

Configuration Manual

MSc Research Project
Cloud Computing

Taofeek-Femi Abdullahi

Student ID: 19104219

School of Computing
National College of Ireland

Supervisor: Manuel Tova-Izquierdo

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Taofeek-Femi Abdullahi
Student ID:	19104219
Programme:	Cloud Computing
Year:	2020
Module:	MSc Research Project
Supervisor:	Manuel Tova-Izquierdo
Submission Due Date:	17/08/2020
Project Title:	Configuration Manual
Word Count:	1173
Page Count:	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	16th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Taofeek-Femi Abdullahi
19104219

1 Introduction

In this configuration manual, we detail the steps taken to fully replicate the runtime environment for ADEPS. We will also discuss the prerequisites that needs to be installed on the system. In this manual, we will be using a Linux virtual machine provisioned on VirtualBox and we will show all the steps required for ADEPS to be fully functional. Our basic virtual machine specifications are described below:

- Operating System: Linux Ubuntu 18.04 (64-bit)
- RAM: 4096MB
- Disk Size: 10GB

A virtual machine with the specifications detailed above should be sufficient for ADEPS.

2 Prerequisites

ADEPS was designed to be as interactive as possible when carrying out tasks such as deploying custom runtimes to IBM OpenWhisk.

- A user will be required to have familiarity around the Linux terminal. Also basic debugging tools and knowledge of basic Linux command are also advised.
- A user will be required to have basic knowledge of Docker and how containers work. User should also be familiar with Dockerhub and how the repository works. In addition, a user is required to be able to write and understand a Dockerfile.
- A user should understand serverless computing concepts and how to navigate the IBM OpenWhisk platform.

3 Required Libraries and Modules

The Linux machine that ADEPS will be run on should have the following libraries and modules installed.

3.1 Git

For an easy install of ADEPS, it is advisable to install Git on your local machine. We can install git using apt-get.

Run these commands:

```
sudo apt-get update
sudo apt-get install git
```

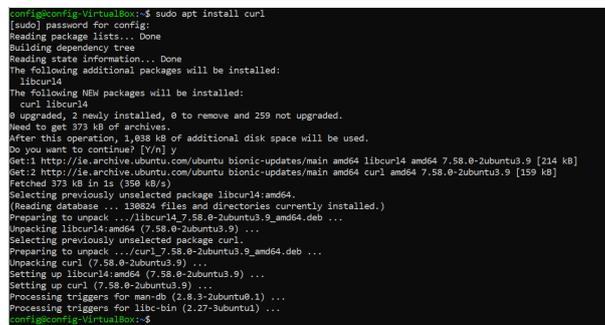
3.2 Curl

Curl will be used in combination with Git for a smooth and easy install of ADEPS. Curl does not come pre-installed on a Linux machine and will need to be installed if not already installed.

Run this command:

```
sudo apt install curl
```

Your install screen should look like Figure 1



```
cmpr@cmpr14:~/VirtualBox$ sudo apt install curl
[sudo] password for cmpr:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 259 not upgraded.
Need to get 373 kB of archives.
After this operation, 1,038 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ia.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcurl4 amd64 7.58.0-2ubuntu3.9 [214 kB]
Get:2 http://ia.archive.ubuntu.com/ubuntu bionic-updates/main amd64 curl amd64 7.58.0-2ubuntu3.9 [159 kB]
Fetched 373 kB in 1s (350 kB/s)
Selecting previously unselected package libcurl4:amd64.
(Reading database ... 138204 files and directories currently installed.)
Preparing to unpack .../libcurl4_7.58.0-2ubuntu3.9_amd64.deb ...
Unpacking libcurl4:amd64 (7.58.0-2ubuntu3.9) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.58.0-2ubuntu3.9_amd64.deb ...
Unpacking curl (7.58.0-2ubuntu3.9) ...
Setting up libcurl4:amd64 (7.58.0-2ubuntu3.9) ...
Setting up curl (7.58.0-2ubuntu3.9) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
cmpr@cmpr14:~/VirtualBox$
```

Figure 1: Curl install screen

3.3 Python 3 and Pip

ADEPS is written in Python and will therefore need python3 installed on the local machine. Some machines already come with python pre-installed and others don't. To find out if you have a python installation, please run **python3 --version**. If you do not get results, you will have to install python manually.

Run this command:

```
sudo apt-get install python3.6
```

Once you have python installed, please proceed to install pip3 for python3.

