**IAD Software Environment Setup**

**Key steps:**

**1. Install NVDIA Drivers**

**2.Install CUDA and Configure Cuda**

**3.Setup a virtual environment**

**4.Install packages and dependencies in the environment**

**5.Install the wheel files**

**6.Configure the database**

**System Requirements**

**AMD or Intel 64 Bit Processors**

**NVIDIA (>930M with CUDA)**

**Ubuntu (18.04 LTS)**

**1. Install NVDIA Drivers**

Go to software update>additional drivers

choose recommended NVIDIA Driver and update  
reboot the system

CUDA CONFIGURATION

**Cuda Installation**:

**open terminal and do**

sudo apt-get install g++ freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev libglu1-mesa libglu1-mesa-dev

Wget h[ttps://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86\_64/cuda-repo-ubuntu1804\_10.0.130-1\_amd64.deb](https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-repo-ubuntu1804_10.0.130-1_amd64.deb)

sudo apt-key adv --fetch-keys <https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub>

sudo dpkg -i cuda-repo-ubuntu1804\_10.0.130-1\_amd64.deb

sudo apt-key adv --fetch-keys <https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub>

sudo apt install cuda-10.0

echo 'export PATH=/usr/local/cuda-10.0/bin:$PATH' >> ~/.bashrc

echo 'export LD\_LIBRARY\_PATH=/usr/local/cuda-10.0/lib64:$LD\_LIBRARY\_PATH' >> ~/.bashrc

**Download cudnn-10.0-linux-x64-v7.6.5.32.tgz from nvidia developer site and the run**:

tar -xzvf cudnn-10.0-linux-x64-v7.6.5.32.tgz

sudo cp -P cuda/include/cudnn.h /usr/local/cuda-10.0/include

sudo cp -P cuda/lib64/libcudnn\* /usr/local/cuda-10.0/lib64/

sudo chmod a+r /usr/local/cuda-10.0/lib64/libcudnn\*

**Opencv Build and Installation**:

sudo apt-get install build-essential checkinstall cmake pkg-config yasm

sudo apt-get install libtiff5-dev libjpeg-dev

sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libxine2-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libv4l-dev

sudo apt-get install libtbb-dev

sudo apt-get install libqt4-dev libgtk2.0-dev

cd ~

wget -O opencv.zip https://github.com/opencv/opencv/archive/4.2.0.zip

wget -O opencv\_contrib.zip https://github.com/opencv/opencv\_contrib/archive/4.2.0.zip

unzip opencv.zip

unzip opencv\_contrib.zip

cd opencv-4.2.0

mkdir build

cd build

**Check for CUDA\_ARCH\_BIN for corresponding gpu from** [**https://en.wikipedia.org/wiki/CUDA#GPUs\_supported**](file:///opt/onlyoffice/desktopeditors/editors/web-apps/apps/documenteditor/main/index.html?_dc=0&lang=en-EN&frameEditorId=placeholder#GPUs_supported)**and Check path of opencv\_contrib carefully**

cmake -DWITH\_XINE=ON -D WITH\_OPENGL=ON -D WITH\_TBB=ON -DBUILD\_EXAMPLES=ON -DBUILD\_opencv\_world=ON -DBUILD\_opencv\_gapi=OFF -DWITH\_NVCUVID=OFF -DWITH\_CUDA=ON -DCUDA\_FAST\_MATH=ON -DWITH\_CUBLAS=ON -DWITH\_MKL=ON -DMKL\_USE\_MULTITHREAD=ON -DMKL\_WITH\_TBB=ON -DWITH\_CUFFT=ON -DCUDA\_ARCH\_BIN=6.1 -DOPENCV\_EXTRA\_MODULES\_PATH="/home/aindra/opencv\_contrib-4.2.0/modules/" ..

make -j7

sudo make install

**Boost Library Installation:**

sudo apt-get install libboost-all-dev

sudo usermod -a -G dialout $USER

**Spdlog Library Installation:**

git clone <https://github.com/gabime/spdlog.git>

cd spdlog && mkdir build && cd build

cmake .. && make -j

sudo make install

**Aindra Libraries Installation, Execute after successful installation of frontend and backend:**

**Create python 2.7 virtual environment**:

virtualenv productionenv (if virtual environment doesn’t exist)

source productionenv/bin/activate

Download all aindra packages in one directory named **packages**

pip install aindra\_ds-\* (latest .whl package in download section of data\_structures repository)

pip install aindra\_iad-\* (latest .whl package in download section of IAD\_algorithm repository)

pip install opencv-python==4.2.0.32

pip install wsi\_generator-\* (latest .whl package in download section of wsi\_generator repository)

pip install edge\_algorithm-\* (latest .whl package in download section of edge\_algorithm repository)

sudo apt-get install g++ freeglut3-dev build-essential libx11-dev libxmu-dev libxi-dev libglu1-mesa libglu1-mesa-dev

