

## Contents

I.	Part I: Installing Host Machine Required Software .....	3
1.	<b>Host machine requirements</b> .....	3
2.	Installing WSL, Ubuntu and Git for Linux .....	4
2.1.	Check versions for WSL, Ubuntu, and Git.....	4
	 Open Powershell As Admin and Ck wsl version (mp4).....	4
2.2.	Uninstall WSL/Ubuntu (Optional) .....	5
2.3.	Install WSL2 and Ubuntu after uninstalling WSL/Ubuntu .....	5
2.4.	Setup Ubuntu Username and Password.....	6
3.	Linux Administrator Operations.....	6
3.1.	Install git for Linux (if not installed).....	6
4.	Verify LXSSMANAGER is running.....	7
	This service makes available a subsystem that deals with Linux executables (LX's) called ELF's.	
	 Start Service LXSS (mp4) .....	7
5.	Clean Up Prior to Installing Docker .....	7
5.1.	Uninstall Docker Desktop (if installed) .....	7
5.2.	Check for Docker for Linux .....	7
5.3.	Uninstall Docker (if required) .....	7
5.4.	Remove Existing Containers/Images (if required).....	8
6.	Install Docker in Ubuntu .....	8
6.1.	Obtain a Docker ID.....	8
	 Create docker hub account (mp4).....	8
6.2.	Install Docker .....	8
6.3.	Configure the Linux host machine to work better with Docker .....	8
6.4.	Verify the Docker installation.....	8
6.5.	Important Docker Commands:.....	9
II.	Part II: Build and Run the Test Suite .....	10
1.	Build the Test Runner .....	10
1.1.	First Time Building the Test Runner .....	10

1.2.	<b>Agree to the EULA:</b>  Instructions on agreeing to the EULA (PDF): For the test runner to start you are required to agree to the EULA. ....	10
1.3.	<b>Update or rebuild the Test Runner see: Update Test Runner</b> .....	11
2.	Import Content.....	11
2.1.	Clean Up Docker in the event of a problem .....	11
3.	Configure access to the test runner:.....	12
4.	Run a Test (“Phase 2”).....	12
4.1.	<b>To run a test (Single Device (Host and DUT running on the same device :</b> .....	12
	 <b>Running a Test Single Device (mp4)</b> .....	12
4.2.	<b>To run a test on a 2-device setup</b> .....	13
5.	Analyze a Recording (“Phase 3”).....	14
5.1.	Build the image.....	14
5.2.	Add Results .....	14
5.3.	<b>Configure the Observation Framework</b>  <b>Configure the Observation Framework (mp4)</b> 14	14
5.4.	Run the analysis  <b>Analyze a Recording (mp4)</b> .....	14
5.5.	Getting the analysis results .....	15
5.6.	Debugging .....	15
6.	<b>Additional Notes:</b> .....	15
6.1.	Steps to Create an Inbound Firewall Rule.....	15
6.2.	Configuring IP address: .....	16
6.3.	Adding the observation-config.ini file.....	16
7.	Helpful Links .....	17

# Installation of the WAVE Streaming Media Test Suite – Devices for Docker CE for Linux and Windows with WSL2

These instructions are for the user intending to install the *WAVE Streaming Media Test Suite – Devices* on a Windows 10/11 machine, running Ubuntu Linux on that machine using the Windows WSL2 services. These instructions have been tested specifically for those installation parameters, although the knowledgeable developer will be able to adapt the instructions to their preferences, such as running a different Linux distro instead of Ubuntu.

The WAVE Project offers the *Test Suite* and these instructions as open source (for details see LICENSE.md on the main Github Deployment page, <https://github.com/cta-wave/dpctf-deploy>). However, we regret that we cannot provide general technical support. Bugs or feature requests may be filed at <https://github.com/cta-wave/dpctf-deploy/issues>.

