# The System

# Development Guide

Contents

[Getting Started 2](#_Toc39248055)

[Development Prerequisites 2](#_Toc39248056)

[Architecture 2](#_Toc39248057)

[Current Fonebook Environments 3](#_Toc39248058)

[Development 3](#_Toc39248059)

[Clone Source Code 4](#_Toc39248060)

[Client Side Development 5](#_Toc39248061)

[Server Side Development 5](#_Toc39248062)

[Local Database 6](#_Toc39248063)

[Push Code to Bitbucket 6](#_Toc39248064)

[New Instance Setup 7](#_Toc39248065)

[DNS Entry 7](#_Toc39248066)

[Add Subdomain 8](#_Toc39248067)

[Server Setup 8](#_Toc39248068)

[Server Connection 8](#_Toc39248069)

[Install Software 9](#_Toc39248070)

[Configure YUM Repositories 9](#_Toc39248071)

[Install Additional Software 10](#_Toc39248072)

[Edit Hostname 10](#_Toc39248073)

[Add Security/Firewall 10](#_Toc39248074)

[Add Users 11](#_Toc39248075)

[Performance Tuning 11](#_Toc39248076)

[Configure Nginx 12](#_Toc39248077)

[Create Certificates 14](#_Toc39248078)

[Enable MongoDB: 14](#_Toc39248079)

[Addition Security and Sudo Access 15](#_Toc39248080)

[Enable Server-Side Application on Boot 15](#_Toc39248081)

[Enable Certificate Renewal 15](#_Toc39248082)

[Configure Backups 16](#_Toc39248083)

[Application Deployment 16](#_Toc39248084)

[Client Side 16](#_Toc39248085)

[Server Side 16](#_Toc39248086)

[Database 16](#_Toc39248087)

# Getting Started

## Development Prerequisites

In order to do any development on Fonebook the following software should be installed on the computer where development will be done.

* Node.js (<https://nodejs.org/en/>) \*Always install the LTS version
* Git (<https://git-scm.com/>)
* Visual Studio Code (<https://code.visualstudio.com/>)
* Go (<https://golang.org/>)
* MongoDB Community Server (<https://www.mongodb.com/download-center/community>)
* MongoDB Compass (<https://www.mongodb.com/products/compass>)

After all the prerequisites have been successfully installed you need to create the required folders to use Go. To create these folders the steps below can be followed

1. Open **Run** from the windows menu
2. Enter **%userprofile%** and click OK
3. Enter the **go** folder
4. Create 3 folders
   1. bin
   2. pkg
   3. src
5. Open the **src** folder
6. Create a folder **main**

If you prefer doing it with command prompt

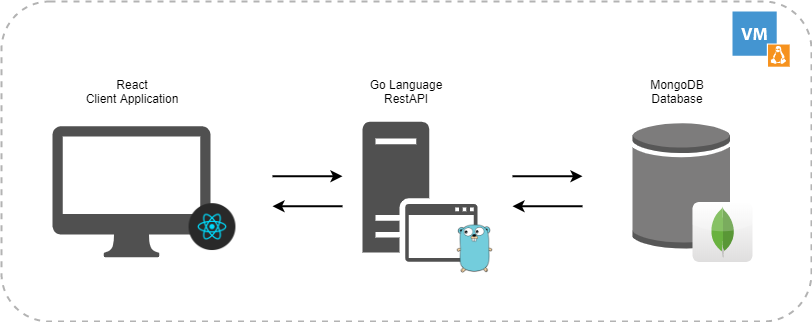
1. Open **Command Prompt**
2. Run the following commands
   1. mkdir %userprofile%\go\bin
   2. mkdir %userprofile%\go\pkg
   3. mkdir %userprofile%\go\src
   4. mkdir %userprofile%\go\src\main

## Architecture

Fonebook was developed using ReactJS for the client side application, Go Language as a RestAPI Server Application and MongoDB for the database. The client application, server side application and database is hosted on a Linux VM.

Each client using Fonebook has their own Linux VM with their own subdomain under the main domain *sabco.za.com*.

The source code is stored in a Bitbucket Git repository. This is important as the version control allows you to revert changes if needed.



