Creación de las 5 instancias ec2:

Nombre de la instancia

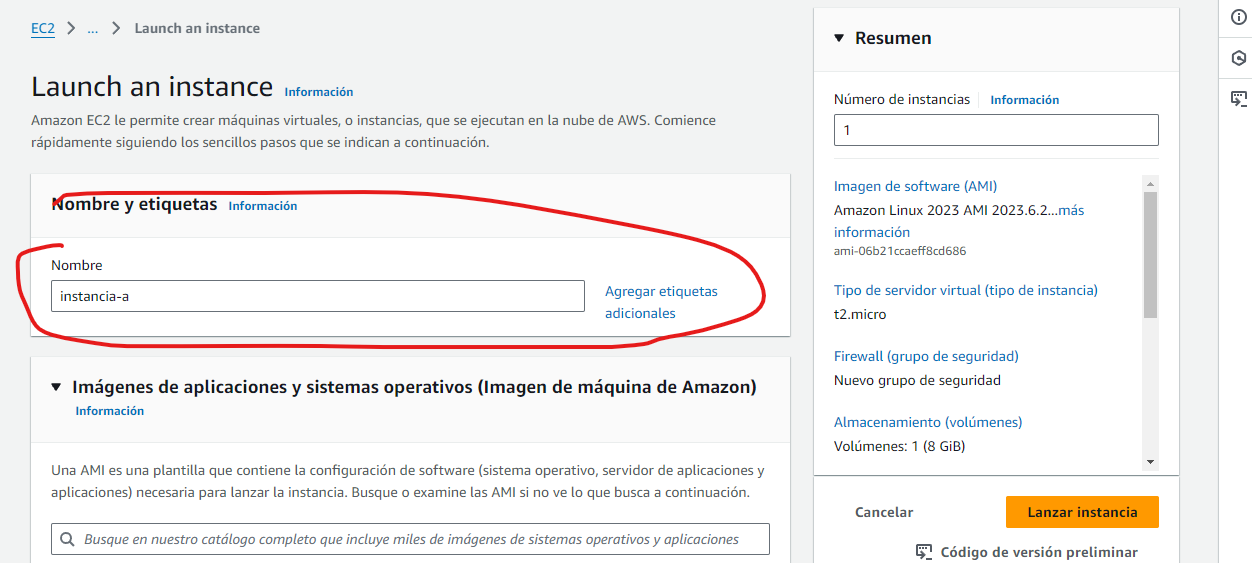
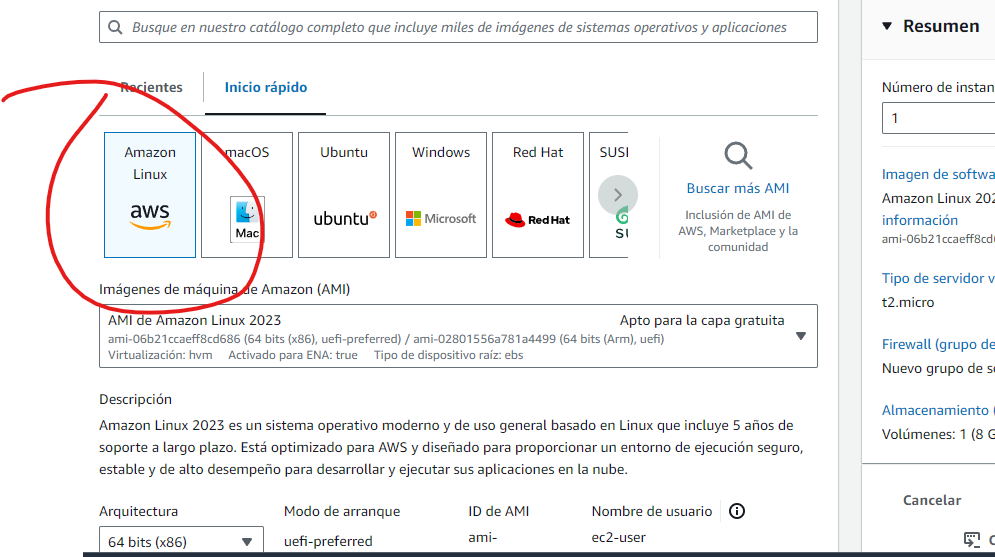
