
vpp-firstcut Documentation

Release 0.1

John DeNisco

May 29, 2018

Contents

1	Concepts	3
2	Setup	5
3	Tasks	7
3.1	Writing and pushing VPP Documentation	7
4	Guides	11
5	Reference	13

FD.io (Fast data - Input/Output) is a collection of several projects and libraries to amplify the transformation to support flexible, programmable and composable services on a generic hardware platform. FD.io offers the Software Defined Infrastructure developer community a landing site with multiple projects fostering innovations in software-based packet processing towards the creation of high-throughput, low-latency and resource-efficient IO services suitable to many architectures (x86, ARM, and PowerPC) and deployment environments (bare metal, VM, container).

A key component is the Vector Packet Processing (VPP) library donated by Cisco.

CHAPTER 1

Concepts

Note: To Do

CHAPTER 2

Setup

Note: To Do

3.1 Writing and pushing VPP Documentation

3.1.1 Getting and Building the VPP Documentation

Overview

This repository contains the sources for much of the VPP documentation. These instructions show how most of the VPP documentation sources are obtained and built.

Build and View Instructions

I build and load the documents using a mac, but these instructions should be portable to any platform. I used the Python virtual environment.

For more information on how to use the Python virtual environment check out [Installing packages using pip and virtualenv](#).

1. Get the repository

```
git clone https://github.com/fdioDocs/vpp-docs
cd vpp-docs
```

2. Install the virtual environment

```
python -m pip install --user virtualenv
python -m virtualenv env
source env/bin/activate
pip install -r etc/requirements.txt
```

Note: To exit from the virtual environment execute:

```
deactivate
```

3. Build the html files

```
cd docs  
make html
```

4. View the results.

To view the results start a browser and open the file:

```
<THE CLONED DIRECTORY>/docs/_build/html/index.html
```

3.1.2 Pushing a patch to the VPP Documentation

Pushing a Patch

I build and load the documents using a mac, but these instructions should be portable to any platform. I used the Python virtual environment.

1. Review the changes

```
git status
```

2. Specify which files that will be pushed

```
git add <filename>
```

3. Commit the changes locally

```
git commit -s
```

4. Submit the changes for review

```
git review
```

Reviewing a Patch

1. Getting the patch for review

```
git review -d <review number>
```

1. Look at the changes

```
git status
```

2. Edit the changes you would like to add

3. Specify which files you changed

```
git add <filename>
```

4. Commit the changes locally

```
git commit --amend -s
```

5. Submit the changes for review

```
git review
```

Getting the Latest Sources

```
git reset --hard origin/master  
git checkout master
```


CHAPTER 4

Guides

Note: To Do

CHAPTER 5

Reference

Note: To Do
