Ansible script

```
---
- hosts: all
  become: yes
  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install basic packages
      apt:
        name:
          - ubuntu-desktop
          - git
          - curl
          - wget
          - build-essential
          - software-properties-common
          - python3-pip
          - python3-opencv
        state: present

    - name: Add NVIDIA PPA
      apt_repository:
        repo: "ppa:graphics-drivers/ppa"
        state: present

    - name: Install NVIDIA driver
      apt:
        name: "nvidia-driver-560"
```

```
    state: present
    update_cache: yes

- name: Install CUDA 12.1
  apt:
    name: "cuda-toolkit-12-1"
    state: present

- name: Install cuDNN 8.9
  apt:
    name: "libcudnn8"
    state: present

- name: Install TensorRT 8.6
  apt:
    name: "tensorrt"
    state: present

- name: Install Docker dependencies
  apt:
    name:
      - apt-transport-https
      - ca-certificates
      - curl
      - software-properties-common
    state: present

- name: Add Docker GPG key
  apt_key:
    url: https://download.docker.com/linux/ubuntu/gpg
    state: present
```

```
  - name: Add Docker repository
    apt_repository:
      repo: "deb [arch=amd64] https://download.docker.com/linux/ubuntu jammy stable"
      state: present

  - name: Install Docker
    apt:
      name:
        - docker-ce
        - docker-ce-cli
        - containerd.io
      state: present
  - name: Install Docker Compose
    get_url:
      url: https://github.com/docker/compose/releases/download/v2.20.0/docker-compose-linux-x86_64
      dest: /usr/local/bin/docker-compose
      mode: '0755'

  - name: Add user to Docker group
    user:
      name: "{{ ansible_user }}"
      groups: docker
      append: yes

  - name: Install VSCode
    apt:
      name: code
      state: present
```

```
- name: Install VSCode extensions
  command: code --install-extension {{ item }}
  with_items:
    - ms-vscode.cpptools
    - ms-azuretools.vscode-docker
    - ms-toolsai.jupyter
    - ms-toolsai.jupyter-keymap
    - ms-toolsai.jupyter-renderers
    - ms-toolsai.jupyter-slideshow
    - shd101wyy.markdown-preview-enhanced
    - davidanson.vscode-markdownlint
    - ms-python.vscode-pylance
    - ms-python.python
    - ms-python.vscode-python-debugger

- name: Install QGroundControl
  get_url:
    url: https://d176tv9ibo4jno.cloudfront.net/latest/QGroundControl.AppImage
    dest: /opt/QGroundControl.AppImage
  register: qgc_download

- name: Make QGroundControl executable
  file:
    path: /opt/QGroundControl.AppImage
    mode: '0755'
  when: qgc_download.changed

- name: Install Python packages
  pip:
    name:
      - pyopengl
```

```
    - opencv-python==4.9.0.80
    - torch==2.1.0+cu121
    - torchvision==0.16.0+cu121
    - ultralytics==8.3.24

- name: Clone CVAT repository
  git:
    repo: https://github.com/opencv/cvat.git
    dest: /opt/cvat

- name: Deploy CVAT with Docker Compose
  command: docker-compose up -d
  args:
    chdir: /opt/cvat

- name: Download ZED SDK
  get_url:
    url: https://download.stereolabs.com/zedsdk/4.2/ubuntu22
    dest: /tmp/zed_sdk.run

- name: Install ZED SDK
  command: /tmp/zed_sdk.run -- silent
  args:
    creates: /usr/local/zed  # Prevents reinstallation if already installed
```

Dockerfile scrip

```
# Use the official NVIDIA CUDA 12.1 base image with Ubuntu 22.04
FROM nvidia/cuda:12.1.0-devel-ubuntu22.04

# Set environment variables
ENV DEBIAN_FRONTEND=noninteractive

# Install system dependencies
RUN apt-get update && apt-get install -y \
    ubuntu-desktop \
    git \
    curl \
    wget \
    build-essential \
    software-properties-common \
    python3-pip \
    python3-opencv \
    libgl1 \
    libglib2.0-0 \
    && rm -rf /var/lib/apt/lists/*

# Install ZED SDK
RUN wget https://download.stereolabs.com/zedsdk/4.2/ubuntu22 -O /tmp/zed_sdk.run
RUN chmod +x /tmp/zed_sdk.run
RUN /tmp/zed_sdk.run --silent

# Install cuDNN and TensorRT (pre-installed in the base image)
# Verify versions:
RUN echo "cuDNN version: $(cat /usr/include/cudnn_version.h | grep CUDNN_MAJOR -A 2)" &&
\
```

```
    echo "TensorRT version: $(dpkg -l | grep tensorrt | awk '{print $3}')"

# Install PyTorch with CUDA 12.1 support
RUN pip3 install --upgrade pip && \
    pip3 install torch==2.1.0 torchvision==0.16.0 --index-url
https://download.pytorch.org/whl/cu121

# Install Python packages from requirements.txt
COPY requirements.txt /tmp/requirements.txt
RUN pip3 install -r /tmp/requirements.txt

# Install QGroundControl (optional, if needed in the container)
RUN wget https://d176tv9ibo4jno.cloudfront.net/latest/QGroundControl.AppImage -O
/opt/QGroundControl.AppImage && \
    chmod +x /opt/QGroundControl.AppImage

# Install VSCode Server (optional, for browser-based VSCode)
RUN curl -fsSL https://code-server.dev/install.sh | sh

# Set up working directory
WORKDIR /workspace

# Expose ports for Jupyter, CVAT, and VSCode
EXPOSE 8888 8080 3000

# Default command (start a shell)
CMD ["/bin/bash"]
```