DSAI Workflow Guide

DSP-Lab BIU 2025

Table of content

Conventions:

- Terminal commands and `code` in purple, e.g `docker ps`
- Elements to be changed by you in red, e.g username

Connection to the servers

1. 1. DSI-head
2. 2. SSH config file
3. 3. SSH keys

DSI-head

Connecting to the compute nodes is done via the dsihead server.

This server is reserved for that only, and you may not run any task on it.

Command:

```
ssh username@dsihead.lnx.biu.ac.il
```

SSH config file

Using an SSH configuration file can make connecting and working in the DSAI easy.

The file normally exists in:

Linux/MacOS - `/home/username/.ssh/config`

Windows- `C:\Users\username\.ssh\config`

By adding the following attribute to the SSH config file you will be able to connect to the dsi-head without writing the full command:

```
Host dsi-head
    HostName dsihead.lnx.biu.ac.il
    User username
```

Now you will be able to connect to the dsi-head with the command:

```
ssh dsi-head
```

SSH config file

The SSH config file can also help you connect directly to a compute node:

```
Host dsigpu01
    HostName dsigpu01
    User username
    ProxyCommand ssh -X -q dsi-head nc dsigpu01 22
```

Now you will be able to connect to the compute node ‘dsigpu01’ with the command:

```
ssh dsigpu01
```

But, Make sure your config file has the dsi-head attribute as well!

SSH config file

A complete SSH config file example for one compute node:

```
Host dsi-head
    HostName dsihead.lnx.biu.ac.il
    User username

Host dsigpu01
    HostName dsigpu01
    User username
    ProxyCommand ssh -X -q dsi-head nc dsigpu01 22
```

A list of available compute nodes is in slide 10.

SSH key

By using an SSH key you can avoid typing your password every time you need to connect to a compute node or to the dsi-head.

On your PC - create and ssh-key with the Terminal:

```
ssh-keygen
```

This will create an ssh key pair (private and public keys) at the following path:

Linux/MacOS - `/home/username/.ssh/id_rsa.pub`

Windows- `C:\Users\username\.ssh\id_rsa.pub`

The name of the key my vary - it will be printed in the logs of the command above.

SSH key

After creating a key, we need copy it to the dsi-head.

Linux/MacOS -

```
ssh-copy-id dsi-head
```

Windows- No similar command is available in windows, manually copy the **PUBLIC (.pub)** key content and paste it in the remote server:

```
ssh dsi-head

nano ~/.ssh/authorized_keys

# if ~/.ssh is and unknown directory simply create it with:

    mkdir ~/.ssh

# Then paste your key content and save the file using ctrl+X
```

Now you will be able to ssh to the dsi-head without typing your password.

Getting to know our servers

GPU:

dsigpu01-08

dsiaspl01

dsiaspl02

dsiaspl03 - For undergraduate students

dsiaudience01

dgx03 (down …)

CPU:

dsicpu01

Optional resources : (upon Coordination with Sharon & Pini)

hpc8h200-01 - GPU 0

hpc8l4-01 - GPU 6

Dgx01 - On demand only

Docker

Docker

Docker is a platform that allows you to package applications and their dependencies into lightweight, portable containers that can run consistently across different environments.

Docker - https://www.docker.com/

Docker hub - https://hub.docker.com/

In the DSAI, each compute node has its own Docker directory hence if you build/pull an image on one machine it **will not transfer** to other.

Running a container as the user

By default the ‘docker run’ command will run the container as root. That may cause problems security and with file permissions on the host machine.

To avoid it we need to create a container with our user in it.

Before creating your docker container, you need your UID.

```
ssh dsi-head

id -u
```

The number printed is your UID for the Dockerfile - Next slide.

By using a Dockerfile and docker-compose file you eliminate the use of the `‘docker run …’` command.

Dockerfile

```
FROM docker/image/of/choice:version
ARG USERNAME=username
ARG UID=UID
ARG GID=2102
ENV PATH="/home/$USERNAME/.local/bin:$PATH"
RUN apt-get update && apt-get install -y sudo python3-pip x11-apps
RUN groupadd --gid 2102 dsi && \
   groupadd --gid 4960 docker && \
   groupadd --gid 6648 ug_dsi && \
   groupadd --gid 6653 ug_gannot && \
   groupadd --gid 6656 ug_dgx && \
   groupadd --gid 41052 ug_hpc && \
   groupadd --gid 668400055 ug_audience && \
   useradd --uid $UID --gid $GID --groups
dsi,docker,ug_dsi,ug_gannot,ug_dgx,ug_hpc,ug_audience \
   --create-home --shell /bin/bash $USERNAME && \
   echo "$USERNAME ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/$USERNAME
ENV DISPLAY=:0
ENV NVIDIA_VISIBLE_DEVICES=all
ENV NVIDIA_DRIVER_CAPABILITIES=all
RUN mkdir -p /dev/shm && chmod 1777 /dev/shm
USER $USERNAME
CMD ["/bin/bash"]
```

Running a container - docker compose

```
version: "3.8"

services:
  your_service_name:
    build:
      context: /path/to/directory/of/Dockerfile
      dockerfile: /path/to/Dockerfile
    image: username_image_name:version
    container_name: username_containername
    privileged: true
    network_mode: "host"
    shm_size: "24G"
    environment:
      - DISPLAY=${DISPLAY}
      - NVIDIA_VISIBLE_DEVICES=all
      - NVIDIA_DRIVER_CAPABILITIES=all
    user: "UID:2102"
    volumes:
      - ${HOME}:/home/username
      - /tmp/.X11-unix:/tmp/.X11-unix
      - $HOME/.Xauthority:/home/username/.Xauthority:rw
      - /etc/resolv.conf:/etc/resolv.conf
      - /dsi:/dsi
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              capabilities: [gpu]
    stdin_open: true
    tty: true
```

Running a container - docker compose

Deploying a docker container is available only on compute nodes.