CAMERA SETUP

**Camera Software Installation : Download spinnaker-1.23.0.27-amd64-Ubuntu18.04-pkg.zip from download section of IAD\_algorithm repository**

Extract zip folder

cd spinnaker-1.23.0.27-amd64-Ubuntu18.04-pkg/spinnaker-1.23.0.27-amd64/

**follow instructions mentioned in readme file**

ENVIRONMENT SETUP

# Installation

## IAD Aindra Virtual Env Setup & Installation(AI Team)

Create a virtual environment:

virtualenv productionEnv

OR

~~virtualenv --no-site-packages -p /usr/bin/python2.7 kivyinstall~~

CUDA installation

and Tensorflow installation related libraries if any AI Team need do installation

# Atfer finish above setup will follow below steps:

## MySQL Installation (5.7 Version)

Refer to <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-16-04> for MySQL Installation details

sudo apt-get update

sudo apt-get install mysql-server

sudo mysql\_secure\_installation

Test MySQL installation

systemctl status mysql.service

mysqladmin -p -u root version

Should display a result:

Server version 5.7.23-0ubuntu0.16.04.1

## IAD MySQL Related Aindra Software Installation

Clone the git repository

Open the user\_creation\_commands file from IAD\_Database source folder and create iad user , iad password and iadmaster database.

linux> sudo mysql

then type commands given in the file

mysql -u<userid> -p <password> <dbname> < <schema file>

Then run the following to install IAD Software database tables:

mysql -u iadadmin -piadadmin iad\_master < iadmaster\_schema.sql

SET GLOBAL validate\_password\_policy=LOW;

**Start the virtual environment:**

linux prompt> source productionEnv/bin/activate

You can now see the below prompt of virtualenvironment name

( productionEnv) >

**Now, Install the required software libraries inside this virtual environment:**

sudo apt-get update

pip install python-pip

pip install -–upgrade protobuf==3.6.0 // pip install protobuf==3.6.0

sudo apt-get install -y libmysqlclient-dev libsdl-image1.2-dev libsdl-mixer1.2-dev libsdl-ttf2.0-dev libsmpeg-dev libsdl1.2-dev libportmidi-dev libswscale-dev libavformat-dev libavcodec-dev zlib1g-dev

sudo add-apt-repository ppa:mc3man/bionic-media

sudo apt-get update

sudo apt-get install ffmpeg

sudo apt install -y libav-tools # if fails use : sudo apt install libvips-tools

sudo apt-get install -y libgstreamer1.0 gstreamer1.0-plugins-base gstreamer1.0-plugins-good

sudo add-apt-repository ppa:kivy-team/kivy

sudo apt-get install python3-kivy

sudo apt-get install build-essential

pip install -U --force-reinstall Cython==0.25.2

pip install git+https://github.com/kivy/kivy.git@1.11.1

(if python3.6 edit config file and add this line

Config.setdefault('kivy', 'log\_name', 'kivy\_%%y-%%m-%%d\_%%\_.txt')

)

Config.setdefault('kivy', 'log\_name', 'kivy\_%%y-%%m-%%d\_%%\_.txt

pip install KivyCalendar

pip install image

pip install inotify

pip install reportlab

pip install SQLAlchemy

pip install mysql-python

pip install scipy==0.17.0

pip install --upgrade pip

pip install opencv-python==4.2.0.32

pip install oscpy

python -m pip install pygame

pip install SQLAlchemy-Paginator

pip install retrying

pip install git+https://gitlab.com/kivymd/KivyMD.git

cd ~/.kivy/garden

git clone https://github.com/kivy-garden/garden.filebrowser

pip install /home/aindra/Documents/aindra\_ds-1.2.1-py2-none-any.whl # abhay wheel file

export PYTHONPATH=/home/aindra/IAD/iad-frontend-app/

cd ~/IAD/iad-frontend-app/src/iad/aindra/frontend

python IAD\_Server.py

BACK-END Setup

---------------------------------------------------------------------

echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg] https://packages.cloud.google.com/apt cloud-sdk main" | sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list

sudo apt install curl

curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key --keyring /usr/share/keyrings/cloud.google.gpg add -

sudo apt-get update

sudo apt-get install google-cloud-sdk

which gcloud

sudo python -m pip install google-cloud

pip install PyMySQL

pip install google-cloud-storage

which gsutil

export CLOUD\_SDK\_REPO="cloud-sdk-$(lsb\_release -c -s)"

echo "deb http://packages.cloud.google.com/apt $CLOUD\_SDK\_REPO main" | sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list

curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

echo 0|sudo tee /sys/module/usbcore/parameters/usbfs\_memory\_mb

### **How do I get set up?**

* On the IAD Device, Open the Browser and Login to Bitbucket Release Repository
* Download the software zip file for Iad\_fornt\_end and IAD\_BACKEND in latest OR
* git clone -b develop https://palani\_s@bitbucket.org/AdarshNatarajan/iad-backend-app.git //perticular branch
* git clone -b feature-develop https://palani\_s@bitbucket.org/AdarshNatarajan/iad-frontend-app.git
* Go To ~home directory of Linux
* Unzip the Iad\_fornt\_end and IAD\_BACKEND files
* IAD\_BACKEND software is ready to be launched/started
* Add the following entry into crontab: 05 10 \* \* \* /media/aindra/GIT\_IADBACKEND\_WORKING/iad-backend-app/IAD\_BACKEND\_INTERFACE/iad/aindra/backend/run\_iad\_backend\_as\_cron.sh > /dev/null 2>&1

Before starting IAD\_FORNTEND Follow below steps:

To add a new environment variable in Ubuntu , use the following steps:

1.Open a terminal (by pressing Ctrl+Alt+T)

` 2.sudo -H gedit /etc/environment

3.Type your password

3.Edit the text file just opened:

4.e.g. if you want to add ENCRYPT\_KEY = 'iadadmin', then just write ENCRYPT\_KEY = 'iadadmin' in a new line like..below

//VX(visniox)101(deviceId)hcg(organizationcode)2102(day/month)

ENCRYPT\_KEY\_hcg ='VX101hcg2102'

MASTER\_DB\_USER = 'iadadmin'

MASTER\_DB\_PWD = 'iadadmin'

DATABASE = 'iad\_master'

5.Save it

6.Once saved, logout and login again.

Open Iad front end configure file and change below varibles:

DEVICE\_ID = 'VX01191101'

DEVICE\_MODEL\_NUMBER = 'IAD-01-01'

DEVICE\_CONFIGURE\_DATE = '2019-09-29' # Syntax is yyy-mm-dd

ORGANIZATION\_NAME = 'hcg'

PARENT\_LOGS\_DATA\_FOLDER='/home/aindra/CCI/IAD\_DEMO/demologs/'

PARENT\_SRC\_FOLDER='/home/aindra/IAD\_DEMO/iad-frontend-app/src/iad/aindra/'

MYSQL\_DB\_PATH ="mysql+pymysql://iadadmincci:[iadadmincci@localhost](mailto:iadadmincci@localhost):3306/iad\_master\_cci?charset=utf8mb4"

and save it

Just turning off notification of updates will not be enough. There are several background jobs that run apt-get update (including a nightly cron job if I remember right).

To stop the pop-up run update-manager

Then in settings, set Update automaticly to 'Never' and Notify of new version to 'Never'

I suggest leave security at immediately and, setting "others" to every two weeks.

You also need to make sure your not running apt-get update anywhere else. This can happen when you install software, run update-manager, or in a cron job.

If that doesn't work for you:

You can do this by

editing /etc/apt/apt.conf.d/10periodic and changeing

APT::Periodic::Update-Package-Lists "1";

to

APT::Periodic::Update-Package-Lists "0";

Last Resort

You can also do apt-get remove update-manager to remove the feature all together.

# How to Disable/Enable Automatic Error Reporting in Ubuntu 18.04 LTS

sudo service apport stop

### Disable Apport at Boot

sudo nano /etc/default/apport //enabled=1

### Uninstall Apport

sudo apt-get purge apport

OR

In the right pane, look up for the Problem Reporting feature, whose value is ‘Automatic’ by default.

Click on Problem Reporting to customize the error reporting feature that reports the technical problems to Canonical anonymously

Here you can switch off the “Send error reports to Canonical”button in order to disable error reporting altogether. Or you can select if you want the system to automatically send error reports or else to show a dialog for each error before sending its report to Canonical.

After create Desktop image following steps:

Launcher Instructions:

=======================

sudo apt install --no-install-recommends gnome-panel

gnome-desktop-item-edit ~/Desktop/ --create-new

# Disable the on screen keyboard when using a touch screen

### [Block Caribou - GNOME Shell Extension](https://www.google.com/url?sa=t&amp;amp;rct=j&amp;amp;q=&amp;amp;esrc=s&amp;amp;source=web&amp;amp;cd=1&amp;amp;ved=2ahUKEwinp8jO8-vnAhWtxzgGHbgvBwMQFjAAegQIAhAB&amp;amp;url=https%3A%2F%2Fextensions.gnome.org%2Fextension%2F1326%2Fblock-caribou%2F&amp;amp;usg=AOvVaw2NDmUN55zg2Y9jVY5u_o9L)