

# Docs on SwarmLab.io !

## Πίνακας περιεχομένων

1. Prepare your development and test environment.....	2
1.1. <a href="#">Install docker</a> .....	2
1.1.1. <a href="#">Get swarmlab.io certificates</a> .....	2
1.2. <a href="#">Install dual</a> .....	2
2. LINUX.....	2
2.1. <a href="#">An Introduction to Linux</a> .....	2
2.2. <a href="#">The proc filesystem</a> .....	2
2.3. <a href="#">VIM - A simple workflow example with the vim editor</a> .....	3
2.4. <a href="#">Bash - Associative array examples</a> .....	3
2.5. <a href="#">Bash - Heredocs, Positional Parameters</a> .....	3
3. Internet of things (IoT).....	3
3.1. <a href="#">Intro IoT</a> .....	3
3.2. <a href="#">IoT swarm implementation</a> .....	3
3.3. <a href="#">Create a mote with Raspberry Pi/Arduino</a> .....	3
3.4. <a href="#">Sensor node/mote</a> .....	3
4. Security.....	4
4.1. <a href="#">Create Swarmlab cluster</a> .....	4
4.2. <a href="#">Scan and network statistics</a> .....	4
4.3. <a href="#">Network analysis</a> .....	4
4.4. <a href="#">Network Filter</a> .....	5
4.4.1. <a href="#">Traversing of tables and chains</a> .....	5
4.5. <a href="#">Iptables with shorewall! - Three-Interface Firewall</a> .....	5
4.6. <a href="#">virtual private network (VPN)</a> .....	5
4.7. <a href="#">SSH Tunneling</a> .....	5
5. Cloud.....	5
5.1. <a href="#">Cloud computing</a> .....	6
5.2. <a href="#">Swarm architecture</a> .....	6
5.3. <a href="#">Swarm architecture example</a> .....	6
5.4. <a href="#">Swarm Storage - INTRO</a> .....	6
5.5. <a href="#">Swarm Storage - CONFIG</a> .....	6
5.6. <a href="#">Swarm Storage - INSTALL</a> .....	6
6. <a href="#">Frequently Asked Questions</a> .....	6
7. <a href="#">How do I ask a good question?</a> .....	6

## Lab docs

**Internet of Things.** Is as a networked interconnection of devices in everyday use that are often equipped with ubiquitous mechanism.

**Security.** Is a set of mechanisms to protect sensitive data from vulnerable attacks and to guarantee confidentiality, integrity and authenticity of data.

# 1. Prepare your development and test environment

## 1.1. Install docker



**Docker** is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers.

Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels

### 1.1.1. Get [swarmlab.io](https://swarmlab.io) certificates

## 1.2. Install dual

# 2. LINUX

## 2.1. An Introduction to Linux



Absolute basics

## 2.2. The `proc` filesystem

## 2.3. VIM - A simple workflow example with the vim editor

## 2.4. Bash - Associative array examples

## 2.5. Bash - Heredocs, Positional Parameters

# 3. Internet of things (IoT)

## 3.1. Intro IoT



How It Works, Apps, Swarm: The Five Principles of Swarm Intelligence

## 3.2. IoT swarm implementation



A swarm implementation that will allow communication between all of the members/nodes.

- [source code](#)
- [info in adoc](#)
- [info in pdf](#)

## 3.3. Create a mote with Raspberry Pi/Arduino



The Raspberry Pi is a low-cost credit-card sized single-board computer. The Raspberry Pi was created in the UK by the Raspberry Pi Foundation. The Raspberry Pi Foundation's goal is to "advance the education of adults and children, particularly in the field of computers, computer science and related subjects."

## 3.4. Sensor node/mote



A sensor node, also known as a mote, is a node in a sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network.

## 4. Security

**Security** is defined as a set of mechanisms to protect sensitive data from vulnerable attacks and to guarantee confidentiality, integrity and authenticity of data.

**Network security**, in a cloud environment (**IaaS, PaaS, and SaaS**) OR **Cloud of Things** consists of the security of the underlying **physical environment** and the **logical security** controls that are inherent in the service or available to be consumed as a service.

