Nuclio model deployment - Tutorial

A) Install Docker Engine for Ubuntu :

1) Set up the repository:

1- Update the apt package and install packages to allow apt to use a repository over HTTPS

Python

```
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
```

2- Add Docker's official GPG key

Python

```
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

3- Use the following command to set up the repository:

Python

```
echo \
 "deb [arch="$(dpkg --print-architecture)"
signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu  \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
 sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

4- Update the apt package index:

Python

```
sudo apt-get update
```

2) Install Docker Engine:

1- To install the last version :

Python

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

2- To check the installation :

Python

```
sudo docker run hello-world
```

B) Clone CVAT github repo :

Python

```
git clone https://github.com/opencv/cvat.git cvat
```

C) Installation Auto Annotations (Nuclio) :

Go within the CVAT repo :

Python

```
cd cvat
```

To bring up cvat with auto annotation tool (If you did any changes to the Docker Compose files, make sure to add --build at the end) :

Python

```
sudo docker compose -f docker-compose.yml -f components/serverless/docker-compose.serverless.yml up -d
```

(To Stop the containers :)

Python

```
docker compose -f docker-compose.yml -f
components/serverless/docker-compose.serverless.yml down
```

Install nuctl : *(with sudo if necessary)*

Python

```
wget
https://github.com/nuclio/nuclio/releases/download/<version>/nuctl-<version>
-linux-amd64
```

After downloading the nuclio, give it a proper permission and do a softlink :

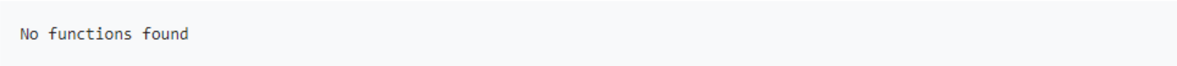
Python

```
sudo chmod +x nuctl-<version>-linux-amd64
sudo ln -sf $(pwd)/nuctl-<version>-linux-amd64 /usr/local/bin/nuctl
```

Test to check the installation :

Python

```
sudo nuctl get functions
```



```
No functions found
```

D) Using builtin DL models :

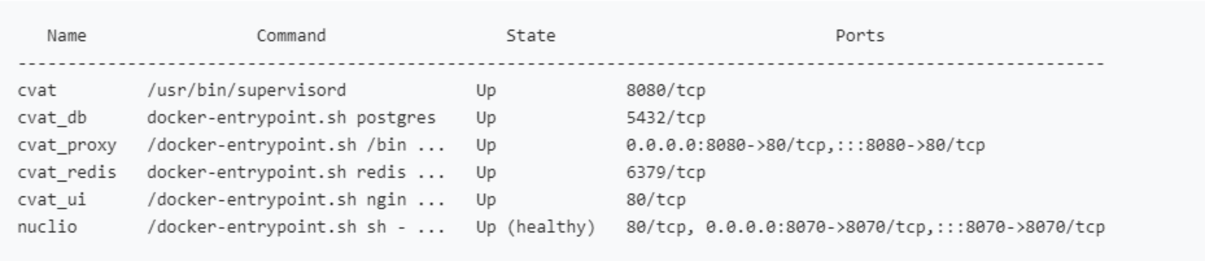
Build CVAT with severless support :

Python

```
sudo docker compose -f docker-compose.yml -f docker-compose.dev.yml -f
components/serverless/docker-compose.serverless.yml up -d --build
```

Check the container status :

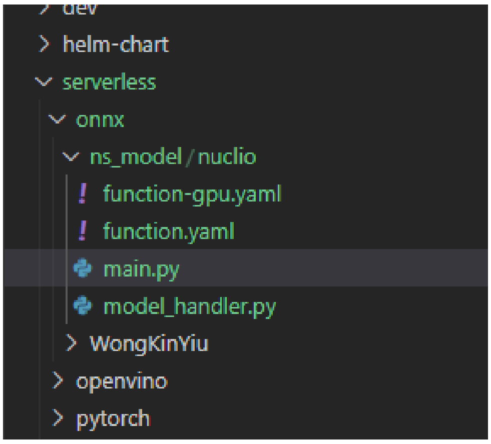
```
Python
sudo docker compose -f docker-compose.yml -f docker-compose.dev.yml -f
components/serverless/docker-compose.serverless.yml ps
```



```
    Name              Command                 State                    Ports
-------------------------------------------------------------------------------------------------------------
cvat         /usr/bin/supervisord            Up            8080/tcp
cvat_db      docker-entrypoint.sh postgres   Up            5432/tcp
cvat_proxy   /docker-entrypoint.sh /bin ...  Up            0.0.0.0:8080->80/tcp,:::8080->80/tcp
cvat_redis   docker-entrypoint.sh redis ...  Up            6379/tcp
cvat_ui      /docker-entrypoint.sh ngin ...  Up            80/tcp
nuclio       /docker-entrypoint.sh sh - ...  Up (healthy)  80/tcp, 0.0.0.0:8070->8070/tcp,:::8070->8070/tcp
```

