# **RunTime Server Terror Documentation**

Michael DeMeo | Todd Murphy | Het Patel | Panchajanya Vangapandu | Vishalkumar Patel (later addition)

### **Initial Installation**

* Download Ubuntu 18.04.03 LTS Desktop on virtual machine instance from<https://www.ubuntu.com/download/desktop>
* VM Requirements
  + At least 4GB of memory
  + Linux Ubuntu ISO mentioned above
  + 30GB of virtual hard disk space
* Open the terminal
* Update and upgrade packages:
  + *sudo apt-get update*
  + *sudo apt-get upgrade*
* Install git:
  + *sudo apt-get install git*
* Install vim as text editor:
  + *sudo apt-get install vim*
* *Install Aptitude for package management*
  + *sudo apt-get install aptitude*
* Install SSH:
  + *sudo apt-get install openssh-server*
  + *sudo apt-get install openssh-client*
* Clone RunTime Terror’s GitHub Repository:
  + *git clone*[*https://github.com/tmurphy605/IT490/*](https://github.com/tmurphy605/IT490/)

### 

### **RabbitMQ & PHP Installation**

* Install the RabbitMQ server
  + *sudo apt-get install rabbitmq-server*
* Install PHP
  + *Sudo apt-get install php*
  + *sudo apt-get install php-amqp*
* Access the RabbitMQ Management page
  + Open browser and type *localhost:15672* into the browser
* Login information to access the management page
  + *username: test*
  + *password: test*
* Create user “test” under the Admin section and give it a password as test
* Create a new vHost named it490 and give access to user test and guest
  + Also give users access to vHost /
* Create a new exchange named test on vHost it490 and give it a type of Topic
* Create a new queue named test on the vHost it490
* Bind the queue and exchange together
* Do keep in mind that the broker host needs to have the ip of 192.168.0.144

### **Backend**

* Clone RunTime Terror’s GitHub Repository:
  + *git clone*[*https://github.com/tmurphy605/IT490/*](https://github.com/tmurphy605/IT490/)
* Configure IP addresses of the virtual machines according to the hosts file:
  + [*https://github.com/tmurphy605/IT490/blob/master/dep/hostsFile*](https://github.com/tmurphy605/IT490/blob/master/dep/hostsFile)

### 

### **Database**

* Install mysql:
  + *sudo apt-get install mysql*
* Create a database user:
  + *GRANT ALL PRIVILEGES ON \*.\* TO <username>@<localhost> IDENTIFIED BY <password>;*
* Create a test-database:
  + *CREATE DATABASE <test db name>;*
* Enter the newly created test-database:
  + *USE <test db name>;*
* Create the following tables: ‘Friends’, ‘Login’, and ‘Userld’
  + *CREATE TABLE <table name> (<column1 name > <column1 type>, <column2 name > <column2 type>);*
* Friends fields:
  + Email: varchar(255)
  + Friend: varchar(255)
* Login fields:
  + Email: varchar(255)
  + Password: varchar(255)
    - This is hashed in the file handler\_registration with the md5 function
  + firstName: varchar(255)
  + lastName: varchar(255)
  + code: varchar(255)
* Userld fields:
  + Email: varchar(255)
  + Title: varchar(255)
  + Type: varchar(255)

### **Database replication process:**

* **Master configuration:**
  + Cd /etc/mysql.mysql.cnf
  + Change bind address to your ip
  + Uncomment the this specific line log\_bin = /var/log/mysql/mysql-bin.log
  + Uncomment the server id =1
  + Login to mysql, create a user and give replication privileges to that user
  + Run the following commands: show master status
  + Record file name and file position which is the critical part of replication
* **Slave configuration**
  + Cd /etc/mysql.mysql.cnf
  + Change bind address to your ip
  + Uncomment the this specific line log\_bin = /var/log/mysql/mysql-bin.log
  + Uncomment the server id =2
  + Login to mysql, create a user and give replication privileges to that user
  + Run the following commands: show master status
  + Record file name and file position which is the critical part of replication
  + Run the following commands: stop slave;
  + CHANGE MASTER TO MASTER\_HOST='IP of first VM’, MASTER\_USER='USER', MASTER\_PASSWORD='password', MASTER\_LOG\_FILE=’filename', MASTER\_LOG\_POS= file position;
  + Run the following commands: start slave;
* **Master configuration:**
  + Run the following commands: stop slave;
  + CHANGE MASTER TO MASTER\_HOST='IP of second VM’, MASTER\_USER='USER', MASTER\_PASSWORD='password', MASTER\_LOG\_FILE=’filename', MASTER\_LOG\_POS= file position;
  + Start slave;
  + Now you can create database in master device and you can see database in other device

### **SystemD**

* **What is systemD?**
  + SystemD is linux service which contains ‘.service’ file. Once we enable systemd service, we can start specific jobs and services when the VM boots up or restarts.
  + For our project we need to start the RabbitMQServer when the system boots up.
* **How to setup the systemD?**
  + Go to /etc/systemd/system
  + Create a file for startup with a ‘.service’ extension
  + File consists of 3 parts Unit, Service, Install
  + Unit part refers to any source that system knows to operate and manage. The resources are defined using configuration files called unit files
  + Service part basically acts like a symbolic link. It takes the link of the file and connects it to the systemd
  + Install section declares units for multi-user targets
  + Once the file has been created, do a system restart.
  + Check the status of the service by this command : systemctl status ‘filename.service’

### **Firewalls**

* Download and install Iptables:
  + *sudo apt-get install iptables-persistent*
* Allow local network traffic
  + *iptables -I INPUT -s 127.0.0.1 -j ACCEPT*
* Whitelist IP addresses of all machines related to the project
  + *iptables -I INPUT -s <YOUR IP ADDRESS> -j ACCEPT*
* Deny all other traffic
  + *iptables -P INPUT DROP*

### **API Call**

* API url
  + "http://www.omdbapi.com/?i=tt3896198&apikey=92e1a0bb&t="
* Information retrieved by the API
  + Movie title
  + Rating
  + Poster
  + Genres

### **Apache Server Installation (Help)**

* Updating local repository
  + sudo apt-get update
* Installing the apache server
  + sudo apt-get install apache2
* Check the status on the apache server
  + sudo systemctl status apache2
  + url http//<ip address>.

### **GitHub Link**

<https://github.com/tmurphy605/IT490>

### **Trello**

See Trello.json at <https://github.com/tmurphy605/IT490/blob/master/Trello.json>

### **Slack**

See IT490 Team Slack export Sep 23 2019 - Dec 19 2019.zip at <https://github.com/tmurphy605/IT490/blob/master/IT490%20Team%20Slack%20export%20Sep%2023%202019%20-%20Dec%2019%202019.zip>

## 