



**El futuro digital  
es de todos**

MinTIC



CONFIGURAR ESCENARIO DE INTEROPERABILIDAD MEDIANTE X-ROAD

## **Escenario Básico de Interoperabilidad**

**3**

**Guía de Instalación  
de docker**



# Introducción

Docker es un proyecto de código abierto, cuya plataforma permite el despliegue de diferentes tipos de componentes de software dentro de contenedores. Para mayor información acerca de Docker, consultar su [página web oficial](https://www.docker.com/). (<https://www.docker.com/>)

A continuación se presentan tres guías de instalación de Docker, de acuerdo al sistema operativo: Ubuntu, Windows, macOS.

## Instalación en Ubuntu

**Nota:** esta sección presenta la guía para la instalación de Docker en el sistema operativo Ubuntu. Si se desea instalar en otra distribución de Linux, por favor consultar la [documentación oficial de instalación de Docker](https://docs.docker.com/engine/install/). (<https://docs.docker.com/engine/install/>)

### Prerrequisitos

Docker es compatible con arquitecturas x86\_64 (o amd64), armhf y arm64 con alguna de las siguientes versiones de Ubuntu:

- Ubuntu Focal 20.04 (LTS)
- Ubuntu Bionic 18.04 (LTS)
- Ubuntu Xenial 16.04 (LTS)

### Instalación de Docker

1. Iniciar la terminal.
2. Antes de iniciar la instalación de Docker, remover cualquier instalación previa. Para esto, ejecutar el siguiente comando:
  3. `$ sudo apt-get remove docker docker-engine docker.io containerd runc`
4. Actualizar el índice de paquetes de **apt** e instalar los controladores que permitirán realizar la descarga por HTTPS.

```
$ sudo apt-get update
$ sudo apt-get install -y \
    apt-transport-https \
    ca-certificates \
    curl \
```



```
gnupg-agent \
software-properties-common
```

- Añadir la llave GPG oficial de Docker y verificar la huella de la llave.

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

```
$ sudo apt-key fingerprint 0EBFCD88
```

Se debería obtener la siguiente huella como resultado

```
9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
```

```
docker@x-road:~$ sudo apt-key fingerprint 0EBFCD88
pub    rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid          [desconocida] Docker Release (CE deb) <docker@docker.com>
sub    rsa4096 2017-02-22 [S]
```

- Ejecutar el siguiente comando para consultar la arquitectura del sistema operativo Ubuntu:

```
$ uname -m
```

Como resultado se debe obtener *X86\_64*, *amd64*, *armhf* o *arm64*

```
docker@x-road:~$ uname -m
x86_64
```

- Añadir el repositorio estable de docker a apt.

Tener en cuenta que según la arquitectura obtenida en el paso anterior, se deberá ejecutar el comando correspondiente:

Para arquitecturas *X86\_64* y *amd64*:

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

Para arquitecturas *armhf*:

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



Para arquitecturas *arm64*:

```
$ sudo add-apt-repository \
  "deb [arch=arm64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

8. Instalar Docker Engine:

```
$ sudo apt-get update -y
$ sudo apt-get install -y docker-ce docker-ce-cli containerd.io
```

9. Añadir los privilegios Docker para el usuario actual de la terminal:

```
$ sudo usermod -aG docker $USER
```

10. Verificar que la instalación de Docker haya sido satisfactoria:

```
$ docker version
```

El resultado debe ser similar a:

```
docker@x-road:~$ docker version
Client: Docker Engine - Community
  Version:           19.03.13
  API version:      1.40
  Go version:       go1.13.15
  Git commit:       4484c46d9d
  Built:            Wed Sep 16 17:02:52 2020
  OS/Arch:          linux/amd64
  Experimental:    false

Server: Docker Engine - Community
  Engine:
    Version:          19.03.13
    API version:     1.40 (minimum version 1.12)
    Go version:      go1.13.15
    Git commit:      4484c46d9d
    Built:           Wed Sep 16 17:01:20 2020
    OS/Arch:         linux/amd64
    Experimental:   false
  containerd:
    Version:          1.3.7
    GitCommit:        8fb4e9a7d01810a393d5d25a3621dc101981175
  runc:
    Version:          1.0.0-rc10
    GitCommit:        dc9208a3303feef5b3839f4323d9beb36df0a9dd
  docker-init:
    Version:          0.18.0
    GitCommit:        fec3683
```

11. Instalar Docker Compose:

```
$ sudo curl -L  
"https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname  
-s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