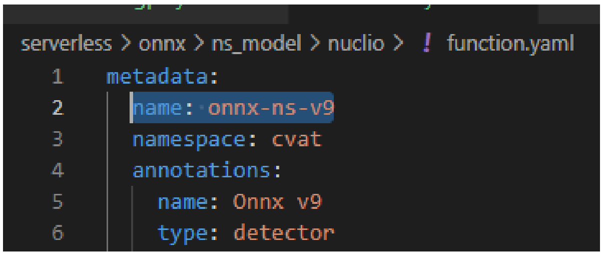
<u>Remark :</u> ***I was inspired by the WongKinYiu Onnx Yolov7 project located in*** `severless/onnx/WongKinYiu/yolov7/nuclio`

Create a new folder into `severless/onnx/`**. The folder will follow this scheme :** `<folder_name>/nuclio`**.**
For instance, `ns_model/nuclio` *into* `severless/onnx/` *gives the folder path* `severless/onnx/ns_model/nuclio` *:*



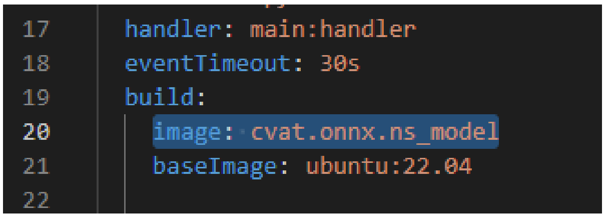
In the function.yaml file, you must :

- *modify the name of the nuclio function*



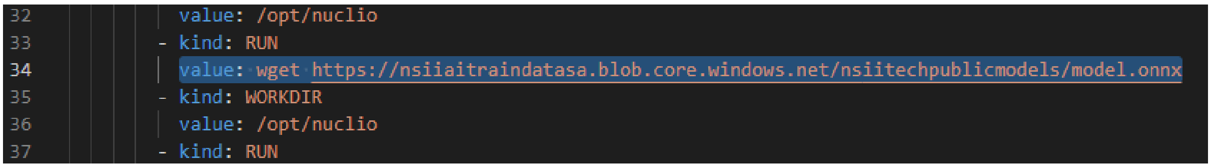
```
metadata:
  name: onnx-ns-v9
  namespace: cvat
  annotations:
    name: Onnx v9
    type: detector
```

- *modify the name of the image*



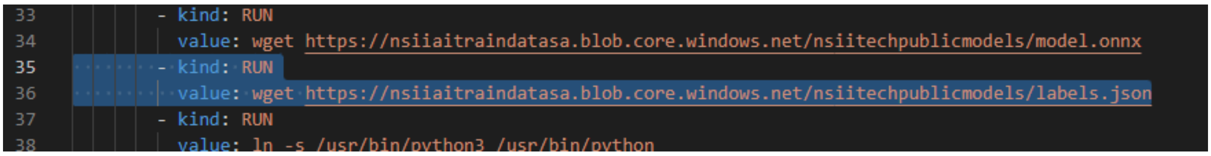
```
handler: main:handler
eventTimeout: 30s
build:
  image: cvat.onnx.ns_model
  baseImage: ubuntu:22.04
```

- *change the model url to download the onnx model*



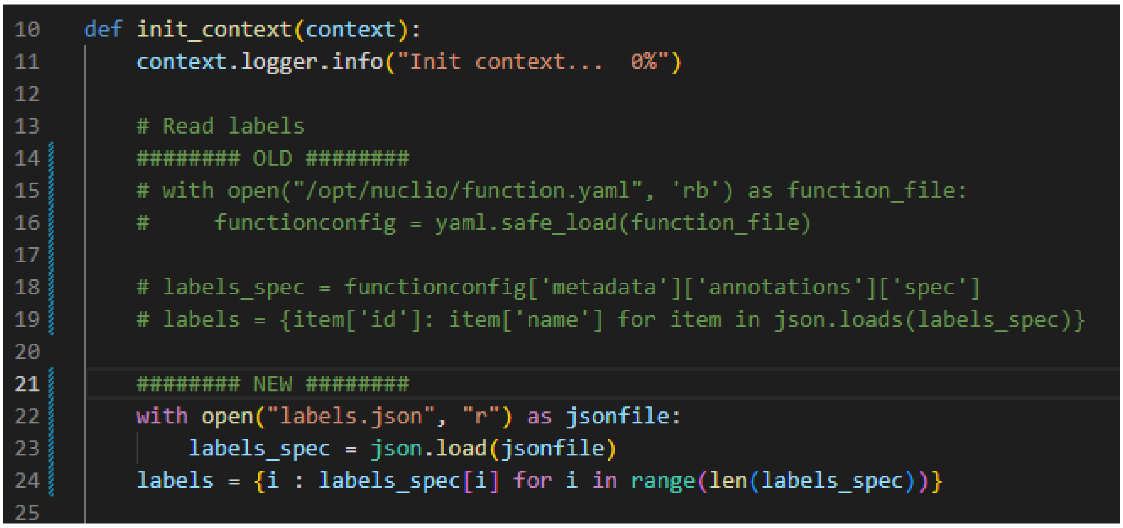
```
  value: /opt/nuclio
- kind: RUN
  value: wget https://nsiiaitraindatasa.blob.core.windows.net/nsiitechpublicmodels/model.onnx
- kind: WORKDIR
  value: /opt/nuclio
- kind: RUN
```