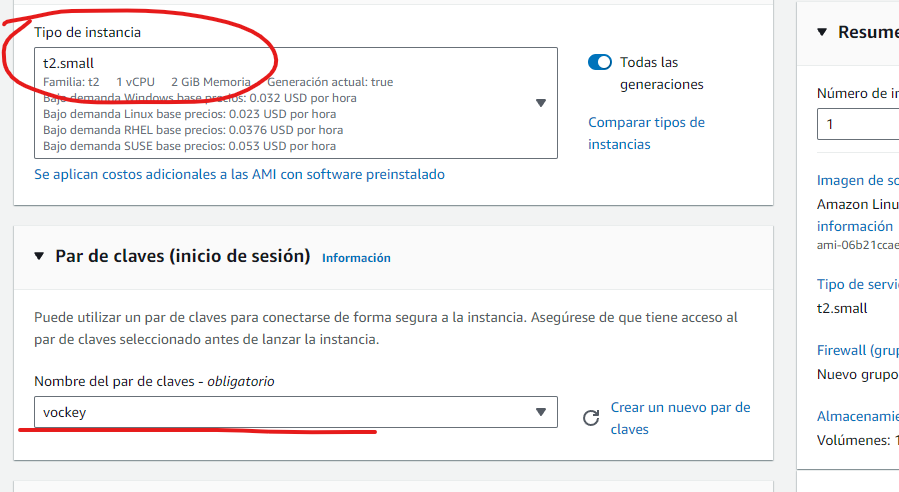
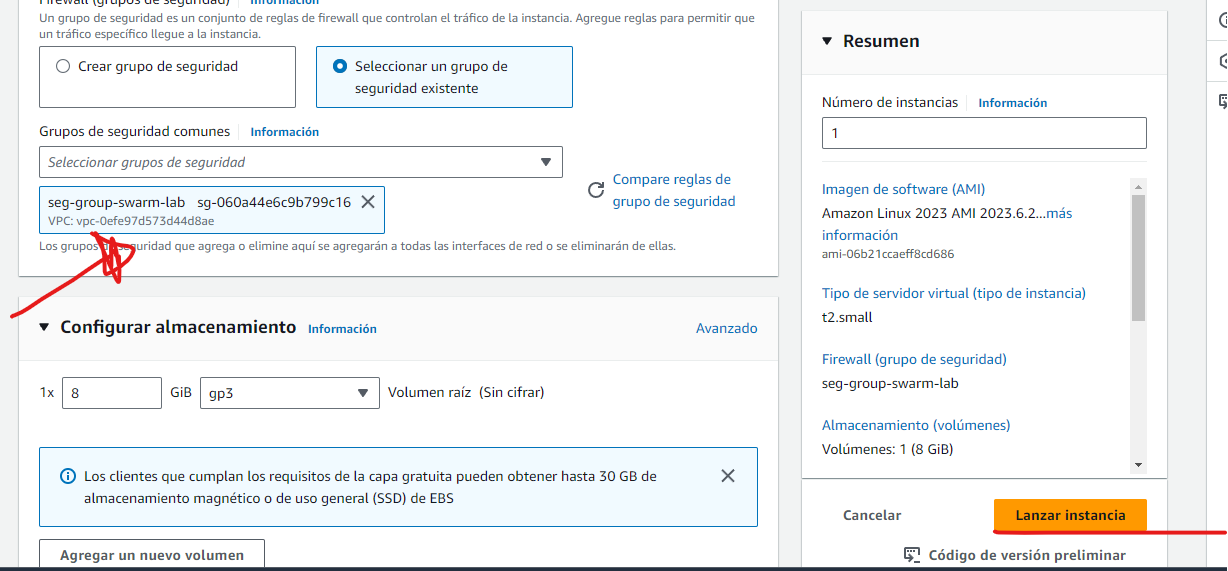
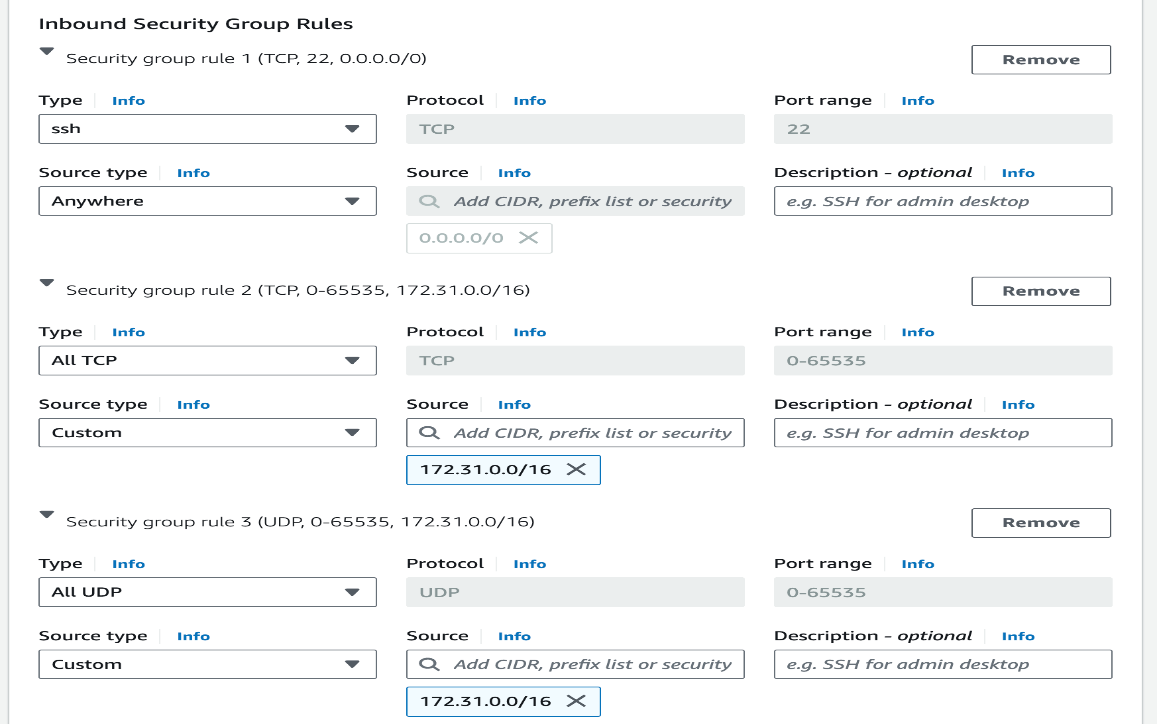