Run this command:

```
sudo apt install python3-pip
```

Your install screen should look like Figure 2

Finally, install the Docker SDK for python using **pip3 install docker**

3.4 Docker

One of the major functionalities of ADEPS is being able to build a Docker image and publish to Dockerhub. In order for this functionality to work, Docker will have to be installed on the local machine. Even though ADEPS uses the Docker SDK in its internal operations, it still needs to hand over some control to the docker engine on the local

```

root@config-VirtualBox:~# sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  build-essential cpp-7 dh-python dpkg-dev fakeroot g++ g++-7 gcc gcc-7 gcc-7-base gcc-8-base libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan4 libatomic1 libc-dev-bin libc6-dbg libc6-dev
  libc6-i386 libcilkrts5 libexpat1-dev libfakeroot libgcc-7-dev libgcc1 libgomp1 libitm1 liblsan0 libmpx2 libpython3-dev
  libpython3.6 libpython3.6-dev libpython3.6-minimal libpython3.6-stdlib libquadmath0 libstdc++-7-dev libstdc++6
  libtsan0 libubsan0 linux-libc-dev make manpages-dev python-pip-whl python3-dev python3-distutils python3-lib2to3
  python3-setuptools python3-wheel python3.6 python3.6-dev python3.6-minimal
Suggested packages:
  gcc-7-locales debian-keyring g++-multilib g++-7-multilib gcc-7-doc libstdc++6-7-dbg gcc-multilib autoconf automake
  libtool flex bison gcc-doc gcc-7-multilib libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan4-dbg
  liblsan0-dbg libtsan0-dbg libubsan0-dbg libcilkrts5-dbg libmpx2-dbg libquadmath0-dbg libstdc++6-dbg libstdc++-7-dbg
  make-doc python-setuptools-doc python3.6-venv python3.6-doc binfmt-support
The following NEW packages will be installed:
  build-essential dh-python dpkg-dev fakeroot g++ g++-7 gcc gcc-7 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan4 libatomic1 libc-dev-bin libc6-dev libcilkrts5 libexpat1-dev libfakeroot
  libgcc-7-dev libitm1 liblsan0 libmpx2 libpython3-dev libpython3.6-dev libquadmath0 libstdc++-7-dev libtsan0
  libubsan0 linux-libc-dev make manpages-dev python-pip-whl python3-dev python3-distutils python3-lib2to3 python3-pip
  python3-setuptools python3-wheel python3.6-dev
The following packages will be upgraded:
  cpp-7 gcc-7-base gcc-8-base libc6 libc6-dbg libc6-i386 libgcc1 libgomp1 libpython3.6 libpython3.6-minimal
  libpython3.6-stdlib libstdc++6 python3.6 python3.6-minimal
14 upgraded, 39 newly installed, 0 to remove and 245 not upgraded.
Need to get 87.3 MB/202 MB of archives.
After this operation, 287 MB of additional disk space will be used.
Do you want to continue? [Y/n] y

```

Figure 2: Pip3 install screen

machine. The Docker team has made it easy to install Docker on your machine and it can be done by running this simple command that executes a shell script:
curl -sSL https://get.docker.com/ — sh¹

You should see a screen like the one in Figure 3.

```

root@config-VirtualBox:~# curl -sSL https://get.docker.com/ | sh
# Executing docker install script, commit: 26f9636cf2b3f9a00498ac43694bf1c7d9ce16c
# sudo -E sh -c apt-get update -qq >/dev/null
# sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null
# sudo -E sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | apt-key add -qq - >/dev/null
Warning: apt-key output should not be parsed (stdout is not a terminal)
# sudo -E sh -c echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable" > /etc/apt/sources.list.d/docker.list
# sudo -E sh -c apt-get update -qq >/dev/null
# [ n ]
# sudo -E sh -c apt-get install -y -qq --no-install-recommends docker-ce >/dev/null
# sudo -E sh -c docker version
Client: Docker Engine - Community
 Version: 19.03.12
 API version: 1.40
 Go version: golang1.10.8
 Git commit: 48a66213fe
 Built: Mon Jun 22 15:45:36 2020
 OS/Arch: linux/amd64
 Experimental: false

Server: Docker Engine - Community
 Engine:
  Version: 19.03.12
  API version: 1.40 (minimum version 1.12)
  Go version: golang1.10.8
  Git commit: 48a66213fe
  Built: Mon Jun 22 15:44:07 2020
  OS/Arch: linux/amd64

```

Figure 3: Docker install screen

Test your Docker install by running **docker --version**. You should see the version of your newly installed Docker. Docker requires all commands to run in sudo mode. Which is why on a fresh install, commands like **docker ps** will show permission errors. In order for ADEPS to work without errors, sudo privileges will have to be given to the docker group. Please run this command to make these changes **sudo usermod -aG docker \$USER**. Please restart the machine for these changes to take effect.

