
THE VICTRE CONTAINER

INSTRUCTIONS FOR DOWNLOAD AND INSTALL

AUTHORS

PURVA SAWANT, DIDSr/CDRH/FDA (Student intern)
DIKSHA SHARMA, DIDSr/CDRH/FDA
ALDO BADANO, DIDSr/CDRH/FDA

DISCLAIMER

This software and documentation (the "Software") were developed at the Food and Drug Administration (FDA) by employees of the Federal Government in the course of their official duties. Pursuant to Title 17, Section 105 of the United States Code, this work is not subject to copyright protection and is in the public domain. Permission is hereby granted, free of charge, to any person obtaining a copy of the Software, to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, or sell copies of the Software or derivatives, and to permit persons to whom the Software is furnished to do so. FDA assumes no responsibility whatsoever for use by other parties of the Software, its source code, documentation or compiled executables, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. Further, use of this code in no way implies endorsement by the FDA or confers any advantage in regulatory decisions. Although this software can be redistributed and/or modified freely, we ask that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.

PURPOSE

This file gives a description of how the VICTRE pipeline is built in a container and deployed on different platforms.

This file contains Dockerhub password.

REQUIREMENTS

Pre-requisites:

The VICTRE pipeline requires a Linux kernel and CUDA supported NVIDIA card.

You can run the Pipeline either on a local server or in the cloud.

A) Running on **local server**, follow the below steps to install docker and required drivers.

1. Docker: Installation Required by user (Ubuntu):

STEPS:

1. First, add the GPG key for the official Docker repository to the system:

```
$ curl -fsSL  
https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

2. Add the Docker repository to APT sources:

```
$ sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

3. Next, update the package database with the Docker packages from the newly added repo:

```
$ sudo apt-get update
```

4. Make sure you are about to install from the Docker repo instead of the default

Ubuntu 16.04 repo

```
$ apt-cache policy docker-ce
```

You should see output similar to the follow:

docker-ce:

Installed: (none)

Candidate: 17.03.1~ce-0~ubuntu-xenial

Version table:

17.03.1~ce-0~ubuntu-xenial 500

500

```
https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
```

17.03.0~ce-0~ubuntu-xenial 500

500

```
https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
```

5. Now, Install docker (You need be the root user to install docker)

```
$ sudo apt-get install -y docker-ce
```

6. Check if Docker demon has started

```
$ sudo systemctl status docker
```

The output should be similar to:

â–¤ docker.service - Docker Application Container

Engine

Loaded: loaded

(/lib/systemd/system/docker.service; enabled; vendor preset: enabled)

Active: active (running) since Sun 2016-05-01

06:53:52 CDT; 1 weeks 3 days ago

Docs: <https://docs.docker.com>

Main PID: 749 (docker)

7. Make sure it starts at every server
\$ sudo systemctl enable docker

2. NVIDIA Drivers: Installation Required by user (Ubuntu):

1. Linux headers : \$ sudo apt-get install -y gcc
linux-headers-\$(uname -r)
2. \$ sudo apt-get purge NVIDIA*
\$ sudo add-apt-repository ppa:graphics-drivers
\$ sudo apt-get update
\$ sudo apt-get install NVIDIA-387 (Install the
latest one :

<https://launchpad.net/~graphics-drivers/+archive/ubuntu/ppa>)

3. Install NVIDIA-Docker: Installation Required by user (Ubuntu):

1. \$ sudo docker volume ls -q -f driver=NVIDIA-docker
| xargs -r -I{} -nl docker ps -q -a -f volume={} | xargs -r docker rm
-f
\$ sudo apt-get purge NVIDIA-docker
2. Repositories:
Install as per your Ubuntu Distribution :
<https://NVIDIA.github.io/NVIDIA-docker/>
3. sudo apt-get install NVIDIA-docker2
4. sudo pkill -SIGHUP dockerd

B) Running **in the cloud**; we used Amazon Web Services (AWS) for running VICTRE on the cloud and describe below the steps to follow. Any cloud service provider can be used as long as they offer GPU instances with enough memory size.

