

Práctica 2 - exploratoria

En esta práctica realizaremos la instalación de Docker mediante el aprovisionamiento que nos brinda Vagrant, software el cual hemos instalado y configurado en la práctica 1. Para esto reutilizaremos la imagen de Ubuntu que hemos creado en VirtualBox y también parte de su configuración para lograr mediante poner una aplicación en contenedores, en la que tendremos dos de ellos. Uno corresponderá a un servidor web y la otra corresponderá a una imagen de servidor de base de datos. En este caso los alumnos podrán elegir la tecnología que les sea más familiar para el servidor, la base de datos y el lenguaje de programación para la aplicación, considerar que luego deberán escribir al menos una prueba unitaria (UnitTest).

Los contenedores Docker no persisten datos, cualquier cambio generado en el contenedor ya sea modificar algún archivo o generan nuevos, los cambios se perderán si se reinicia el servicio o se reconstruye la imagen. Para dar una solución a esto Docker nos permite persistir los datos generados en el contenedor por fuera del mismo mediante el montado de un volumen o un directorio de la máquina host (en nuestro caso es la imagen Ubuntu en Virtualbox) con un directorio del contenedor. Esto se puede realizar al iniciar un contenedor o bien especificarlo en un archivo YAML si utiliza docker compose (docker-compose.yml).

Los contenedores de Docker nos permiten tener una infraestructura inmutable dado que no hay persistencia de datos, sin embargo si deseamos modificar alguna configuración específica, nos permite realizarlo de distintos modos. El alcance de este curso no cubre estos aspectos pero para que se tenga conocimiento de estos métodos los mencionaremos, los comandos que permiten realizar estas variantes son: “docker export” para exportar el contenido de un contenedor (sin incluir los volúmenes), “docker save” para guardar los datos de una nueva imagen y “docker commit” para crear una versión nueva de la imagen a partir del estado actual de un contenedor (esto permite observar las diferencias entre imágenes tal como si fuera el código versionado de una aplicación). A su vez, la versatilidad que nos brinda el uso de las distintas imágenes se puede amplificar mediante el uso de Docker Hub. Lo que nos permite esta herramienta es subir nuestras propias imágenes a un repositorio en la nube, tal como si fuera un repositorio de una aplicación y permitir descargarlas y mantener el versionado desde un lugar centralizado.

Importante

Para cada unidad encontrarán una guía detallada describiendo cada paso necesario para completar las prácticas. Su objetivo es guiar con ejemplos a los alumnos que se encuentren bloqueados con alguna consigna. Así mismo, cada alumno puede tener distintas o ninguna experiencia en el conjunto tecnológico que se brinda en este curso, por lo cual alentamos a que las prácticas se realicen en grupo y que al menos haya algún integrante que cuente con experiencia en desarrollo u operaciones.

También invitamos a los alumnos a que primero intenten resolver las consignas sin antes revisar la guía detallada y en los casos en que alguna tecnología sea de libre elección (esto se aclarará en cada consigna) opten por la que se sientan más cómodos. Por ejemplo, en la creación de una aplicación en lugar de ser en lenguaje PHP podría ser en React, Node.js, Go, etc. siempre y cuando cumpla con el objetivo de la consigna.

La realización de esta práctica requiere conocimientos básicos de linux, git, docker y redes. En caso de no poseer dichos conocimientos recomendamos que utilicen la guía detallada de esta unidad para poder probar los conceptos y herramientas.

Pre requisitos:

Conocimientos en:

- Linux
- Git
- Docker
- Redes

Consigna

El objetivo de esta práctica es generar dos contenedores en Docker, los cuales uno tendrá el servidor web y la instalación de las dependencias necesarias para ejecutar la aplicación y el segundo tendrá una base de datos. Se debe poder realizar una petición al puerto 8080 desde la máquina host y los contenedores deben devolver la petición al usuario con algún contenido de la base de datos

Se deberá completar lo siguiente:

- Los archivos de infraestructura deben estar versionados en un branch con el nombre “unidad-2-docker”, se debe crear a partir de los últimos cambios del branch “unidad-1-vagant”. Proyecto “utn-devops”.
- El script de aprovisionamiento de Vagrant debe realizar lo siguiente:
 - Desinstalar el software instalado de la práctica anterior.

- Instalar Docker.
- Descargar el código de la aplicación dentro de Ubuntu para que el servicio web lo presente.
- Crear un contenedor con un servidor web y que dentro tenga mapeado el código de la aplicación que debe mostrar. Este contenedor debe poder acceder a la base de datos.
 - Para los efectos de la práctica, **no interesa** en absoluto la calidad del código.
- Crear un contenedor con una base de datos y almacenar algún tipo de contenido de manera persistente (utilizar volúmenes).

Entregables:

- URLs de los 2 repositorios git.
 - Para el repositorio de la aplicación indicar branch y una breve explicación de qué dato se consumirá en la base de datos. Esto se utilizará para hacer una coincidencia con la captura del navegador del punto siguiente.
- Captura de pantalla del navegador consumiendo la aplicación dentro la VM de Ubuntu.
- Presentar los entregables en único archivo con el nombre: NOMBRE EQUIPO - Practica 2. **Sólo una entrega por equipo.**