

# Cahier des charges

KostiTeam

October 23, 2016

## Contents

<b>1</b>	<b>Problématique</b>	<b>3</b>
<b>2</b>	<b>Solution</b>	<b>3</b>
<b>3</b>	<b>Pourquoi LXC ?</b>	<b>3</b>
<b>4</b>	<b>Architecture du programme</b>	<b>4</b>
4.1	Vues . . . . .	4
4.2	Diagrammes . . . . .	4
<b>5</b>	<b>Etapas prévues</b>	<b>4</b>
<b>6</b>	<b>Notice d'utilisation</b>	<b>4</b>
<b>7</b>	<b>Rapport de bug</b>	<b>4</b>

## 1 Problématique

La virtualisation peut être utile dans diverses situations. En effet, virtualiser un environnement/un objet permet de trouver ses caractéristiques optimales à moindre coût. De plus, cela permet aussi d'acquérir des compétences sans pour autant disposer de matériel. Dans le cadre d'études informatiques, il peut être pratique de simuler des réseaux dans lesquels des machines dialoguent. Marionnet est un exemple de logiciel de simulation de réseaux. Cependant, après l'avoir utilisé un tant soit peu, il s'avère que plusieurs défauts gênent son utilisation :

- crashes assez fréquents;
- impossible de changer les configurations des machines en marche;
- processus d'arrêt des machines trop fastidieux;
- interface peu ergonomique;
- logiciel trop gourmand en ressources.

Face à ces problèmes, nous avons donc décidé de créer un nouveau simulateur de réseaux.

## 2 Solution

Pour pallier les problèmes exposés précédemment, il nous semble pertinent de ne pas virtualiser les machines utilisées (VM) mais uniquement les environnements (VE). En effet, virtualiser une machine revient à virtualiser un environnement (software), mais aussi le matériel physique (hardware tel que la RAM, processeur etc.).

La virtualisation d'environnement répond plus à nos problèmes de performances : virtualiser un environnement en utilisant le même matériel physique permettrait un gain notable de performances, notamment lorsque le nombre de machines est important.

Pour virtualiser les environnements, nous avons choisi d'utiliser la technologie des "container" de LXC (LinuX Chroot).

## 3 Pourquoi LXC ?

LXC est l'ensemble bien connu d'outils, modules, bibliothèque et connecteurs pour différents langages, et permet de créer des containers Linux. LXC est bas niveau, très flexible et couvre à peu près toutes les technologies de confinement supportées par le noyau.

LXC possède sa propre API, notamment en C/C++, ce qui nous permet de l'intégrer à notre projet efficacement. LXC est plus léger que des technologies comme Docker (qui découle d'ailleurs de LXC)

## 4 Architecture du programme

### 4.1 Vues

### 4.2 Diagrammes

## 5 Etapes prévues

## 6 Notice d'utilisation

## 7 Rapport de bug