- *add 2 additional lines to download the labels json file after the model*



```
- kind: RUN
  value: wget https://nsiiaitraindatasa.blob.core.windows.net/nsiitechpublicmodels/model.onnx
- kind: RUN
  value: wget https://nsiiaitraindatasa.blob.core.windows.net/nsiitechpublicmodels/labels.json
- kind: RUN
  value: ln -s /usr/bin/python3 /usr/bin/python
```

In the `main.py` file, you must modify the way of reading labels in the `init_context` function :



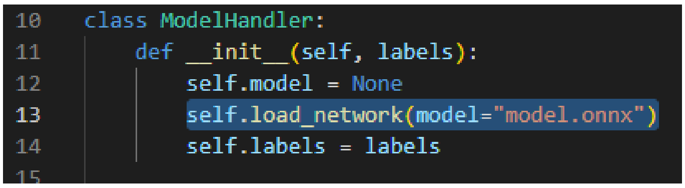
```
def init_context(context):
    context.logger.info("Init context...  0%")

    # Read labels
    ######## OLD ########
    # with open("/opt/nuclio/function.yaml", 'rb') as function_file:
    #     functionconfig = yaml.safe_load(function_file)

    # labels_spec = functionconfig['metadata']['annotations']['spec']
    # labels = {item['id']: item['name'] for item in json.loads(labels_spec)}

    ######## NEW ########
    with open("labels.json", "r") as jsonfile:
        labels_spec = json.load(jsonfile)
    labels = {i : labels_spec[i] for i in range(len(labels_spec))}
```

In the `model_handler.py` file, you must the model name in the `ModelHandler` class :



```
class ModelHandler:
    def __init__(self, labels):
        self.model = None
        self.load_network(model="model.onnx")
        self.labels = labels
```

Then, come back to the shell and create a nuclio project :

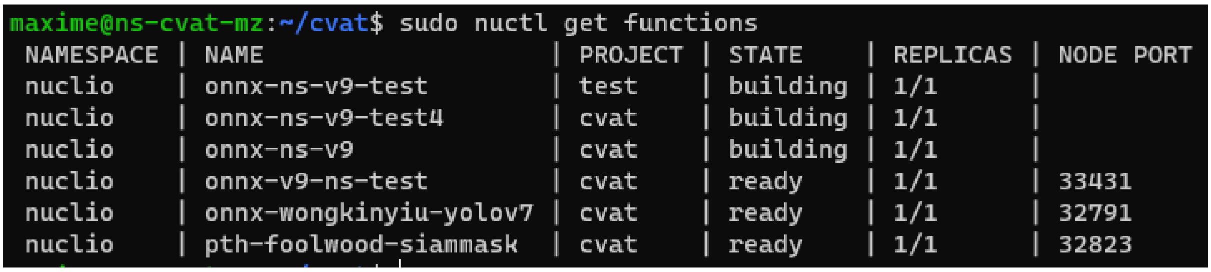
```Python
sudo nuctl create project cvat
```

Deploy your model as a nuclio function into the project just created :

```
Python
sudo nuctl deploy --project-name cvat --path "./serverless/onnx/ns_model"
--platform local
```

See your functions by executing :

```
Python
sudo nuctl get functions
```



```
maxime@ns-cvat-mz:~/cvat$ sudo nuctl get functions
 NAMESPACE | NAME                    | PROJECT | STATE    | REPLICAS | NODE PORT
 nuclio    | onnx-ns-v9-test         | test    | building | 1/1      |
 nuclio    | onnx-ns-v9-test4        | cvat    | building | 1/1      |
 nuclio    | onnx-ns-v9              | cvat    | building | 1/1      |
 nuclio    | onnx-v9-ns-test         | cvat    | ready    | 1/1      | 33431
 nuclio    | onnx-wongkinyiu-yolov7  | cvat    | ready    | 1/1      | 32791
 nuclio    | pth-foolwood-siammask   | cvat    | ready    | 1/1      | 32823
```

Invoke your function :

```
Python
sudo nuctl invoke <function_name>
```