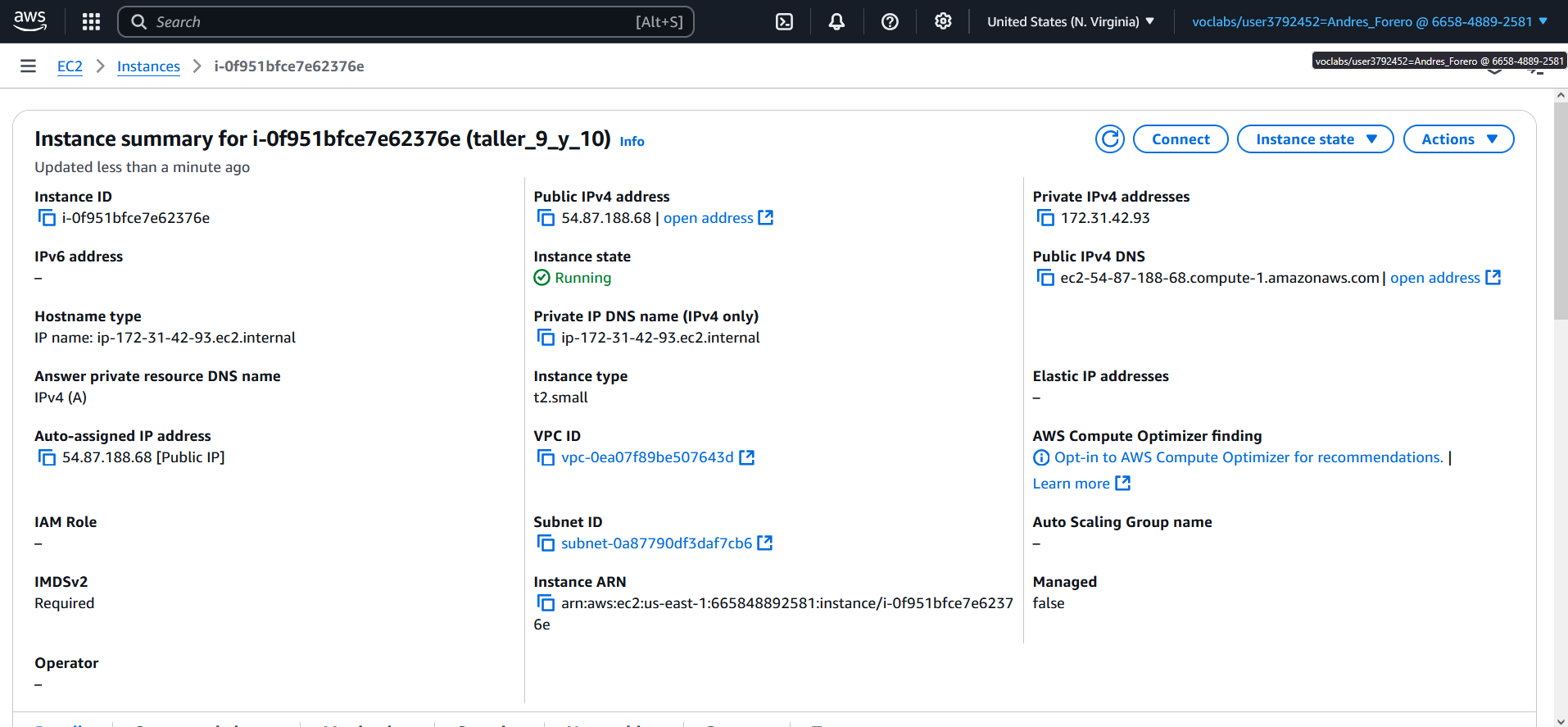
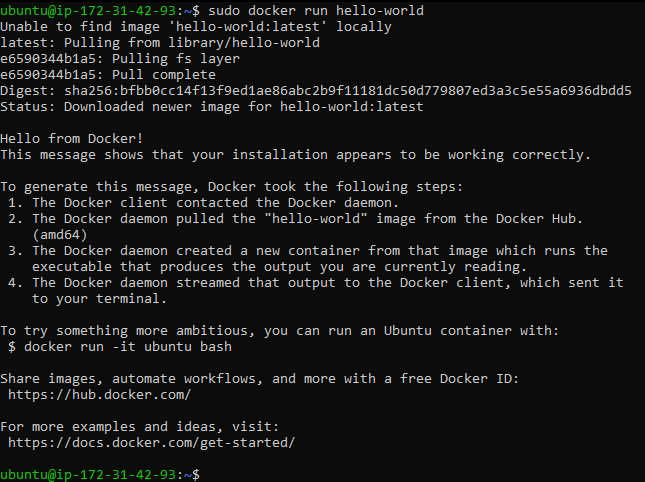
**Taller 9: Despliegue con contenedores: Docker**

Andrés Forero

**Pantallazo de la consola de AWS EC2 con la máquina en ejecución**

****

**Pantallazo y explicación de salida de Docker hello-world**

****

Al ejecutar docker run hello-world, Docker trató de encontrar la imagen localmente, y al no encontrarla, la descargó desde Docker Hub. Luego, creó un contenedor basado en esa imagen, al ejecutarlo muestra el mensaje de bienvenida desde el contenedor de Docker.

**Explicación de Dockerfile**

Este Dockerfile crea un contenedor basado en Python 3.10. Primero, crea un usuario llamado api-user y establece el directorio de trabajo en /opt/bankchurn-api. Luego, copia el código de la aplicación, actualiza pip e instala las dependencias. Se otorgan permisos de ejecución al script run.sh y se cambia la propiedad de los archivos al nuevo usuario. El contenedor se ejecuta como api-user, expone el puerto 8001 y lanza la aplicación usando el script run.sh.

**Pantallazo de salida de docker images**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Pantallazo de salida de docker run**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Pantallazo de navegador con API en ejecución**

**A black and white screen

AI-generated content may be incorrect.**

**Pantallazo de salida de railway up –detach**

**A computer screen shot of a computer code

AI-generated content may be incorrect.**

**Pantallazo de logs de despliegue con mensaje Dockerfile**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Pantallazo del dominio generado**

**A screenshot of a computer

AI-generated content may be incorrect.**

**URL del servicio**

**capable-patience-production.up.railway.app**

**Pantallazo de la API en ejecución con la ruta predict visible**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**