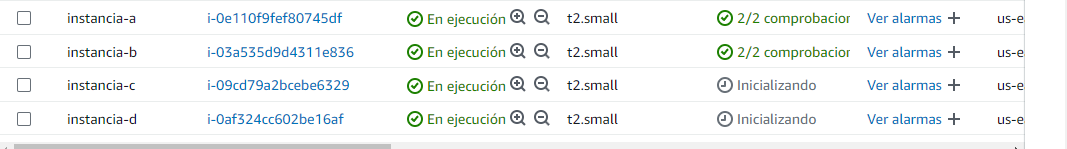
Imagen (sistema operativo)



Tipo de instancia y par de clave



Grupo de seguridad y crear instancia



* Instalar Docker en las instancias EC2 en cada una de las instancias o clona la instancia que ya tiene docker

**sudo yum update**

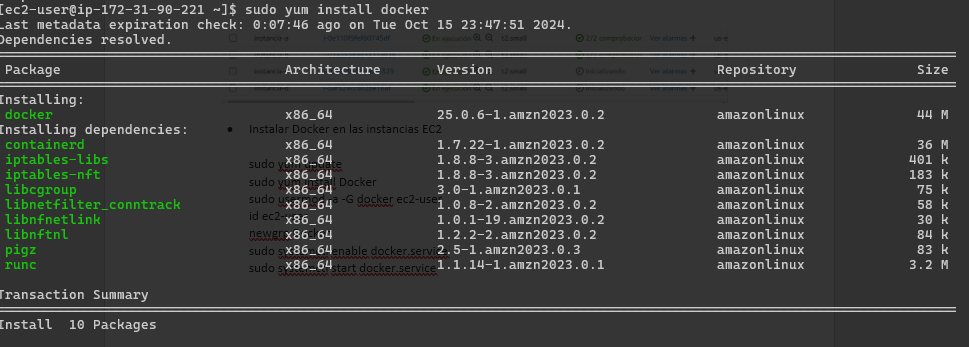
**sudo yum install docker**

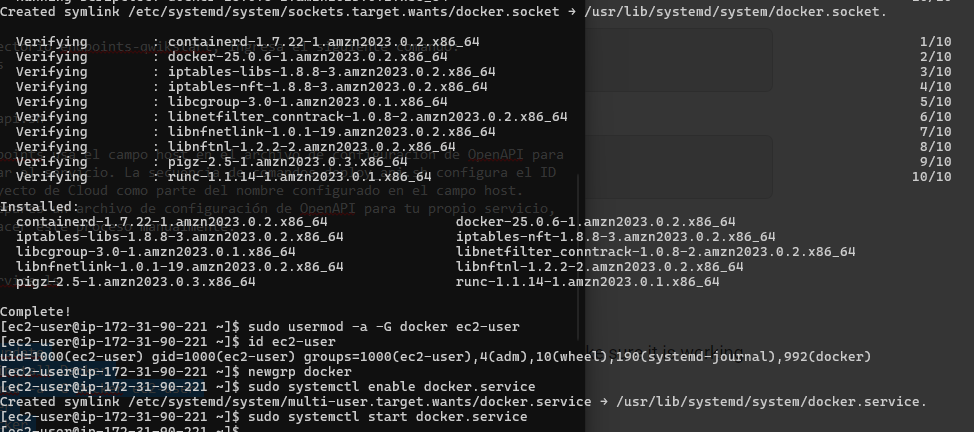
**sudo usermod -a -G docker ec2-user**

**id ec2-user**

**newgrp Docker**

**sudo systemctl enable docker.service**

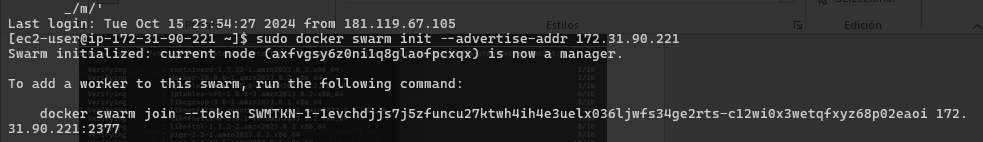
**sudo systemctl start docker.service**



Ahora creamos nuestro primer nodo manager (Leader)