AWS offers a good selection of GPU instances which can be found here: <https://aws.amazon.com/ec2/instance-types/>. We recommend the g3.16xlarge instance for running the pipeline.

1. Creating Amazon account:

Sign in into Amazon console :
<https://aws.amazon.com/console/>

If it is your first time creating an account with AWS, you will need to enter your credit card information.

2. Selecting the AWS Service:

For this application we will require AWS Elastic Cloud Compute that provides resizable compute capacity in the cloud. On your Dashboard click on Services -> Compute -> EC2.

3. Launching an Instance:

i) Click on Launch Instance -> Select Ubuntu Server 16.04 LTS (HVM),

SSD Volume Type - ami-cd0f5cb6

(64 bit) - > g3.16xlarge instance

ii) Before launching, edit the storage option and make the RAM

space to atleast 50 GB which would be required to support VICTRE simulation.

iii) Click Review and Launch.

4. Key pair:

A key pair consists of a public key that AWS stores, and a private key file that you store.

Together, they allow you to connect to your instance securely.

For Windows AMIs, the private key file is required to obtain the password used to log into your instance.

For Linux AMIs, the private key file allows you to securely SSH into your instance.

Once the key dialog box appears, from the drop down choose create a new pair. Enter an appropriate key pair name and download it.

NOTE: Do not lose this key, it is generated only once by the system.

It will take a minute to set up your instance.

Once you see the instance status as "running", it means your that instance is now active and you can start the configuration.

5. SSH into you AWS instance :

Give the right permissions to you key - \$ chmod key.pem

SSH : ssh -i "key.pem" ubuntu@exyz.amazonaws.com

Note : If you click on the connect option on the EC2 dashboard, you will know the exact ssh command for your instance.

6. After entering into the instance, follow the above steps I] i) to install further dependencies.

INSTALLING AND USING THE VICTRE CONTAINER

1. Login (*Only require till the VICTRE Container is Private*)

```
$ sudo docker login --username=didsrfda
password: Victrel23@abc
```

2. Pull the VICTRE container from Docker hub:

```
$ sudo NVIDIA-docker pull didsrfda/victre-pipeline:v7
```

1. Enter the container using:

- For AWS cloud users:

```
$ sudo NVIDIA-docker run -p 0.0.0.0:80:5000 -it
didsrfda/victre-pipeline:v5 /bin/bash
```

- For local server (will have to check if the same ports work on our local system)

```
$ sudo NVIDIA-docker run -p 5000:5000 -it
didsrfda/victre-pipeline:v7 /bin/bash
```

- Without the website feature :

```
$ sudo NVIDIA-docker run -it didsrfda/victre-
pipeline:v7 /bin/bash
```

--You will enter the Compile folder by default.

NOTE: 2 docker containers cannot be hosted on the same ports. You would simply have to enter

run the scripts manually in that case. To enter without port allocation use:

```
$ sudo NVIDIA-docker run -it didsrfda/victre-
pipeline:v7 /bin/bash
```

2. The following script installs all CUDA dependencies inside Docker:

```
$ ./compile_mcgpu.sh
```

Select the keyboard layout type and Language.

3. At this stage you have all codes ready to run.

Now, you can enter the main pipeline folder:

```
$ cd /VICTRE/VICTRE_PIPELINE
```

THE VICTRE PIPELINE

1. This folder consists of the :

1. Breast Phantom Generation
2. Breast Compress
3. Breast Crop
4. Breast Mass
5. Lesion Insertion
6. MCGPU
7. DBT_Recon
8. ROI Extraction
9. run.py script : Master script which is the entry

point for running the pipeline

```
$ python run.py $breast_type
```

```
$lesion_or_no_lesion $projection_type
```

On the basis of the \$lesion_or_no_lesion and \$projection_type, corresponding simulation starts.