3.5 IBM CCloud CLI

Deploying serverless functions to IBM OpenWhisk is one of the features of ADEPS. For this reason, the IBM Cloud CLI will have to be installed on the local machine. Run this command to install the IBM CCloud CLI:
curl -sL https://ibm.biz/idt-installer — bash²
 You should see a screen like the one in Figure 4.

¹getDocker : <https://get.docker.com/>
²IBM Cloud CLI: <https://cloud.ibm.com/docs/cli>

```

config@config-VirtualBox:~$ curl -sL https://ibm.biz/ibm-cli-installer | bash
[main] --=[ IBM Cloud Developer Tools for Linux/MacOS - Installer, v1.2.3 ]==--
[install] Starting Update...
[install] Note: You may be prompted for your 'sudo' password during install.
[install_deps_with_apt_get] Checking for and updating 'apt-get' support on Linux
[sudo] password for config:
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:3 http://ie.archive.ubuntu.com/ubuntu bionic InRelease
Get:4 http://ppa.launchpad.net/git-core/ppa/ubuntu bionic InRelease [20.7 kB]
Get:5 http://ie.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:6 http://ppa.launchpad.net/git-core/ppa/ubuntu bionic/main amd64 Packages [3,036 B]
Get:7 http://ie.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:8 http://ppa.launchpad.net/git-core/ppa/ubuntu bionic/main i386 Packages [3,036 B]
Get:9 http://ppa.launchpad.net/git-core/ppa/ubuntu bionic/main Translation-en [2,252 B]
Fetched 281 kB in 1s (338 kB/s)
Reading package lists... Done
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://ppa.launchpad.net/git-core/ppa/ubuntu bionic InRelease
Hit:3 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:4 http://ie.archive.ubuntu.com/ubuntu bionic InRelease
Get:5 http://ie.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:6 http://ie.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 252 kB in 1s (300 kB/s)
Reading package lists... Done
[install_deps_with_apt_get] Installing/updating external dependency: curl
[install_deps_with_apt_get] Installing/updating external dependency: git
[install_deps_with_apt_get] Installing/updating external dependency: docker

```

Figure 4: IBM CLI install screen

4 Cloud Accounts

The only accounts that will be needed for ADEPS to work are Dockerhub and IBM Cloud. A Dockerhub account is required so that ADEPS can push a created image to or if an image already exists, IBM Openwhisk will need that image as part of its deployment process. On the other hand, an IBM Cloud account is needed to deploy and invoke serverless functions in. IBM OpenWhisk allows 5million invocations for free with a Lite account.

5 Deploying a Function

The following steps will have to be taken to successfully deploy a serverless function.

- Create a Dockerhub repository or make sure you have a repository already created.
- Login to the IBM Cloud CLI using this command: **ibmcloud login -a cloud.ibm.com -o “your-email” -s “dev”**. You will be provided with an input prompt to enter your password. Once you have been authorized, choose your desired region.
- Create a new folder that will contain your serverless function and Dockerfile. These 2 files are required for ADEPS to work. Also, it is advised that these 2 files are in the same location.
Run these commands:
mkdir folderName
touch action.py
touch Dockerfile
- Edit your new files. Your Dockerfile file should contain all the custom libraries and modules you will be using in your serverless function.
Your serverless function should contain only one function which should be named ”main”. It should also return a JSON object.
- Clone ADEPS into the same folder that contains your serverless function and Dockerfile. Run this command:
curl -OL https://raw.githubusercontent.com/adeprsrepo/adeprs/master/adeprs.py

```
config@config-virtualBox:~/action$ ls
action.py  adeps.py  Dockerfile
```

Figure 5: Deploy folder example

Your new folder should now look like figure 5.

- We are now ready to deploy. run the deploy command with the required arguments.
 - -d tells the application that you want to build a new Docker image from your Dockerfile.
 - -r is your Dockerhub reponame.
 - -t is your image tag name.
 - -f is the name of your serverless function in the folder.
 - -a is the name that the function will be saved as on IBM OpenWhisk

Run the command: **python3 adeps.py deploy -r yourRepoName -d true -t yourTagName -f yourFunctionName -a nameToSaveAs**

- Once the deploy command is run, the application prompts you for your Dockerhub username and password. An image is built and pushed to the repo (if you indicated when running the deploy command). Your new action will now be available for invocation on the IBM Cloud Functions dashboard.