Comandos para desplegar el aplicativo Ventas APP y Api Ventas

Tabla de contenido

[1. Creación de las máquinas virtuales 3](#_Toc18101050)

[2. Habilitar el modo swarn 3](#_Toc18101051)

[3. Docker compose Swarm 5](#_Toc18101052)

[4. Ventas APP 6](#_Toc18101053)

Comandos para levantar el aplicativo

# Creación de las máquinas virtuales

Manager 1:

docker-machine create --driver virtualbox --virtualbox-boot2docker-url=https://github.com/boot2docker/boot2docker/releases/download/v18.05.0-ce/boot2docker.iso manager1

Manager 2:

docker-machine create --driver virtualbox --virtualbox-boot2docker-url=https://github.com/boot2docker/boot2docker/releases/download/v18.05.0-ce/boot2docker.iso manager2

Worker 1:

docker-machine create --driver virtualbox --virtualbox-boot2docker-url=https://github.com/boot2docker/boot2docker/releases/download/v18.05.0-ce/boot2docker.iso worker1

Worker 2:

docker-machine create --driver virtualbox --virtualbox-boot2docker-url=https://github.com/boot2docker/boot2docker/releases/download/v18.05.0-ce/boot2docker.iso worker2

# Habilitar el modo swarn

1. En la máquina virtual **manager1** habilitar el modo swarn con el siguiente comando:

docker-machine ssh manager1 "docker swarm init --advertise-addr $(docker-machine ip manager1)"

1. Se debe mostrar un mensaje como el siguiente:

Swarm initialized: current node (tkj5gl66uvp2q0vg3oewuyrvd) is now a manager.

To add a worker to this swarm, run the following command:

docker swarm join --token SWMTKN-1-0brt0fkapt8dv1b95e5wuei7z56s7qdotd91hl6hts3fzfllhm-e69nrpb68pp6tavgo76q6zsfu 192.168.99.101:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

1. Consultamos el token para agregar otro manager, ejecutando el siguiente comando en el manager1, según el mensaje mostrado anteriormente:

docker-machine ssh manager1 “docker swarm join-token manager”

1. Se debe mostrar el token para agregar otros managers:

To add a manager to this swarm, run the following command:

docker swarm join --token SWMTKN-1-0brt0fkapt8dv1b95e5wuei7z56s7qdotd91hl6hts3fzfllhm-cf3qcso00mg7gui3j6qp5hvb4 192.168.99.101:2377

1. Para unir el Manager2 al Manager1 ejecutar el siguiente comando, teniendo en cuenta el comando del paso 4:

docker-machine ssh manager2 “docker swarm join --token SWMTKN-1-0brt0fkapt8dv1b95e5wuei7z56s7qdotd91hl6hts3fzfllhm-cf3qcso00mg7gui3j6qp5hvb4 192.168.99.101:2377”

con ello el Manager1 está unido al Manager2.

1. Para unir los workers 1 y 2 al Manager 1, ejecutar el siguiente comando, teniendo en cuenta el comando del paso 2:

docker-machine ssh worker1 “docker swarm join --token SWMTKN-1-0brt0fkapt8dv1b95e5wuei7z56s7qdotd91hl6hts3fzfllhm-e69nrpb68pp6tavgo76q6zsfu 192.168.99.101:2377”

docker-machine ssh worker2 “docker swarm join --token SWMTKN-1-0brt0fkapt8dv1b95e5wuei7z56s7qdotd91hl6hts3fzfllhm-e69nrpb68pp6tavgo76q6zsfu 192.168.99.101:2377”

1. Conectarse al Manager1 con el siguiente comando:

docker-machine env manager1

1. Esto mostrará el siguiente mensaje:

export DOCKER\_TLS\_VERIFY="1"

export DOCKER\_HOST="tcp://192.168.99.101:2376"

export DOCKER\_CERT\_PATH="C:\Users\usuario\.docker\machine\machines\manager1"

export DOCKER\_MACHINE\_NAME="manager1"

export COMPOSE\_CONVERT\_WINDOWS\_PATHS="true"

# Run this command to configure your shell:

# eval $("C:\Program Files\Docker Toolbox\docker-machine.exe" env manager1)

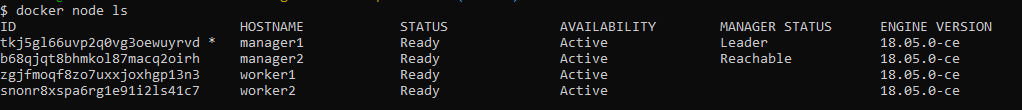
1. Ejecutar el comando que se muestra en el paso 8:

eval $("C:\Program Files\Docker Toolbox\docker-machine.exe" env manager1)