For a rule of thumb: if

```
$lesion_or_no_lesion = 0 --> No Lesion
```

if

```
$lesion_or_no_lesion = 1 --> Lesion
```

if

```
$projection_type = 0 --> FFDM
```

if

```
$projection_type = 1 --> DBT
```

if

```
$projection_type = 2 --> FFDM and DBT
```

There are 6 combinations in which a single breast_type can run in the pipeline. The user chooses which ones via these options.

10. run_scripts : Consists of 6 main scripts to run the pipeline depending on the projection and phantoms the user needs.

2. Calling the run.py python script will start the entire pipeline in the right order.

RUNNING THE VICTRE PIPELINE

There are 2 ways in which you can run the pipeline.

1. From Shell:

i) Run 1 phantom

```
$ cd /VICTRE/VICTRE_PIPELINE
```

```
$ python run.py dense 0 1 (This would start the  
simulation for Dense phantom with no lesion and  
FFDM)
```

By the end of the simulation you can see the outputs
in the by:

```
$ cd $OUTPUT_FOLDER
```

ii) Run multiple phantoms

```
$ cd /VICTRE/VICTRE_PIPELINE/run_full  
run_simulation contains the basic configurations to  
run the phantoms and their types  
$python run.py
```

2. Using the website :

```
$ cd /VICTRE/Website
```

```
$ flask run --host=0.0.0.0
```

This will show a similar message to:

```
root@081af44cfalc:/VICTRE/Website# flask run --  
host=0.0.0.0
```

```
* Serving Flask app "app"
```

```
* Running on http://0.0.0.0:5000/ (Press CTRL+C to  
quit)
```

By default Flask(MVC framework for running websites)
hosts your application on your local port.

The port forwarding used while entering the container
becomes useful in this scenario.

i) On Local host :

You will be able to see the working website in your
browser <http://127.0.0.1:5000/>

ii) On AWS:

It is required that AWS is configured to get the
traffic from port 80(http port)

In order to enable this:

a) Go on to your EC2 dashboard : <https://us-east-2.console.aws.amazon.com/ec2/>

b) Click on the instance which is currently running
the VICTRE container

c) In the Description down below, you will see the
security group. Click on it.

This will take you to the security group page
wherein you need to change the
inbound rules.

d) Down below in the description, click edit --> add
rule --> select http

e) Click save
f) Come back to the EC2 dashboard. There would be
public DNS IPv4 similiar to
eg.ec2-a-b-c-c.us-east-2.compute.amazonaws.com -->
This is the website link to the VICTRE container
g) Copy paste the link, and you should see the the
VICTRE webpage.
The ROI would automatically download on simulation,
you can also check the docker from your shell to
see it's current status.

VICTRE CONTAINER DETAILS

1. Base Layer of Ubuntu 16.04 is installed.
2. Installations of all the CPU packages.
3. Get all VICTRE codes.
4. Set Paths of a few packages and binaries.
5. Compiling of CPU codes.
6. Check if your system supports CUDA. If it does, CUDA and drivers will be installed and respective paths will be automatically be set.
If you do not have NVIDIA cards, the container WILL STILL BE BUILT FOR ALL CPU CODES.

It will show the following message:

-----Checking if CUDA can be installed-----

```
#####
#####
-----CUDA is not compatible with your platform-----
-----
-----You would need NVIDIA card support-----
-----
<-----Some options for Cloud compatible NVIDIA
GPU servers -->
    AWS : https://aws.amazon.com/ec2/instance-types/
    Azure: https://azure.microsoft.com/en-
us/pricing/details/virtual-machines/series/
    Google Cloud: https://cloud.google.com/gpu/

#####
#####
```

NOTE: Layer 2 cannot be built without layer 1, layer 3 cannot be built without layer 2 and so on. The order of layers is important. Refer the Dockerfile for more details of the layers.