Inicializa el clúster en Docker Swarm, usando la IP privada de la instancia:

**sudo docker swarm init --advertise-addr <manager-private-ip>**



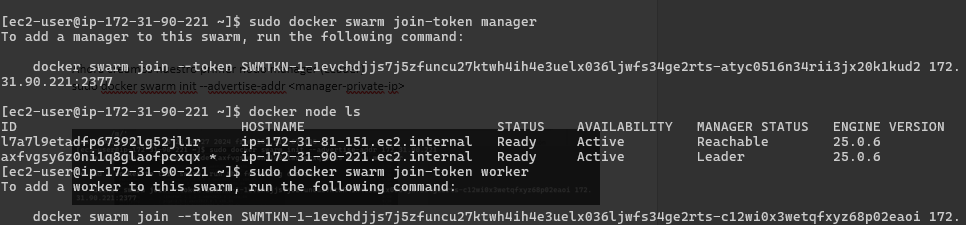
Esto creará el primer nodo manager del clúster. Recibirás un token de "join" para añadir otros nodos.

**Agregar el Segundo Nodo Manager**

Conéctate a la instancia que será el segundo nodo **manager**.

En el primer nodo **manager**, ejecuta el siguiente comando para obtener el token de manager:

**sudo docker swarm join-token manager**



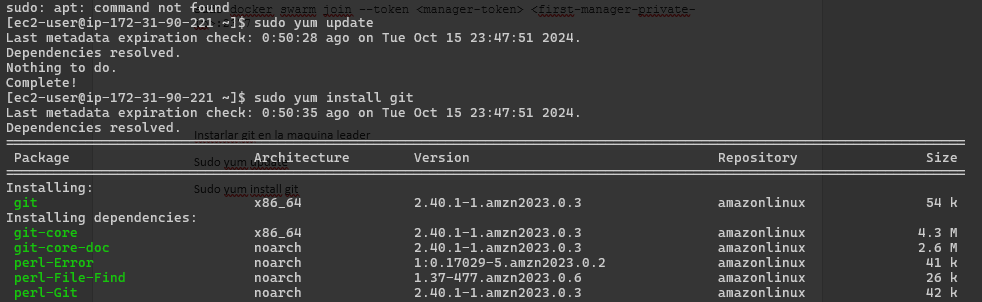
Copia el token generado y, en la segunda instancia **manager**, ejecuta el siguiente comando:

**sudo docker swarm join --token <manager-token> <first-manager-private-ip>:2377**

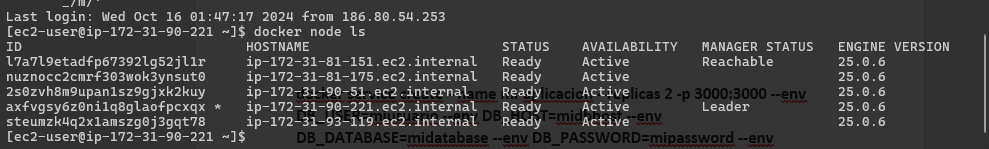
* Instarlar git en la maquina leader

**sudo yum update**

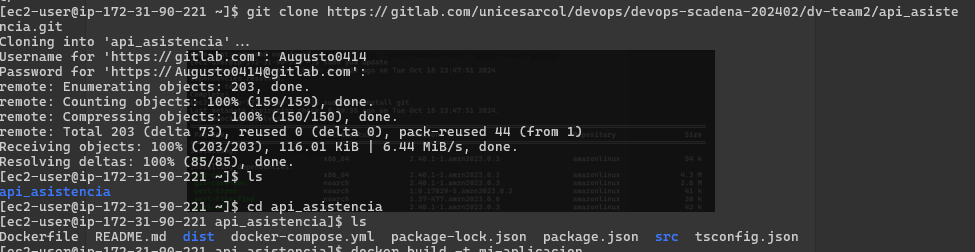
**sudo yum install git**



Clono el siguiente repo en la maquina manager de tipo ( leader)

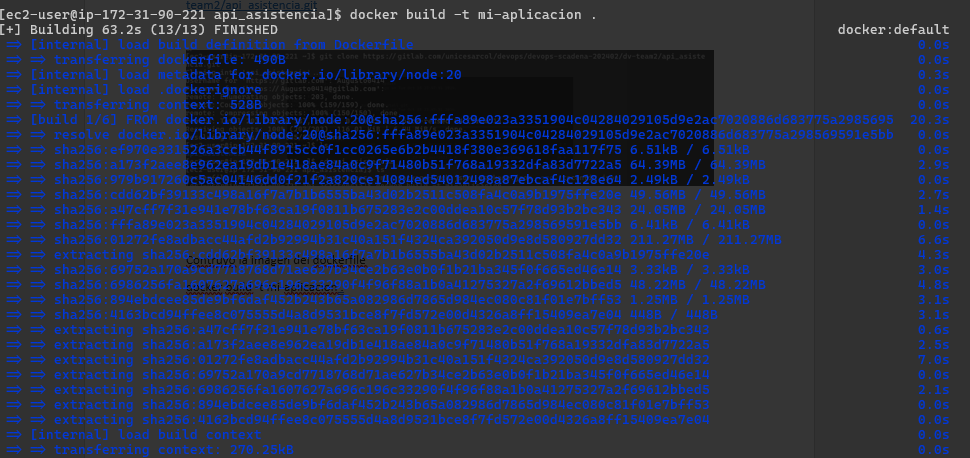


**git clone** [**https://gitlab.com/unicesarcol/devops/devops-scadena-202402/dv-team2/api\_asistencia.git**](https://gitlab.com/unicesarcol/devops/devops-scadena-202402/dv-team2/api_asistencia.git)



