# Docker Commands for DeepStream-MMDeploy Container

## 1. Start Exited Container

To start the exited container with ID `c4f933c8e4fb` and attach to its terminal:

Command:  
```bash  
docker start -ai c4f933c8e4fb  
```

## 2. Start Container with Root Access, GPU, Folder Mount, and Display

To run the container with the necessary privileges, mount required folders, and ensure GPU and display access, use the following command:

Command:  
```bash  
docker run -it --rm --gpus all \  
 --net=host \  
 --privileged \  
 --env DISPLAY=$DISPLAY \  
 -v /tmp/.X11-unix:/tmp/.X11-unix \  
 -v /workspace:/workspace \  
 -v /path/to/host/folder:/path/to/container/folder \  
 --name deepstream\_auto\_ready \  
 --shm-size=1g \  
 -e NVIDIA\_DRIVER\_CAPABILITIES=all \  
 -e NVIDIA\_VISIBLE\_DEVICES=all \  
 --ipc=host \  
 deepstream\_trt\_ready:v1  
```  
```bash  
# Explanation:  
# --gpus all - Enables all available GPUs.  
# --net=host - Uses the host's network.  
# --privileged - Grants the container elevated privileges.  
# --env DISPLAY=$DISPLAY - Provides display access for GUI applications.  
# -v /tmp/.X11-unix:/tmp/.X11-unix - Shares X11 socket to allow GUI.  
# -v /workspace:/workspace - Mounts /workspace directory.  
# -v /path/to/host/folder:/path/to/container/folder - Mounts additional folder from host to container.  
# --name deepstream\_auto\_ready - Assigns a name to the container.  
# --shm-size=1g - Allocates shared memory for video processing.  
# -e NVIDIA\_DRIVER\_CAPABILITIES=all - Grants all driver capabilities.  
# -e NVIDIA\_VISIBLE\_DEVICES=all - Makes all GPUs visible to the container.  
# --ipc=host - Shares IPC namespace with the host for better performance.  
```

## 3. Open Container as Root and Attach to Terminal

To restart the container as root with all permissions and attach to its terminal:  
```bash  
docker start --attach --interactive --user root c4f933c8e4fb  
```

## 4. Check GPU and Running Containers

To verify GPU status and list running containers:  
```bash  
nvidia-smi  
docker ps  
```