WORKFLOW OF SIMULATION SCRIPTS INSIDE THE CONTAINER

All scripts can be found : /VICTRE/VICTRE_PIPELINE/run_scripts

1. l_ffdm.sh - > Phantom Generation -> Lesion insertion -> MCGPU
& flatfiled -> ROI mammo

2. l_dbt.sh 0 - > Phantom Generation -> Lesion insertion -> MCGPU
& flatfiled -> Extraction MCGPU & flatfile
-> ROI DBT

3. l_ffdm_dbt.sh - > Phantom Generation -> Lesion insertion ->
MCGPU & flatfiled -> Extraction MCGPU & flatfile
-> ROI DBT - > ROI DBT

4. nl_ffdm.sh - > Phantom Generation -> MCGPU & flatfiled -> ROI
mammo

5. nl_dbt.sh -> Phantom Generation -> MCGPU & flatfiled ->
Extraction MCGPU & flatfile
-> ROI DBT

6. nl_ffdm_dbt.sh - > Phantom Generation -> MCGPU & flatfiled ->
Extraction MCGPU & flatfile

-> ROI DBT - > ROI DBT

**Phantom Generation includes : Breast phantom generation +
Breast Compress + Breast Crop

BUILDING THE VICTRE CONTAINER

Once you are in the folder of the contents you would like your docker container to have,

you can run build the container using:

```
$ sudo docker build --tag $NAME_CONTAINER .
```

```
eg. $ sudo docker build --tag victre .
```

PUSHING THE CONTAINER ON DOCKER HUB

1. Login

```
$ sudo docker login --username=didsrfda  
password: Victrel23@abc
```

2. List all available Images on your system:

```
$ sudo docker images
```

The output will be similar to:

	REPOSITORY	TAG	IMAGE ID
CREATED			
	victre-pipeline	latest	2c4def87406a
24 minutes ago	8.6GB		
	ubuntu	16.04	dd6f76d9cc90
4 days ago			

3. Make a tag for your Image

```
$ sudo docker tag --IMAGE_ID UNAME/Image_name:version_no  
eg. $ sudo docker tag 2c4def87406a didsrfda/victre-
```

pipeline:v1

4. Push your Image to the docker hub.

```
$ sudo docker push $UNAME/IMAGE_name:version_no  
eg. $ sudo docker push didsrfda/victre-pipeline:v1
```

USEFUL SYMBOLIC LINKS FOR MOVING THE CONTAINER

i) Hetero, scattered, fatty and dense breast config files:
\$CONFIG_FILES=/VICTRE/VICTRE_PIPELINE/Config_files"

ii) Phantom generation :
\$BREASTPHANTOM_GEN=/VICTRE/VICTRE_PIPELINE/breastPhantom50"

iii) Breast Compress :
\$BREASTCOMPRESS=/VICTRE/VICTRE_PIPELINE/breastCompress"

iv) Breast Crop :
\$BREASTCROP=/VICTRE/VICTRE_PIPELINE/breastCrop"

v) Lesion Insertion :
\$LESION_INSERTION=/VICTRE/VICTRE_PIPELINE/Lesion_insertion"

vi) MCGPU Material Files :

\$MATERIAL_FILES=/VICTRE/VICTRE_PIPELINE/Config_files/Material_files"

vii) MCGPU :
\$MCGPU=/VICTRE/VICTRE_PIPELINE/MCGPU"

viii) MCGPU flatfiled
\$MCGPU_FLATFIELD=/VICTRE/VICTRE_PIPELINE/MCGPU_flatfield"

iiii) ROI :
\$ROI=/VICTRE/VICTRE_PIPELINE/ROI"

ix) DBT Reconstruction :
\$DBT=/VICTRE/VICTRE_PIPELINE/DBT_Recon"

x) All Output files:
\$OUTPUT_FILES

NOTE: If you system does not have the minimum space to install the
VICTRE Container, the container will
not be built.