12. Otorgar permisos de ejecución a Docker Compose:

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

13. Verificar que la instalación de Docker Compose haya sido satisfactoria:

```
$ docker-compose -version
```

El resultado debe ser similar a:

```
docker@x-road:~$ docker-compose -version  
docker-compose version 1.27.4, build 40524192
```

# Instalación en Windows

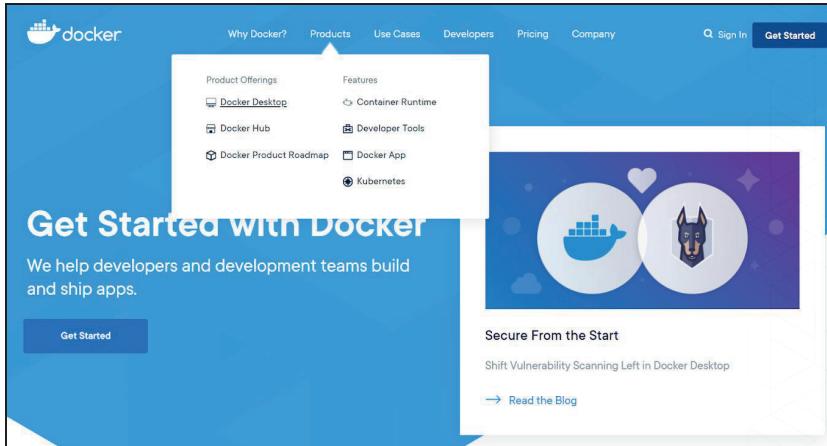
## Prerrequisitos

Para la instalación de Docker en el sistema operativo **Microsoft Windows** hay que considerar los siguientes requerimientos:

- Sistema operativo **Windows 10** de **64-bit** en las versiones:
  - Professional
  - Enterprise
  - Education (Build 16299 or later)
- Se deben habilitar las características de *virtualización* y *contenedores* en el sistema operativo.
- Se debe contar con las siguientes especificaciones de software para que el cliente de virtualización se desempeñe correctamente:
  - Procesador de 64 bits con traducción de direcciones de segundo nivel (SLAT).
  - Memoria RAM de 4GB.
  - En la configuración de la BIOS debe habilitarse la opción de soporte de virtualización de hardware.

## Instalación de Docker

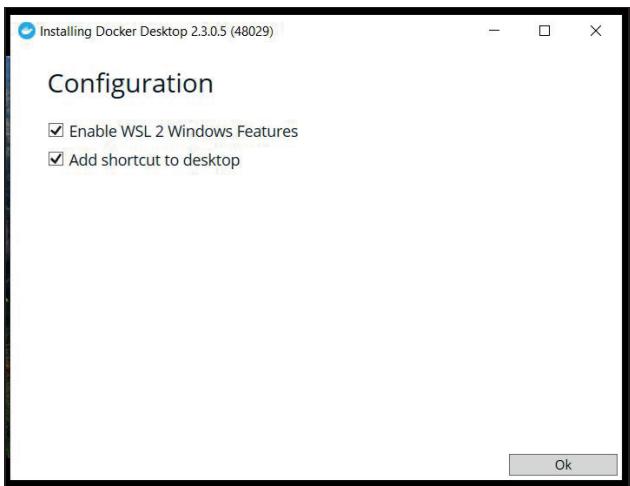
1. Ingresar a la página principal de Docker e ir a la sección de *Productos* para descargar la versión de escritorio: <https://www.docker.com/products/docker-desktop>



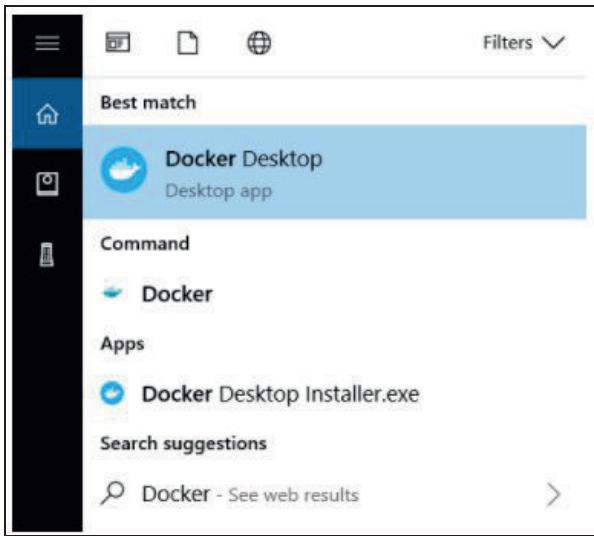
2. Descargar el ejecutable de la versión Stable de Windows:



3. Asegurarse de que las características de Hiper-Virtualización de Windows estén habilitadas en el sistema operativo.
4. Una vez descargado el instalador, ejecutarlo en el computador haciendo clic sobre el instalador y continuar los pasos que brinda el asistente para su instalación.