## Part I: Installing Host Machine Required Software

### 1. Host machine requirements

See

<https://github.com/cta-wave/dpctf-deploy?tab=readme-ov-file#host-machine-requirements>

**GENERAL NOTES: For additional information and instructions see [README](#)**

**<https://github.com/cta-wave/dpctf-deploy>**

- These instructions are for Windows 10 build 19041+ or Windows 11 build 20262+, WSL2 and a recent Ubuntu distribution (recommend 20.04 or higher).
- These instructions are NOT for use with Docker Desktop.
- It is recommended you update Windows and then existing or newly installed WSL2, Linux and Docker packages before starting to install and build the test suite.
- It is recommended that the Host PC have 2 monitors to view the instructions during the installation of the required packages and the Test Suite Build. It also allows the installer to copy and paste commands from the instructions and external sources.
- There have been instances where previous installations of WSL2, Ubuntu and Docker have caused issues when building and/or running the Test Suite. If you run into issues during the installation or running the Test Suite, consider uninstalling WSL, Ubuntu, Git for Linux and Docker and reinstalling them and delete the Test Suite before reinstalling.
- WSL2 is required
- It is highly recommended you start with a clean installation by uninstalling all prior versions of WSL2, Ubuntu and Docker for Linux and reinstall them. However, if WSL2, Ubuntu (and Git for Linux) are already installed, working, and up to date (see section 2.1 below to check), you may choose to skip to section 4 (LSXX) followed by section 5.4 below.

- **IMPORTANT NOTE:** Before leaving your computer for an extended period, turning off your PC, or at the end of the day (in case of an update or crash) always stop Docker by running `docker stop dpctf` to avoid corruption of docker files (see commands below). Restart the service and container to run the Test Suite.
  - In case you forget or the Test Runner stops working, when all else fails, delete the WAVE folders (`dpctf-deploy-master` and `device-observation-framework` folders), and delete the container and image. Then follow the instructions for downloading and installing the Test Runner. See Part II, 2.1 for more details.

## 2. Installing WSL, Ubuntu and Git for Linux

### 2.1. Check versions for WSL, Ubuntu, and Git



[Open Powershell As Admin and Ck wsl version \(mp4\)](#)

#### 1. WSL2 is required to run the test suite using

Docker CE. To check your WSL version, open a Windows Powershell as an administrator, type `wsl` and hit enter. You may need to enter your password for Linux. It will return with the `$` prompt if you have Ubuntu (or another Linux distro). If it does, type

```
wsl.exe --list --verbose
```

If you are running WSL 2 with one or more Linux distributions you should see something like

```
NAME      STATE   VERSION
*Ubuntu   Running 2
Ubuntu-22.04 Stopped 2
```

If you do not see Version 2 (“Version 2” refers to WSL version), go to [Install WSL 2 and Ubuntu](#) below. If you see WSL Version 2 but not an Ubuntu/Linux distribution, install Ubuntu.

#### 2. **Check Ubuntu is installed and update.**

- In a Linux terminal Type

```
lsb_release -a
```

Your version appears on the “Description” line. If it is not installed, see section 2.3 “Install WSL2 and Ubuntu” below.

- To update Ubuntu see <https://ubuntu.com/tutorials/upgrading-ubuntu-desktop#1-before-you-start>

Be sure to follow the first 4 steps

- Check version of Git for Linux

```
git --version
```

If not installed, go to section 3.1 “Install git for Linux”.

## 2.2. Uninstall WSL/Ubuntu (Optional)

**NOTE: If you decide to start with a clean installation of WSL and Ubuntu (recommended), you must first uninstall them.**

1. Uninstall WSL: There are many sites on uninstalling WSL. Examples:
  - Windows 11: See <https://pureinfotech.com/uninstall-wsl-windows-11/>
  - Windows 10: See <https://pureinfotech.com/uninstall-wsl2-windows-10/>
2. To completely uninstall an Ubuntu distribution from WSL (without removing WSL2): See: <https://www.windowcentral.com/how-completely-remove-linux-distro-wsl>

## 2.3. Install WSL2 and Ubuntu after uninstalling WSL/Ubuntu

This step may not be required if you have up to date versions of WSL2 and Ubuntu already installed and choose not to uninstall and reinstall them. However, it is recommended to start with a clean installation to avoid complications arising from prior installations.

1. Open a Windows Powershell as administrator and typing  

```
wsl --install
```

Ubuntu may be installed along with WSL2.  
This operation will take some time.
2. Turn on “Windows subsystem for Linux” (wsl2) and “Virtual Machine Platform” in Features by navigating to *Control Panel | Programs | Turn Windows feature on or off*, or by typing “Turn Windows Features On/Off” in the search bar; then check the boxes for both.