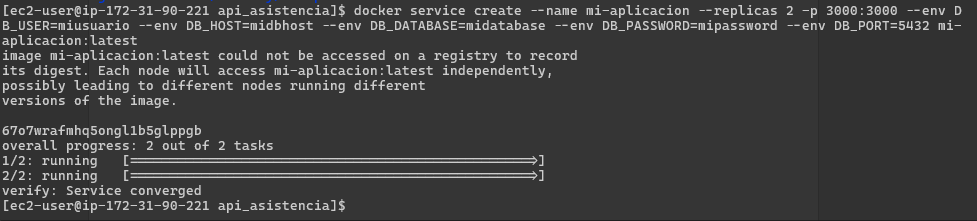
Contruyo la imagen del dockerfile

**docker build -t mi-aplicacion .**



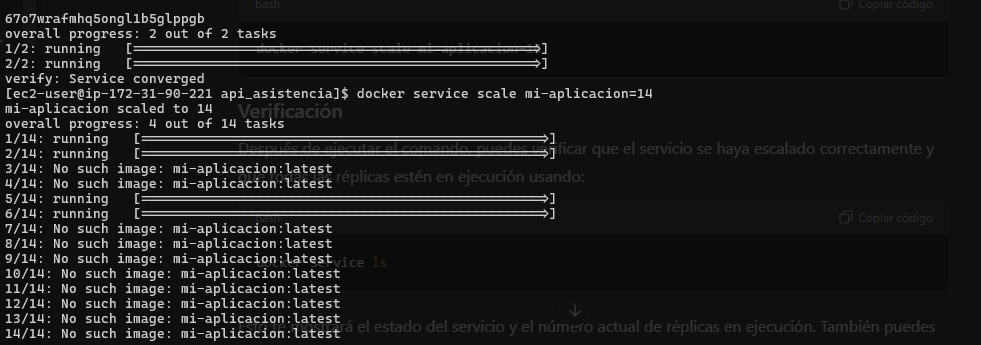
**Despliega** la aplicación en el clúster de Docker Swarm como un **servicio y con 2 instancias iniciales**.

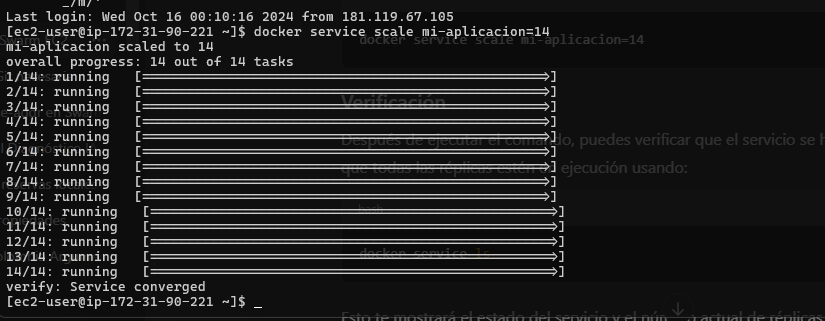
**docker service create --name mi-aplicacion --replicas 2 -p 3000:3000 --env DB\_USER=miusuario --env DB\_HOST=midbhost --env DB\_DATABASE=midatabase --env DB\_PASSWORD=mipassword --env DB\_PORT=5432 mi-aplicacion:latest**



* Escalar el numero de replicas a 14

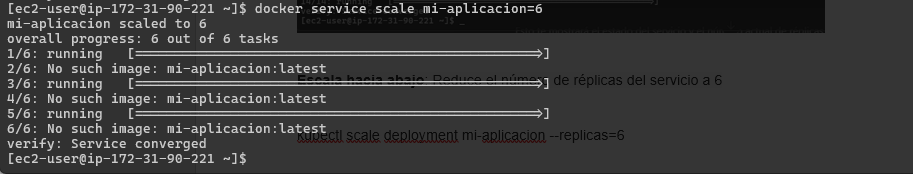
**docker service scale mi-aplicacion=14**





**Escala hacia abajo**: Reduce el número de réplicas del servicio a 6

**docker service scale mi-aplicacion=6**



Es recomendable subir la imagen a un registro de Docker (como Docker Hub) para que todas las instancias de tu servicio en todos los nodos utilicen la misma versión de la imagen.

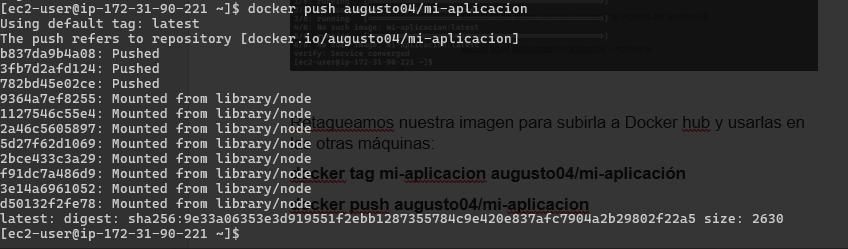
Retagueamos nuestra imagen para subirla a Docker hub y usarlas en las otras máquinas:

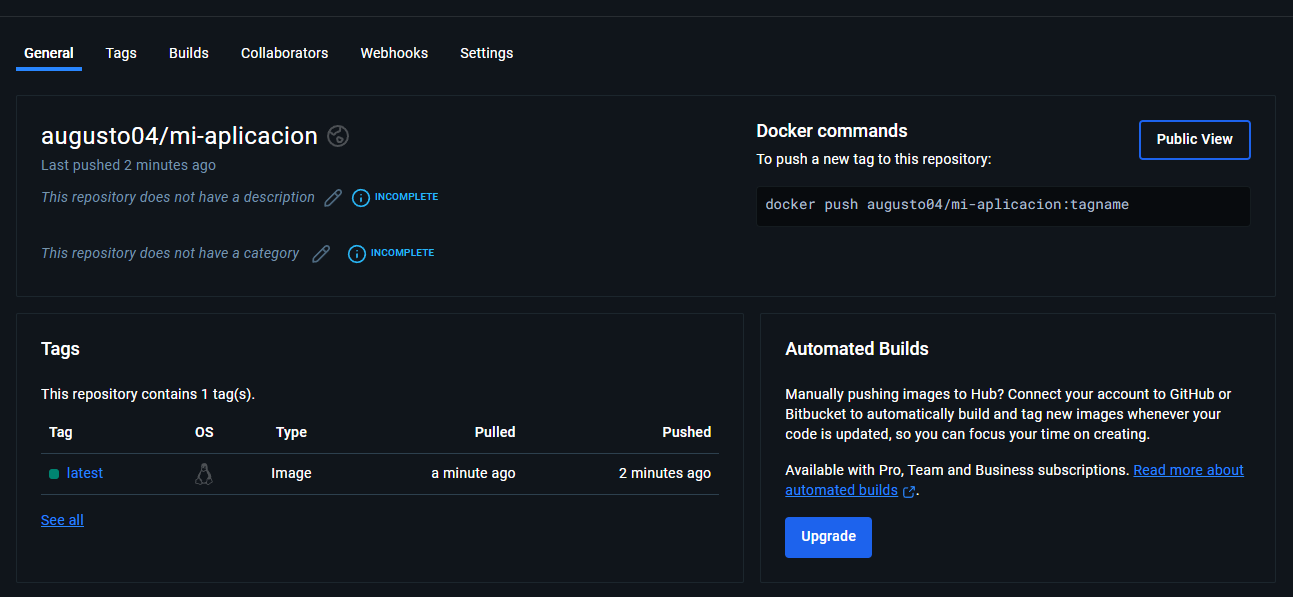
**docker tag mi-aplicacion:latest augusto04/mi-aplicacion:tagname**

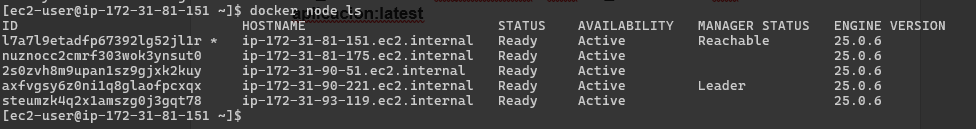
**docker push augusto04/mi-aplicacion:tagname**

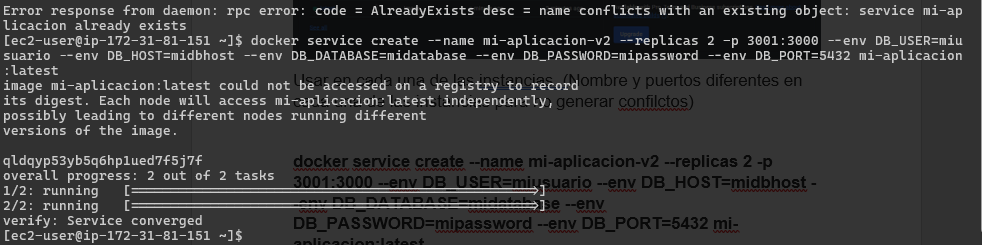
**docker service update --image augusto04/mi-aplicacion:tagname mi-aplicacion**