5. Cuando la instalación se haya terminado, hacer clic en *Cerrar* para completar el proceso de instalación.
6. Para iniciar Docker Desktop, buscar Docker en el buscador de Windows y seleccionar Docker Desktop en los resultados de la búsqueda.



7. Verificar que el ícono de la ballena está activo, con el mensaje de *Docker Desktop is running...*, lo que significa que Docker ya está corriendo y puede ser accesible desde cualquier terminal.
8. Abrir una terminal (cmd o símbolo del sistema) y ejecutar el comando **docker**. Allí deberá salir una lista de opciones que servirán de ayuda durante el uso de la plataforma.



```
C:\Users\ME>docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "C:\\Users\\ME\\.docker")
  -c, --context string Name of the context to use to connect to the
                        daemon (overrides DOCKER_HOST env var and
                        default context set with "docker context use")
  -D, --debug          Enable debug mode
  -H, --host list      Daemon socket(s) to connect to
  -l, --log-level string
                        Set the logging level
                        ("debug"|"info"|"warn"|"error"|"fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string   Trust certs signed only by this CA (default "C:\\Users\\ME\\.docker\\ca.pem")
  --tlscert string     Path to TLS certificate file (default "C:\\Users\\ME\\.docker\\cert.pem")
  --tlskey string       Path to TLS key file (default "C:\\Users\\ME\\.docker\\key.pem")
  --tlsverify          Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  builder    Manage builds
  config     Manage Docker configs
  container  Manage containers
  context    Manage contexts
  image      Manage images
```

- Ejecutar dentro de la consola de comandos, la instrucción **docker --version** para revisar que Docker fue instalado correctamente y que se tiene una versión funcional de la herramienta.

```
Microsoft Windows [Versión 10.0.19041.508]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\ME>docker --version
Docker version 19.03.12, build 48a66213fe

C:\Users\ME>
```

- Ejecutar el comando **docker-compose --version** para verificar que el paquete docker-compose haya sido instalado correctamente.

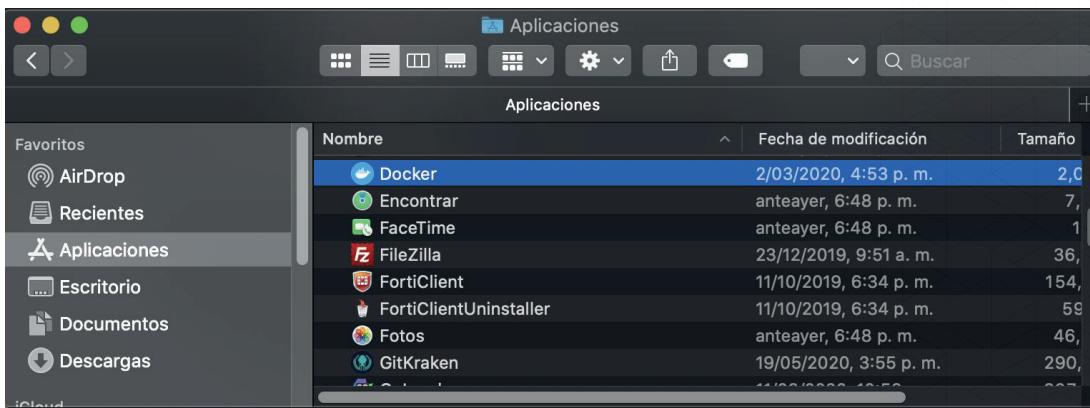


```
C:\Users\ME>docker-compose --version
docker-compose version 1.27.2, build 18f557f9
C:\Users\ME>
```