Screenshot:  [Turn Windows Features On or Off \(PDF\)](#)

- If the Virtual Machine Platform feature does not exist or won't let you enable it, see <https://support.microsoft.com/en-us/windows/enable-virtualization-on-windows-11-pcs-c5578302-6e43-4b4b-a449-8ced115f58e1>
3. Restart computer
  4. Set Default to WSL 2  
In a Powershell as Administrator, Type  

```
wsl --set-default-version 2
```

This ensures that any Linux distribution will use WSL2.
  5. Check to see that Ubuntu was installed when WSL2 was installed. Type  

```
lsb_release -a
```

It will show the version #. Ignore the message “No LSB modules are available”.
  6. If Ubuntu is not installed, you will need to install it. See the following link <https://linuxsimply.com/linux-basics/os-installation/wsl/ubuntu-on-wsl2/>. It will show you

how to load a Linux Distribution. Ubuntu 22.04 is preferred. **This link also covers enabling WSL 2, and setting WSL 2 to default above:**

## 2.4. Setup Ubuntu Username and Password

1. Open a Linux terminal from the start menu, locate Ubuntu and click on it. This will open a Linux bash terminal.
2. If this is the first time you run the terminal, follow prompts to add a Linux User Name and PW,

**Note that these are not your Windows/user name or PW.**



[Setup Ubuntu Username\\_Password and Check version \(mp4\)](#)

3. **Update/Upgrade packages:**

Type

```
sudo apt update && sudo apt upgrade
```

---

**NOTE: Now that WSL2 and Ubuntu are installed, unless otherwise indicated all commands are performed using a Linux (Ubuntu) bash terminal as an administrator.**

- **To open a Linux (Ubuntu) terminal from the start menu, locate the Ubuntu app and click on it. This will open an Ubuntu bash terminal or Type “Ubuntu” in the search bar and select it.**

**NOTE: When running the commands that follow, unless you have set your username with root privileges you may get “command not found”. If you do, preface the command with “sudo”.**

**NOTE: Depending on whether your \$Path includes the Path to your Unix directory, some Powershell commands such as <netsh> and <ipconfig> may require them to be prefaced with “./” e.g. <./netsh...> or <./ipconfig>.**

---

## 3. Linux Administrator Operations

### 3.1. Install git for Linux (if not installed)

1. Change **directory:**

Type `cd ~` and hit enter.

Then Type or copy

```
sudo apt install git-all
```

2. Configure git as per <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>
3. Run

```
git config --global user.name "John Doe"  
git config --global user.email johndoe@example.com
```

## 4. Verify LXSSMANAGER is running.

This service makes available a subsystem that deals with Linux executables (LX's) called ELF's.



[Start Service LXSS \(mp4\)](#)

Press WIN+R

1. Type: services.msc
2. Find LXSSMANAGER in the list
3. If this service is not running, right click it and select START or RESTART

**NOTE: You must repeat this after any restart of your PC.**

## 5. Clean Up Prior to Installing Docker

In this step, we check versions and uninstall items if necessary. If this is a new install on a clean machine, these steps are not necessary, and you can skip to section 6 "[Install Docker in Ubuntu](#)".

### 5.1. Uninstall Docker Desktop (if installed)

Use of Docker Desktop is not supported.

If Docker Desktop is installed it should be uninstalled using Windows uninstall. In the search bar enter "add or uninstall programs" and look for Docker Desktop.

- Uninstall (Click on the 3 dots, then click uninstall).

### 5.2. Check for Docker for Linux

Docker for Linux is required.

Check if Docker for Linux (not Docker Desktop) is installed.

- In a Linux (bash) terminal type

```
docker -v
```

or

```
docker --version
```

It should return the Docker version e.g. "Docker version 26.1.4, build 5650f9b" if it is installed.

**NOTE: It is highly recommended that you uninstall and reinstall Docker unless you know it is already working correctly and/or it is needed for other work. Remnants from prior installations may cause issues.**

### 5.3. Uninstall Docker (if required)

If a version of Docker is installed and you wish to start with a new install, see

<https://docs.docker.com/engine/install/ubuntu/#uninstall-old-versions>

<https://docs.docker.com/engine/install/ubuntu/#uninstall-docker-engine>

## 5.4. Remove Existing Containers/Images (if required)

If this is a re-install of the Test Suite, you should remove existing Docker containers and images before re-installing see <https://www.digitalocean.com/community/tutorials/how-to-remove-docker-images-containers-and-volumes>

# 6. Install Docker in Ubuntu

## 6.1. Obtain a Docker ID

Before installing Docker you will need first to obtain a Docker ID and PW at <https://hub.docker.com/signup>



[Create docker hub account \(mp4\)](#)

## 6.2. Install Docker

To install Docker for Ubuntu, See <https://docs.docker.com/engine/install/ubuntu/> (for other distributions see list at left on the page)

**NOTE:** If you get a **“failed public key” error message** when installing Docker with an Ubuntu 22.04 distro, follow the instructions at:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-22-04>.

