Francois D'Ugard



MISSION CRITICAL CLOUD

User Guide

# Copyright Notice

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation (THE SOFTWARE). THE SOFTWARE is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.  
  
THE SOFTWARE is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. In no event shall the authors be liable for any claim, damages, or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software. See the GNU General Public License for more details.  
  
You should have received a copy of the GNU General Public License along with this software. If not, see http://www.gnu.org/licenses.

Table of Contents

[Copyright Notice 1](#_Toc406070231)

[1. Introduction 5](#_Toc406070232)

[2. Hardware and Software Reuirements 5](#_Toc406070233)

[3. Installation and Setup 5](#_Toc406070234)

[4. Getting Started 5](#_Toc406070235)

[5. Quick reference 6](#_Toc406070236)

[6. Accessing online help 6](#_Toc406070237)

[7. References 6](#_Toc406070238)

# Introduction

This is the User Guide for the MC2 Service. This document provides basic installation, setup, and demonstration instructions to use the service. If you have any question please refer to the Accessing online help section of this document.

# Hardware and Software Requirements

Hardware

* 1 personal computer for development, testing, and demos. Minimum requirements: Dual-core CPU @ 1.2 GHz, 2GB RAM, 128GB HDD.
* 2+ servers where the cloud infrastructure will be set up. Minimum requirements: Dual-core CPU @ 1.4 GHz, 8GB RAM, 500 GB HDD, 2 x 1Gbps PCI LAN interfaces.

Software

* OpenStack cloud computing platform.
* IPOP Peer to peer VPN networking controller
* Python 2.7
* Ubuntu Server 12.04 or 14.04 LTS.

# Installation and Setup

The following instructions describe how to setup and install the XMPP service and

# XMPP Server Setup

Begin by executing ./mcvpn.py on the desired physical host server.

$ git clone <https://github.com/FIU-SCIS-Senior-Project-2014-Fall/Mission-Critical-Cloud.git>

$ cd Mission-Critical-Cloud/mcvpn/

$ ./mcvpn.py

This command will download install and start the ejabberd service and install the xmpp server.

Alternatively you can follow the directions to install XMPP Server.

https://github.com/ipop-project/documentation/wiki/Installing-XMPP-server

Run start.py to begin running XMPP server.

$ cd Mission-Critical-Cloud/mcvpn/

$ ./start.py

# Virtual Machine Controller Setup

Launch virtual machines as required within your Cloud Framework of choice i.e. OpenStack or Amazon AWS.

SSH into each virtual machine you have created.

Clone the Mission Critical Cloud files into the base directory of each virtual machine that will be a node member. Once the git clone command completes change directory into the ipop-tincan folder.

1. git clone https://github.com/FIU-SCIS-Senior-Project-2014-Fall/Mission-Critical-Cloud.git

2. cd ~/Mission-Critical-Cloud/ipop-14.07.0\_ubuntu12

3. Edit the config.json to specify the IP address of your XMPP server, give unique user names and IP addresses for each virtual machine you want in your cluster.

{

"xmpp\_username": "namenode1@ejabberd", # <--- Unique User name \*Choose one of the usernames that was setup for the XMPP server previously

"xmpp\_password": "password", # <--- Optional choose a good password \*It's easier to test if you leave them all the same.

"xmpp\_host": "131.94.128.21", # <--- IP Address of XMPP Server

"ip4": "172.31.x.y", # <--- Choose Unique IP Address

"ip4\_mask": 24,

"tincan\_logging": 0,

"controller\_logging": "DEBUG"

}

4. Repeat the previous steps on all Virtual Machines that require the MCC service.

Run ./start.sh to automatically run the MC^2 service. The basic installation comes preconfigured to function properly and allows of virtual machines to self-discover. You can change the default settings by editing the CONFIG.json file in the ipop directory.

./start.sh

or

./x.sh

Alternatively run:

./reload.sh

Or:

./reload\_debug.sh

To kill the service and pull the latest code from git hub and run the service silently or verbosely.

Stop/kill the service by executing.

./kill.sh

# Getting Started

To verify installation retrieve the IPV6 address of any of the virtual machine nodes that has the MC2 service running.

$ ./getstate.sh

Copy the ip address and use it to ping from another virtual machine in group.

$ ping [ipv ip]

# Accessing online help

Go to the projects main website at mcc-dev.cis.fiu.edu to get help. Alternatively you can access the GitHub page as well

# References

MC^2 Homepage: mcc-dev.cis.fiu.edu

Github: https://github.com/FIU-SCIS-Senior-Project-2014-Fall/Mission-Critical-Cloud.git