1. Para verificar como está creado y organizado los managers y los workers, ejecutar el siguiente comando:

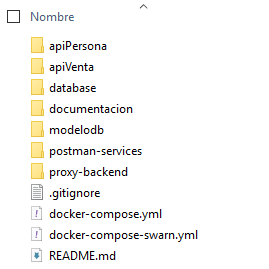
docker node ls

1. Se debe mostrar la siguiente información:



# Docker compose Swarm

1. Ubicarse en la carpeta raíz del proyecto **docker-api-ventas**. Debe figurar como se muestra en la imagen:



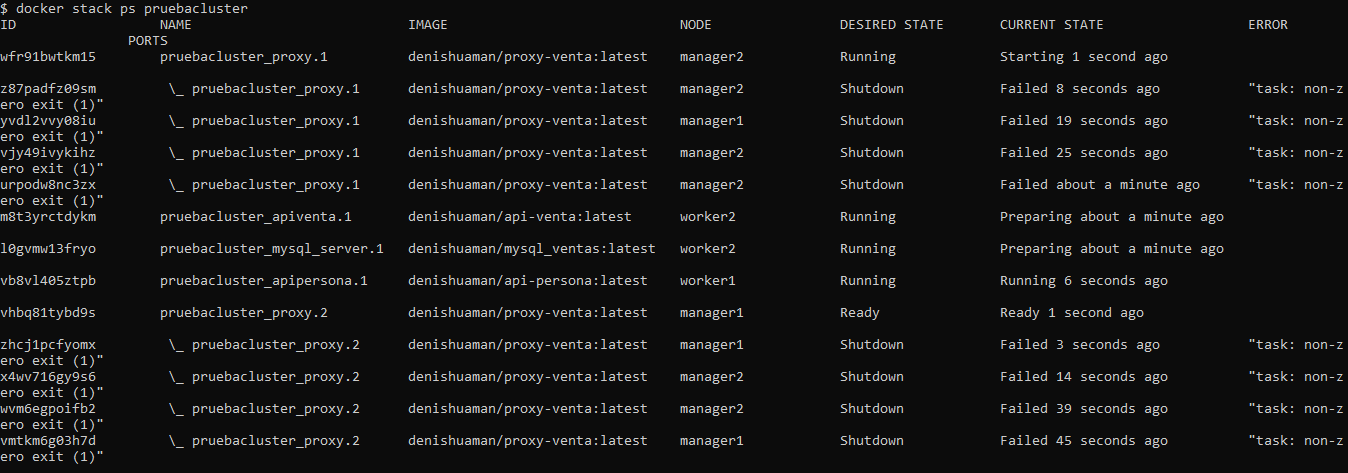
1. Ejecutar el siguiente comando:

docker stack deploy -c docker-compose-swarm.yml pruebacluster

1. Si se desea ver el progreso del despliegue, ejecutar el siguinte comado:

docker stack ps pruebacluster

se debe mostrar la siguiente salida en consola:



1. Para ver donde se está ejecutando los servicios, ejecutar el comando:

docker service ps pruebacluster\_postgresql\_server

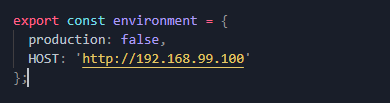
docker service ps pruebacluster\_apipersona

docker service ps pruebacluster\_apiventa

docker service ps pruebacluster\_proxy

# Ventas APP

1. Ubicarse en la ruta **..\docker-api-ventas\ventas-frontend\src\environments**. Abrir el archivo **environment.ts**.



1. Editar la variable HOST por la IP del host donde se encuentre instalado docker. Si se utiliza Docker Toolbox (Windows), generalmente la IP es **192.168.99.100**. Si se utiliza Docker for Windows, la IP es **localhost**. Si está instalado en Linux, será **localhost**. Como NGINX tiene dos replicas, entonces se debe cambiar la IP por las dos máquinas virtuales generadas (manager1 y manager2).
2. Ubicarse en la ruta **..\docker-api-ventas\ventas-frontend**, abrir la consola en esta ubicación y ejecutar el comando.

**ng serve --o**

1. Se debe mostrar la siguiente interfaz gráfica.