You can also check out <https://stackoverflow.com/questions/60137344/docker-how-to-solve-the-public-key-error-in-ubuntu-while-installing-docker> for additional suggestions.

## 6.3. Configure the Linux host machine to work better with Docker

**Follow these steps after Docker installation.**

See <https://docs.docker.com/engine/install/linux-postinstall/> to set up Docker. These instructions will also allow you to run Docker as root, without prefacing Docker commands with “sudo”. See the link for full information. Some IT departments may discourage running as root for security reasons.

## 6.4. Verify the Docker installation

1. To ensure installation was successful in a bash terminal

Run

```
docker --version
```

and

```
docker-compose --version
```

Both should return the version of the install.

If docker-compose is not installed, run

```
sudo apt-get update
```

```
sudo apt-get install docker-compose-plugin
```

Then recheck the version to confirm it is installed.

2. To ensure Docker is running correctly, in bash terminal Run

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

This should return “Hello from Docker!” if successful



[Confirm Docker Working\\_Hello World \(mp4\)](#)

## 6.5. Important Docker Commands:

**NOTE: Before leaving your computer for an extended period of time, turning off your PC or at the end of the day (in case of an update) always stop Docker to avoid corruption of docker files (see commands below). Restart the service or container to run the Test Suite.**

If you forget to stop the service/container before shutting down or restarting your computer (or an update restarts it) and Docker fails to restart, try each of the following in order:

- restart LSXX
  - stop and restart the Docker Service
  - restart the computer
  - If that fails, clean out Docker and rebuild the Test Runner. See II section 1a “To clean out Docker and rebuild the Test Suite“ below.
1. Docker Login/Logout: Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to <https://hub.docker.com> to create one.

```
docker login
```

```
docker logout
```

2. Start / Stop Docker. Use this command to start the Docker service.

Run

```
sudo service docker start
```

```
sudo service docker stop
```

3. To start and stop a single container

Run

```
sudo docker start <container> e.g. docker start dpctf
```

```
sudo docker stop <container> e.g. docker stop dpctf
```

## End of Part I: Installing Host Machine Required Software

---

---

## Part II: Build and Run the Test Suite

Now that the platform is configured with WSL2, Ubuntu, git, Docker, etc., you are ready to install and run the *WAVE Streaming Media Test Suite – Devices*.

**The following instructions include additional information that is not in the Deploy README but that may be useful, especially when installing in a Windows/WSL environment and for those without experience working in a Linux environment. See also the Deploy README at <https://github.com/cta-wave/dpctf-deploy?tab=readme-ov-file#phase-1-deployment-of-the-test-runner-one-time-action-to-be-performed-by-it-personnel>,**

### 1. Build the Test Runner

#### 1.1. First Time Building the Test Runner

1. Open an Ubuntu bash terminal

**NOTE: It's important to change the directory to the location where you will store the test suite repository. e.g.**

```
cd /mnt/c/next_level/next_level...
```

Typically for this install it would be:

```
cd /mnt/c/users/<username>/
```

/<user-name>/ is <your\_user\_name> on your PC

Note: Don't enter the <...> characters. Just your actual username.

Then type or copy

```
git clone https://github.com/cta-wave/dpctf-deploy
```

and hit return.

Then run



[Build the Test Runner \(mp4\)](#)

```
sudo ./build.sh
```

This will add a new folder “dpctf-deploy” to the directory you are in.

2. Change directory to the newly created folder <dpctf-deploy>:

```
cd /mnt/c/users/your_user-name/dpctf-deploy
```

#### 1.2. Agree to the EULA:



[Instructions on agreeing to the EULA \(PDF\):](#)

For the test runner to start you are required to agree to the EULA.

Set **AGREE\_EULA** to **yes**:

```
dpctf-deploy/docker-compose.yml
```

### 1.3. Update or rebuild the Test Runner see: [Update Test Runner](#)

## 2. Import Content



Import\_Content (mp4)

#### 1. run

```
sudo ./import.sh
```

This may take 5 or 10 minutes.

