ▼

```
# Obtén la fecha y hora actual en formato personalizado
current_datetime=$(date +"%Y-%m-%d %H:%M:%S")
```

```
# Imprime la fecha y hora actual utilizando el comando echo
echo "La fecha y hora actual es: $current_datetime"
```

- Guardamos y ejecutamos el Job
- Analizar la salida del mismo

✓ Salida de consola

```
Started by user Pedro Fernandez
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/first-job
[first-job] $ /bin/sh -xe /tmp/jenkins8995177593168907928.sh
+ date +%Y-%m-%d %H:%M:%S
+ current_datetime=2023-09-29 15:33:23
+ echo La fecha y hora actual es: 2023-09-29 15:33:23
La fecha y hora actual es: 2023-09-29 15:33:23
Finished: SUCCESS
```

4- Creando el primer Pipeline Job

- Crear un nuevo ítem, del tipo Pipeline con nombre hello-world
- Una vez creado el job, en la sección Pipeline seleccionamos try sample Pipeline y luego Hello World

Pipeline

Definition

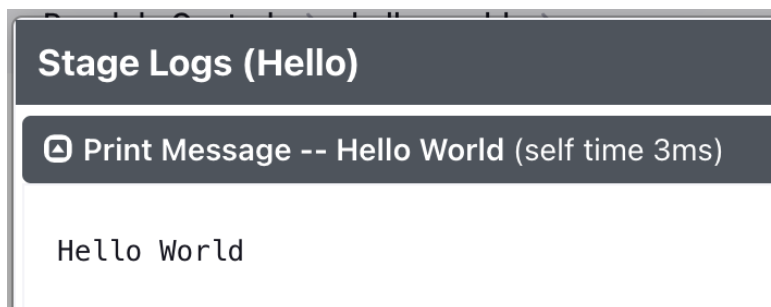
Pipeline script

Script ?

```
1 pipeline {  
2   agent any  
3  
4   stages {  
5     stage('Hello') {  
6       steps {  
7         echo 'Hello World'  
8       }  
9     }  
10  }  
11 }  
12
```

Hello World

- Guardamos y ejecutamos el Job
- Analizar la salida del mismo



5- Creando un Pipeline Job con Git

- Similar al paso anterior creamos un ítem con el nombre github-job
- En script escribir:

Pipeline

Definition

Pipeline script

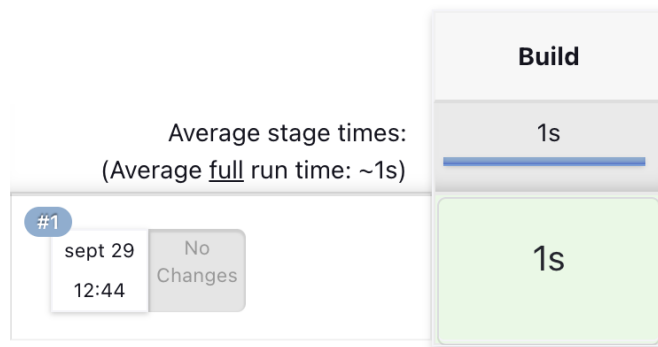
Script ?

```
1 pipeline {  
2   agent any  
3  
4  
5   stages {  
6     stage('Build') {  
7       steps {  
8         // Get some code from a GitHub repository  
9         git branch: 'main', url: 'https://github.com/ingsoft3ucc/SimpleWebAPI'  
10      }  
11    }  
12  }  
13 }  
14  
15
```

try sample Pipeline...

- Guardar y ejecutar el Job

Stage View



Enlaces permanentes

6- Utilizando nuestros proyectos.

6.1- .NET Core

- Crear un Job de estilo libre llamado "git-netcore-job" que construya el proyecto en .NET Core del ejercicio 1 del TP 05.
- Configurar el Job para obtener el código desde el repositorio de cada alumno:

Configurar el origen del código fuente

☐ Ninguno

☒ Git ?

Repositories ?

Repository URL ?

Credentials ?

Avanzado ▾

- Agregar las credenciales correspondientes:

Credentials ?

Avanzado ▾

Domain

Global credentials (unrestricted)

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

piterfmz

☐ Treat username as secret ?

Password ?

.....

- Indicar la rama desde la cual se construira:

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

- Configurar las etapas de Build y Publish del proyecto

Build Steps

.NET: Build project (build) ?

.NET SDK ?
(Default)

Project or Solution ?
SimpleWebAPI.sln

Targets ?

Configuration ?

Work Directory ?

☐ Continue On Error ?
☐ Errors Result in Unstable Build ?
☐ Warnings Result in Unstable Build ?
More Options

.NET: Publish project (publish) ?

.NET SDK ?
(Default)

Project or Solution ?
SimpleWebAPI.sln

Configuration ?

Work Directory ?

☐ Continue On Error ?
☐ Errors Result in Unstable Build ?
☐ Warnings Result in Unstable Build ?
More Options

- Corremos el Job y revisamos la salida.

6.2- Monitorear Cambios en el repo de GitHub

- Configurar Poll cada 1 minuto

☒ Consultar repositorio (SCM) ?

Programador ?

⚠ Do you really mean "every minute" when you say "*****"? Perhaps you meant "H*****" to poll once per hour

Would last have run at Friday, September 29, 2023 at 4:57:39 PM Coordinated Universal Time; would next run at Friday, September 29, 2023 at 4:57:39 PM Coordinated Universal Time.

[\(mostrar detalles\)](#)

[\(mostrar detalles\)](#)

☐ Ignore post-commit hooks ?

- Verificar su funcionamiento al hacer un commit y cuando no se hizo commit
- Explicar por que no es posible tener un WebHook en nuestro TP

En algunos entornos, como redes corporativas o aulas, puede ser difícil o imposible configurar WebHooks debido a restricciones de firewall, proxies, o configuraciones de red que impiden que GitHub se comunique directamente con el servidor de Jenkins. En tales casos, la configuración de "Poll SCM" es una alternativa, aunque menos eficiente.

- Aplicar Diferencia con ejecutar periódicamente:

Disparadores de ejecuciones

☐ Lanzar ejecuciones remotas (ejem: desde 'scripts')

☐ Construir tras otros proyectos ?

☒ Ejecutar periódicamente ?

Programador ?