**mi-aplicacion**

****

****

****

****

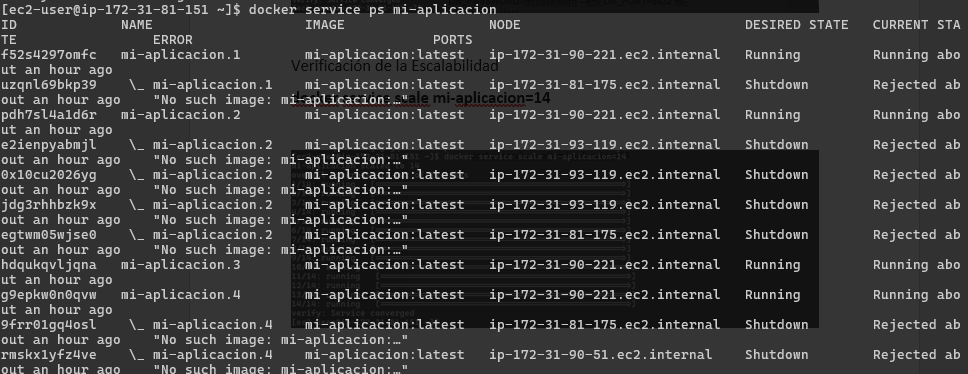
Verificación de la Escalabilidad

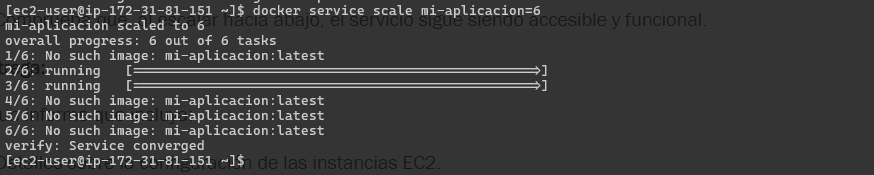
**docker service scale mi-aplicacion=14**

****

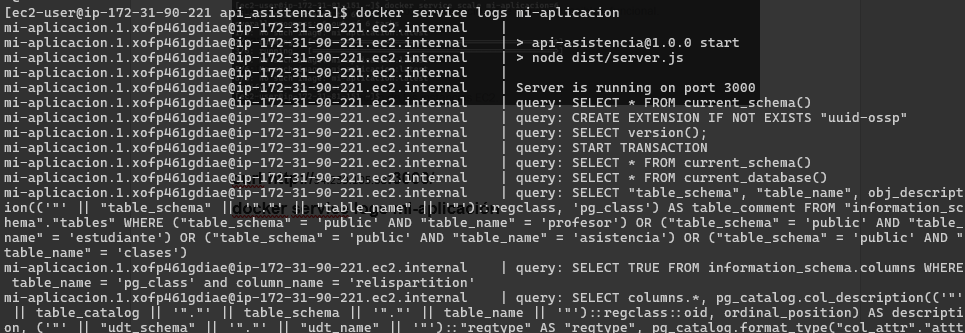
Verifica el estado del servicio nuevamente con:

**docker service ps mi-aplicación**

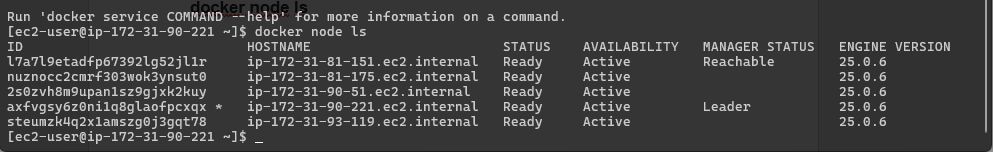
****

**docker service scale mi-aplicacion=6**

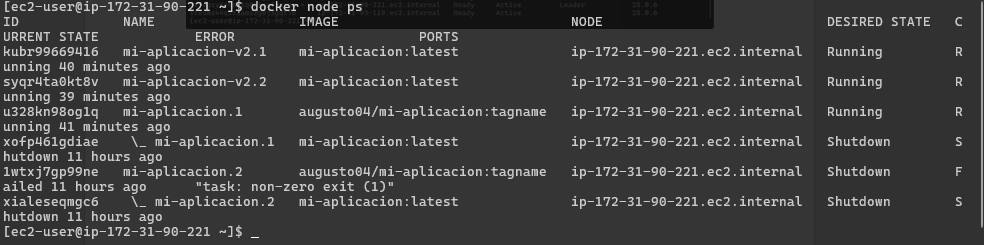
**docker service logs mi-aplicación**

****

**docker node ls**

****

**docker node ps**

****