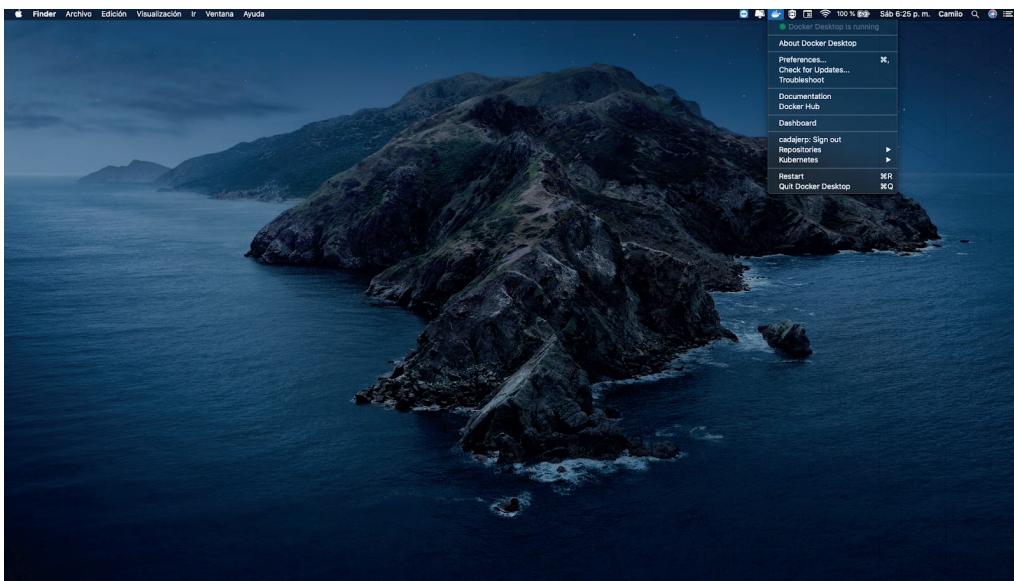
# Instalación en macOS

## Instalación de Docker

1. Ingresar a la siguiente dirección y descargar Docker Desktop versión [Stable](#):  
<https://www.docker.com/products/docker-desktop>
2. Una vez descargado, ejecutar el instalador en el computador y continuar con los pasos de instalación.
3. Buscar el programa instalado en el programa *Finder* de su equipo y abrirlo:



4. Localizar el ícono de Docker en la barra superior derecha:



5. Esperar hasta que Docker cambie su estado de “*docker is starting*” a “*docker is running*”.
6. Abrir la *Terminal*:
7. Ejecutar el comando “`docker`” y verificar que aparezca la siguiente información:

```

MacBook-Pro-de-mac:~ mac$ docker
WARNING: Error loading config file: /Users/mac/.docker/config.json: EOF

Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "/Users/mac/.docker")
  -c, --context string Name of the context to use to connect to the daemon (overrides
                        DOCKER_HOST env var and default context set with "docker context use")
  -D, --debug          Enable debug mode
  -H, --host list      Daemon socket(s) to connect to
  -l, --log-level string Set the logging level ("debug"|"info"|"warn"|"error"|"fatal") (default "info")
  --tls                Use TLS; implied by --tlsv1.2
  --tlscacert string   Trust certs signed only by this CA (default "/Users/mac/.docker/ca.pem")
  --tlscert string     Path to TLS certificate file (default "/Users/mac/.docker/cert.pem")
  --tlskey string       Path to TLS key file (default "/Users/mac/.docker/key.pem")
  --tlsv1.2              Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  builder    Manage builds
  config     Manage Docker configs
  container  Manage containers
  context    Manage contexts
  image      Manage images
  network   Manage networks
  node       Manage Swarm nodes
  plugin    Manage plugins

```

8. Ejecutar el comando “`docker ps`” y verificar que aparezca la siguiente información:



```
MacBook-Pro-de-mac:~ mac$ docker ps
WARNING: Error loading config file: /Users/mac/.docker/config.json: EOF
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
 NAMES
MacBook-Pro-de-mac:~ mac$
```

9. Ejecutar el comando “docker-compose” y verificar que aparezca la siguiente información:

```
MacBook-Pro-de-mac:~ mac$ docker-compose
Define and run multi-container applications with Docker.

Usage:
  docker-compose [-f <arg>...] [options] [COMMAND] [ARGS...]
  docker-compose -h|--help

Options:
  -f, --file FILE          Specify an alternate compose file
                           (default: docker-compose.yml)
  -p, --project-name NAME  Specify an alternate project name
                           (default: directory name)
  --verbose                Show more output
  --log-level LEVEL         Set log level (DEBUG, INFO, WARNING, ERROR, CRITICAL)
  --no-ansi                 Do not print ANSI control characters
  -v, --version              Print version and exit
  -H, --host HOST           Daemon socket to connect to

  --tls                     Use TLS; implied by --tlsverify
  --tlscacert CA_PATH       Trust certs signed only by this CA
  --tlscert CLIENT_CERT_PATH Path to TLS certificate file
  --tlskey TLS_KEY_PATH     Path to TLS key file
  --tlsverify               Use TLS and verify the remote
  --skip-hostname-check     Don't check the daemon's hostname against the
                           name specified in the client certificate
  --project-directory PATH   Specify an alternate working directory
                           (default: the path of the Compose file)
  --compatibility           If set, Compose will attempt to convert keys
                           in V3 files to their non-Swarm equivalent
  --env-file PATH           Specify an alternate environment file
```