With the following command the container will be built and will be deployed as well:

```
docker-compose up --build
```

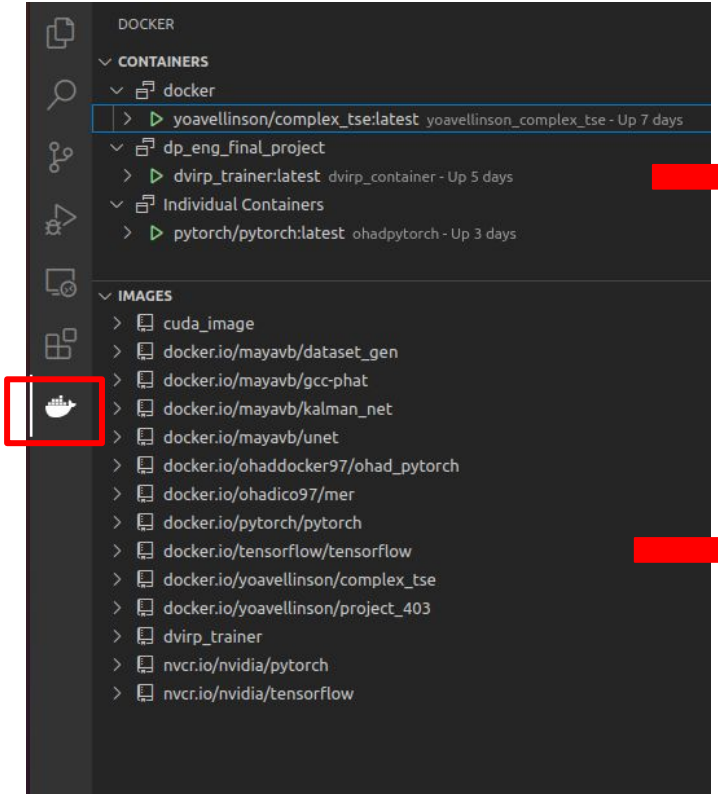
Some compute nodes have different versions of docker that may require this command:

```
docker compose up --build
```

Docker commands

Command	Description
docker ps	List running containers
docker ps -a	List all containers (including stopped)
docker images	List all local images
docker pull <image>	Download image from Docker Hub
docker build -t <name> .	Build image from Dockerfile in current dir
docker rm <container>	Remove a container
docker exec -it <container> bash	Run command in running container
docker commit <container> <new_image_name>	Save the current state of a container as a new image

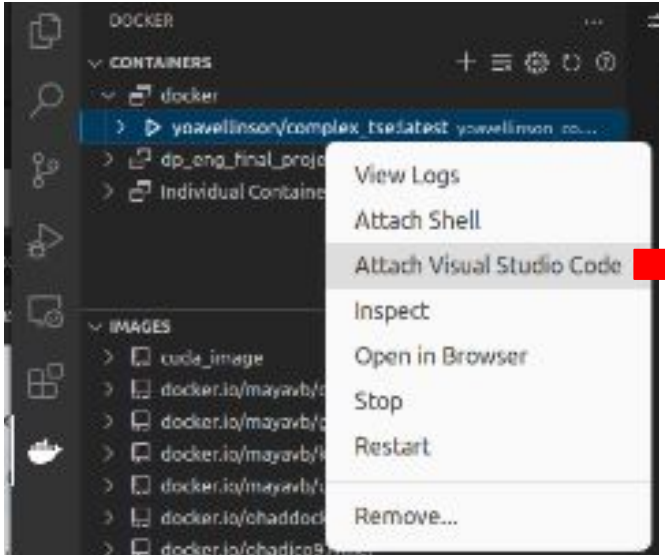
Docker in VS-Code



Running containers on the compute node.

Available images on the compute node.

Best practice!



Attach VS-Code session to a running container

DSAI workflow

1. Managing files and storage
2. Git
3. Recommended software packages and containers

Managing files and storage

Do and Do-not in the DSAI folders

1. Home directory - has a limit of 100GB - if exceeds data will be deleted
2. /dsi/gannot-lab1/datasets + /dsi/gannot-lab2/datasets2: Lab storage for large datasets, before adding/downloading a new dataset make sure it is not already in one of these directories.
3. Always delete compressed files after extraction - e.g .tar/zip/.tar.gz files.
4. /dsi/scratch : Temporary storage that expires (deleted) every few weeks.

Git intro

Always use git to backup your files!

Most used commands:

1. cd /path/to/directory
2. git status
3. git add .
4. git commit -m “commit msg”
5. git push origin <branch>

Recommended software packages and containers

1. IDE: VS-Code: can help manage all the ssh configuration and docker images using the Docker extension and remote-development extension.
2. Software packages:
   a. Wandb - allows you to track your training online.
   b. Screen- allows you to leave a process running in the background.
   c. TMUX - similar to screen with more features.
3. Docker images:

   I mainly recommend using the NVIDIA pytorch containers as a base container because they work great with GPU support out of the box.

   Latest image : nvcr.io/nvidia/pytorch:25.03-py3

Questions?