- Physical environment security ensures access to the cloud service is adequately distributed, monitored, and protected by underlying physical resources.
- Logical network security controls consists of link, protocol, and application layer services.

In a **cloud environment**, a major part of network security is likely to be provided by virtual security devices and services, alongside traditional physical network devices.

Typically, the inspection and control of network traffic do not pass through physical interfaces where classical control devices can analyze or block them.

This is the reason why effective controls require the integration with the software layer - *network security architecture, security gateways (firewalls, WAF, SOA/API), Security Products (IDS/IPS, Sub Tier Firewall, Security Monitoring and Reporting, Denial of Service (DoS) protection/mitigation, and secure “base services” like DNSSEC and NTP.*

### 4.1. Create Swarmlab cluster



This tutorial describes how to create and set up swarmlab clusters

### 4.2. Scan and network statistics



This tutorial demonstrates some common **nmap** port scanning scenarios and explains the output.

### 4.3. Network analysis



**tcpdump** is a common packet analyzer that runs under the command line. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached. Distributed under the BSD license, tcpdump is free software.

## 4.4. Network Filter



Netfilter offers various functions and operations for **packet filtering**, **network address translation**, and **port translation**, which provide the functionality required for **directing packets** through a network and **prohibiting packets** from reaching sensitive locations within a network.

**iptables** is a command line utility for configuring Linux kernel firewall implemented within the Netfilter project. The term "iptables" is also commonly used to refer to this kernel-level firewall. It can be configured directly with iptables, or by using one of the many

### 4.4.1. Traversing of tables and chains



When a packet first enters the firewall, it hits the hardware and then gets passed on to the proper device driver in the kernel.

Then the packet starts to **go through a series of steps in the kernel**, before it is either **sent to the correct application** (locally), or **forwarded to another host** - or whatever happens to i

## 4.5. Iptables with shorewall! - Three-Interface Firewall



**Shorewall** is an open source firewall tool for Linux that builds **upon the Netfilter (iptables/ipchains)** system built into the Linux kernel, making it easier to manage more **complex configuration schemes** by providing a higher level of abstraction for describing rules using text files.

## 4.6. virtual private network (VPN)



A **virtual private network (VPN)** extends a private network across a public network, and enables users to send and receive data across shared or public networks as if their computing devices were **directly connected to the private network**.

## 4.7. SSH Tunneling



**SSH Tunneling**, is the ability to use ssh to **create a bi-directional encrypted network connection** between machines over which data can be exchanged, typically TCP/IP.

# 5. Cloud

**5.1. Cloud computing**

**5.2. Swarm architecture**

**5.3. Swarm architecture example**

**5.4. Swarm Storage - INTRO**

**5.5. Swarm Storage - CONFIG**

**5.6. Swarm Storage - INSTALL**

**6. Frequently Asked Questions**

**7. How do I ask a good question?**

- Introduce the problem before you post any code
- Help others reproduce the problem

In the body of your question, start by expanding on the summary you put in the title.

Explain how you encountered the problem you're trying to solve, and any difficulties that have prevented you from solving it yourself.

The first paragraph in your question is the second thing most readers will see, so make it as engaging and informative as possible.



Not all questions benefit from including code. But if your problem is with code you've written, you should include some. But don't just copy in your entire program! Not only is this likely to get you in trouble if you're posting your employer's code, it likely includes a lot of irrelevant details that readers will need to ignore when trying to reproduce the problem. Here are some guidelines:

- Include just enough code to allow others to reproduce the problem. For help with this, read [How to create a Minimal, Complete, and Verifiable example](#).
- If it is possible to create a live example of the problem that you can link to (for example, on <http://sqlfiddle.com/> or <http://jsbin.com/>) then do so - but also copy the code into the question itself. Not everyone can access external sites, and the links may break over time. Use Stack Snippets to make a live demo of inline JavaScript / HTML / CSS.
- **DO NOT post images of code**, data, error messages, etc. - copy or type the text into the question. Please reserve the use of images for diagrams or demonstrating rendering bugs, things that are impossible to describe accurately via text. For more information please see the Meta FAQ entry [Why not upload images of code/errors when asking a question?](#)