#### 2. When it completes run

```
sudo docker-compose up
```

**NOTE:** This may take 5 or 10 minutes or more and appear to stall at least once. Wait until it is complete. **NOTE:** It will not exit when done. To proceed, leave this terminal window open and start/ open a new terminal bash and use it to continue with section 3 “**Configure access to the test runner**” below followed by section 4 – Run a Test.

#### 2.1. Clean Up Docker in the event of a problem

**NOTE:** If `sudo docker-compose up` fails, try cleaning out Docker and rebuilding the Test Suite. Be cautious when running the following commands, as they will remove all containers and images regardless of their state or usage. Make sure you really want to delete everything before proceeding. Call your IT department if unsure how to proceed. Also see <https://www.digitalocean.com/community/tutorials/how-to-remove-docker-images-containers-and-volumes>

#### 1. To clean out Docker and rebuild the Test Suite Stop all running containers by running

```
docker stop $(docker ps -a -q)
```

Then run

Remove all containers

```
docker container prune
```

Remove all images

```
Docker image prune
```

Follow the instructions to rebuild the test runner. See [Update Test Runner](#).

If you have already run `./build-dof.sh` under section 5.1 *Build the Image* you will have to rerun that again as well. Make sure to run both in the directory

that contains the dpctf-deploy directory e.g. /mnt/c/users/<username>.

### 3. Configure access to the test runner:

Edit the “host\_override” in the config.json file which can be found in the “dpctf-docker” folder. Follow the instructions at: <https://github.com/cta-wave/dpctf-deploy?tab=readme-ov-file#configure-access-to-the-test-runner>

**NOTE:** For information on locating your <host IP> address and your <wsl\_IP> address see Note 6.2 “Configuring IP address”.

For tests run locally (e.g. your PC is both the host and the DUT (single device setup) you can use <localhost> instead of <host IP>. However, you will need to use your <host IP> or “yourhost.domain.tld” if you wish to run a test on a mobile device or a TV (two-device setup). Whichever you choose to use, make sure you enter it in the config.json file and use it to run the test per section 4 “Run a Test”.

**NOTE:** Some tests require certificates to support https. See more details at <https://github.com/cta-wave/dpctf-deploy?tab=readme-ov-file#configure-access-to-the-test-runner> You may need to contact your IT personnel to do this.

**NOTE:** You may need to open port 8000 to incoming connections. See [Note 7.1 Steps to Create an Inbound Firewall Rule](#) in the Additional Notes at the end of these instructions.

### 4. Run a Test (“Phase 2”)

For more details, see **Phase 2:** <https://github.com/cta-wave/dpctf-deploy?tab=readme-ov-file#phase-2-test-execution-and-recording-to-be-performed-by-tester>

**NOTE:** There have been reported cases where tests stall on e.g. frame 1 or other frames. Allowing stalled test to run to “Session completed” often clears this error for subsequent test runs. If the error persists, try running the tests on a different browser.

Ensure Docker is running the dpctf container by running

```
sudo docker start dpctf
```

#### 4.1. To run a test (Single Device (Host and DUT running on the same device) :



[Running a Test Single Device \(mp4\)](#)

1. Open a browser and go to [http://localhost:8000/\\_wave/index.html](http://localhost:8000/_wave/index.html) if you added “localhost” to the config.json file in section 3 “Configure Access to the Test Runner” above.

**or go to**

[http://<host\\_IP>:8000/\\_wave/index.html](http://<host_IP>:8000/_wave/index.html) if you added your host IP to the config.json file in section 3 “Configure Access to the Test Runner” above.

2. Press “Configure Session” and choose the test(s) you wish to run.
3. Take note of the Token. The entire Token is used in both the single-device and 2-device tests to identify the test run for later access. The first 8 Token characters are used to configure the host for a 2-device test.
4. Start recording on the recording device
5. Press "Start session" button to start the test(s). A new tab will open to the left of the configuration tab. Be sure to immediately switch to the new tab to view and record the session. (Do not move your cursor around the test screen).
6. Once the "Session completed" screen appears, stop the recording.
7. The results can be accessed from the Session Completed screen and saved in the dpctf-deploy “Results” folder.
8. Store the recording in the dpctf-deploy directory or create a new folder to save recordings. Be sure to include the full path to the folder in <mp4-filepath> when running the analysis. See “Analyze a Recording” below.
  9. Save the link to the testing session including the session Token for later reference.

**NOTE: If you left the terminal session where you ran “docker-compose up” running you will see the test running.**

#### 4.2. To run a test on a 2-device setup

(separate host and DUT e.g. a phone or Tablet and a TV),  
To watch a complete test run, see:



[Running a Test 2-Devices.mp4](#)

See [MOBILE\\_USAGE.md](#)

[https://github.com/cta-wave/dpctf-deploy/blob/master/MOBILE\\_USAGE.md](https://github.com/cta-wave/dpctf-deploy/blob/master/MOBILE_USAGE.md)

See also [Phase 2: Test execution and recording](#)

<https://github.com/cta-wave/dpctf-deploy/blob/master/README.md#phase-2-test-execution-and-recording-to-be-performed-by-tester>

1. Open a browser on the DUT and enter  
`http://<host ip>:8000/_wave/index.html`
2. Open a browser on the host or companion device open a and enter  
`http://<host ip>:8000/_wave/configuration.html`
  - i. or scan the QR code from the DUT browser
3. After entering the first 8-characters of the token from the DUT into the Host/companion browser, configure the test and start the recording, then start the test session. The video will run on the DUT. Record the video from start to end.

4. Store the recording you made in Phase 2 in the dpctf-deploy directory or create a new folder to save recordings. Be sure to include the full path to the folder in <mp4-filepath> when running the analysis. See “Analyze a Recording” below.

## 5. Analyze a Recording (“Phase 3”)

**NOTE:** For more details see the README at <https://github.com/cta-wave/dpctf-deploy?tab=readme-ov-file#phase-2-test-execution-and-recording-to-be-performed-by-tester>

### 5.1. Build the image

To build the image run the build script in the dpctf-deploy directory:

1. Run

```
sudo ./build-dof.sh
```



[Build the Observation Framework \(mp4\)](#)

### 5.2. Add Results

To allow the Observation Framework to add the results to the Test Runner's session set the correct domain name of the Test Runner in the config file see: <https://github.com/cta-wave/dpctf-deploy?tab=readme-ov-file#configure-the-observation-framework>

### 5.3. Configure the Observation Framework



[Configure the Observation Framework \(mp4\)](#)

1. To create the **observation-config.ini** file see **Note 6.3** Adding the observation-config.ini file.

Alternatively you can follow the Deploy instructions at: <https://github.com/cta-wave/dpctf-deploy?tab=readme-ov-file#configure-the-observation-framework>

2. To view the text you entered

Run

```
sudo cat observation-config.ini
```

### 5.4. Run the analysis



[Analyze a Recording \(mp4\)](#)

1. Run

```
sudo ./analyse-recording.sh <mp4-filepath>/mp4-filename <options>
```

where the <mp4-filepath> is the path to the recording followed by the file name of the recording file (be sure to include ".mp4" at the end of the file name).

e.g. /mnt/c/users/<your\_user\_name>/dpctf-deploy/<xyz1234.mp4> assuming the mp4 file is stored in the dpctf-deploy directory.

**NOTE:** For more information on available <options> see <https://github.com/cta-wave/device-observation-framework/wiki/Debugging-Observation-Failures>

**NOTE:** After running the analysis, the mp4 file will be renamed and stored. To rerun the analysis on the same file you will need to use that new name for the analysis.

## 5.5. Getting the analysis results

1. If configured correctly in the step Configure the Observation Framework, the results of the analysis are now available in the Test Runner's session:  
`http://<yourhost.domain.tld>:8000/_wave/results.html?token=SESSIONTOKEN`
2. The Observation results are also located in the `dpctf-deploy/observation-results` directory.

## 5.6. Debugging

If the observation framework reports errors and/or that the device has failed, some information on analysis and debugging can be found at <https://github.com/cta-wave/device-observation-framework/wiki/Debugging-Observation-Failures>

# 6. Additional Notes:

## 6.1. Steps to Create an Inbound Firewall Rule

1. Open Windows Defender Firewall with Advanced Security:
2. Press Windows Key + R, type `wf.msc`, and press Enter. This will open the Windows Defender Firewall with Advanced Security.
3. Create a New Inbound Rule:
  - In the left-hand pane, click on "Inbound Rules".
  - In the right-hand pane, click on "New Rule..." to start the New Inbound Rule Wizard.
4. Select Rule Type:
  - Select "Port" and click "Next".
5. Specify the Ports:
  - Choose "TCP".
  - In the "Specific local ports" field, enter 8000 and click "Next".
6. Action:
  - Select "Allow the connection" and click "Next".
7. Profile:

- Check all profiles (Domain, Private, and Public) to apply the rule in all network conditions, then click "Next".
8. Name the Rule:
    - Give the rule a meaningful name, such as "Allow Port 8000", and click "Finish".

## 6.2. Configuring IP address:

1. To find your host IP address, open a Windows PowerShell as Administrator. Then type `ipconfig` and copy the IPV4 address of your windows host e.g. 10.x.x.xxx (let's call this `<host_ip>`). This is the IP you should use in "host\_override": "`<host_ip>`" (see [config.json](#)).

**NOTE:** Which Host IP address you use depends on whether you are using Ethernet or WiFi as the network for your host.

In Windows PowerShell run:

`wsl hostname -I` and copy the (first) IP address e.g. 172.24.202.133 (lets call this `<wsl_ip>`)

To find your Host and WSL IP addresses see:  [IP Config and wsl IP address screenshot \(PDF\)](#)

2. Using the above WSL IP addresses, in Windows PowerShell (**must be run as Admin by Admin User!**) run:

```
netsh.exe interface portproxy add v4tov4 listenport=8000
listenaddress=0.0.0.0 connectport=8000 connectaddress=<wsl_ip>
```

```
netsh.exe interface portproxy add v4tov4 listenport=8443
listenaddress=0.0.0.0 connectport=8443 connectaddress=<wsl_ip>
```

NOTE: Make sure you use the correct IP addresses `<host_ip>` & `<wsl_ip>`.

For a screenshot of the commands see:  [Configure access to the Test Runner WSL Proxy Screenshot \(PDF\)](#)

## 6.3. Adding the observation-config.ini file

**For more information see** <https://www.wikihow.com/Create-a-File-in-a-Directory-in-Linux>

1. To configure the Observation Framework create the **observation-config.ini** file in the dpctf-deploy folder using a Linux bash terminal follow these instructions:
  - cd to the dpctf-deploy directory
  - Run

```
cat >observation-config.ini
```

`<press enter>` It will create and open the observation-config.ini file. It will show a blank line since you have not entered any text yet. Next:

- Copy and paste the observation-config.ini text from <https://github.com/cta-wave/device-observation-framework/blob/main/config.ini> into the blank line.  
<press Ctrl+d to save>  
<press ctrl X>
2. To view the text you entered Run  
`cat observation-config.ini`  
or open the file using notepad.

## 7. Helpful Links

### Test Suite Installation README

<https://github.com/cta-wave/dpctf-deploy?tab=readme-ov-file#>

**Docker and WSL2 without Docker Desktop:** <https://dev.to/rombru/docker-and-wsl2-without-docker-desktop-3pg3>

### Check WSL version

<https://superuser.com/questions/1719857/how-to-find-out-wsl2-version>

### Upgrade WSL to WSL2

<https://pureinfotech.com/upgrade-wsl2-wsl1-windows-10/>

**Uninstall WSL and Ubuntu**<https://pureinfotech.com/uninstall-wsl-windows-11> Windows 11

<https://pureinfotech.com/uninstall-wsl2-windows-10> Windows 10

**How to Install Ubuntu in WSL2 in Just 3 Steps:** <https://linuxsimply.com/linux-basics/os-installation/wsl/ubuntu-on-wsl2/>

### Update Ubuntu

<https://ubuntu.com/tutorials/upgrading-ubuntu-desktop#1-before-you-start>

### Git First Time Setup

<https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>

### Uninstall Docker

<https://docs.docker.com/engine/install/ubuntu/#uninstall-old-versions>

<https://docs.docker.com/engine/install/ubuntu/#uninstall-docker-engine>

### Install Docker in Ubuntu

<https://docs.docker.com/engine/install/ubuntu/>

**Running Docker on WSL2 without Docker Desktop (the right way)**

<https://dev.to/felipecrs/simply-run-docker-on-wsl2-3o8>

**Post install: managing docker and setting up permissions, etc.**

<https://docs.docker.com/engine/install/linux-postinstall/>

**Add a file/text to a Linux directory (e.g. observation-config.ini)Creating a WSL virtual environment within Linux (not needed for the Test Suite).**

<https://www.wikihow.com/Create-a-File-in-a-Directory-in-Linux>