## Current Fonebook Environments

There are currently 3 Linux VM servers running. The setups are identical with different domains and code bases.

|  |  |  |  |
| --- | --- | --- | --- |
| **Client** | **Domain** | **IP Address** | **Bitbucket Repo** |
| Adrian’s Call Centre | fonebook.sabco.za.com | 154.66.196.101 | Fonebook |
| Cell C | collections.sabco.za.com | 154.66.196.71 | FonebookCollections |
| EOH | eohcollections.sabco.za.com | 154.66.196.103 | FonebookEOHCollections |

Host: https://www.domains.co.za/

OS: CentOS7

SFTP: Yes (Port 22)

Front-End App Location: /usr/share/nginx/html/fonebook (Hosted with NGINX)

Back-End API Location: /opt/apps/fonebook

Username: appuser

Password: P@$Sbg0mb2dIgt3^lsD1

# Development

Once all the development prerequisites have been installed you can start making code changes. If you are not familiar with the technologies mentioned in the architecture I recommend familiarizing yourself with these before attempting to make code changes. Below are a few useful links to get you started

React

* <https://reactjs.org/tutorial/tutorial.html>
* <https://www.youtube.com/watch?v=dGcsHMXbSOA>

Go Lang

* <https://www.youtube.com/watch?v=ty49_v1tV44>

MongoDB:

* <https://www.youtube.com/watch?v=tXoTd7TH9K8>

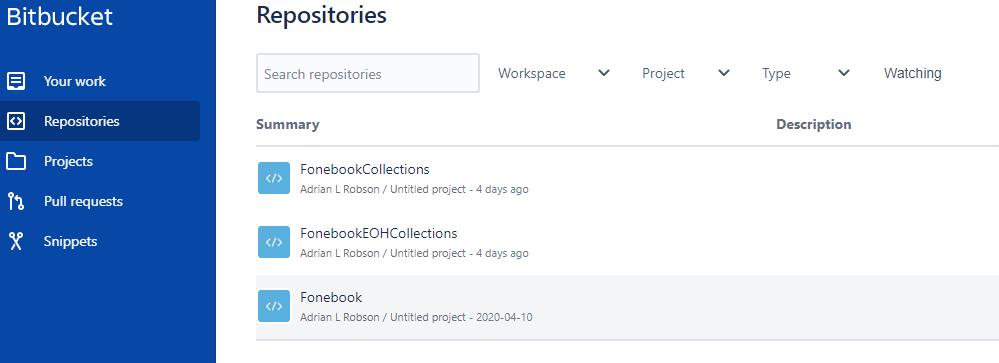
Git:

* <https://www.youtube.com/watch?v=SWYqp7iY_Tc>

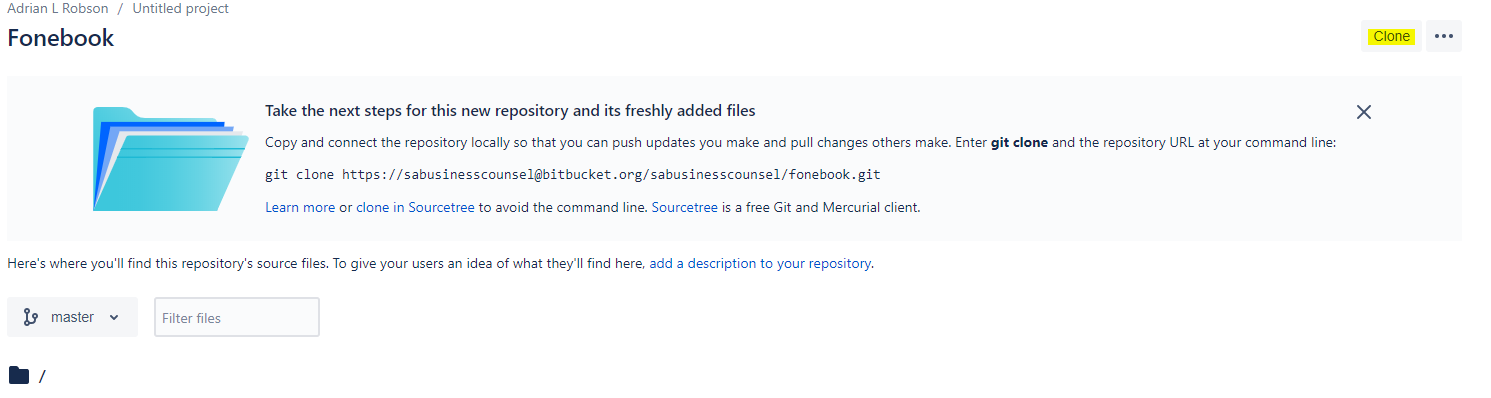
## Clone Source Code

To clone the source code to the development machine follow the next steps

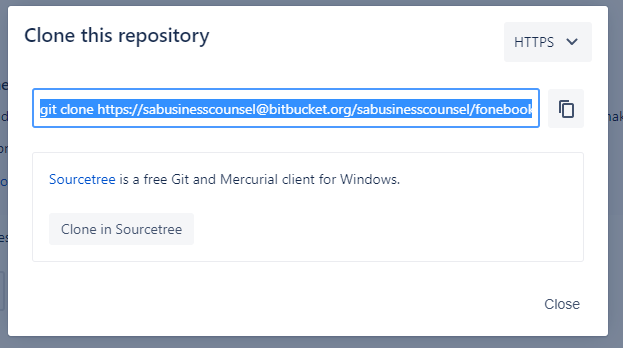
1. Create a folder **Source** on your C:\ drive
2. Login to <https://bitbucket.org/>
3. Go to **Repositories** and click on the repository that you want to clone



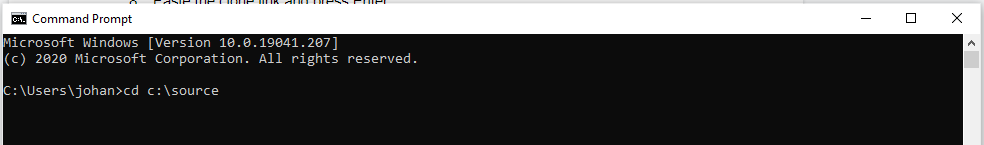
1. Click on **Clone**



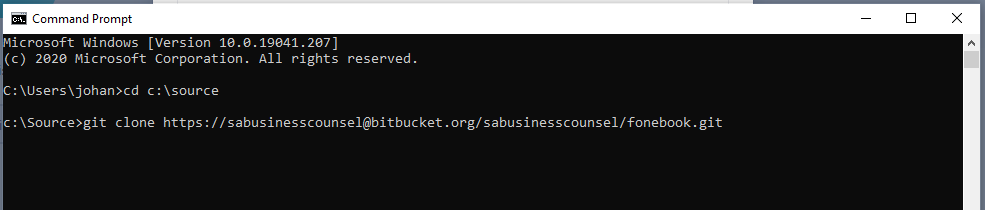
1. Copy the clone link



1. Open **Command Prompt** and navigate to C:\Source using command **cd c:\source**

****

1. Paste the clone link and press Enter



Once the clone is complete you have the code on your computer and can start development.

## Client Side Development

To start client side development follow the steps below

1. Open **Visual Studio Code**
2. Click on **File > Open Folder > Navigate to C:\Source\<project you want to work on>\client** and click on **Select Folder**
3. Click on **Terminal > New Terminal**
4. If it is the first time you are developing on the project you need to install packages. In the terminal execute the command **npm install**
5. Open the **appsettings.js** file in the **src** folder and be aware of the endpoint that the application is pointing to. If it points to a **sabco.za.com** url **it will affect customer data**. If it points to **localhost** ensure that you have the server running on you local environment.
6. To start the application in the development environment, execute the following command in the terminal **npm start**
7. If the application has started you can open your browser and navigate to **localhost:3000**
8. React allows hot reloading, this means that you can make changes while the application is running and it will automatically reload once you save the changes.

All styling is done using the MDBootstrap (<https://mdbootstrap.com/docs/react/>). This site can be referenced for components and styling. IMPORTANT: always make sure that you reference the React documentation on MDBootstrap.

When all development is done and code is working as expected the changes need to be saved to the Bucket code repository. See [Push Code to Bitbucket](#_6n5ze1y5ijvv)

## Server Side Development

To start client side development follow the steps below

1. Ensure that your local mongo database is running
   1. Open **Run**
   2. Enter **services.msc** and click OK
   3. Scroll to **MongoDB Server**
   4. Ensure that it is running
2. Open **Visual Studio Code**
3. Click on **File > Open Folder > Navigate to C:\Source\<project you want to work on>\server** and click on **Select Folder**
4. You can now make any changes to the code.
5. To run the application locally after making any changes click on **Terminal > New Terminal** and execute the command **run.bat**

NOTE: Go does not support hot reloading so the application has to be restarted after making any changes.

When all development is done and code is working as expected the changes need to be saved to the Bucket code repository. See [Push Code to Bitbucket](#_6n5ze1y5ijvv)

## Local Database

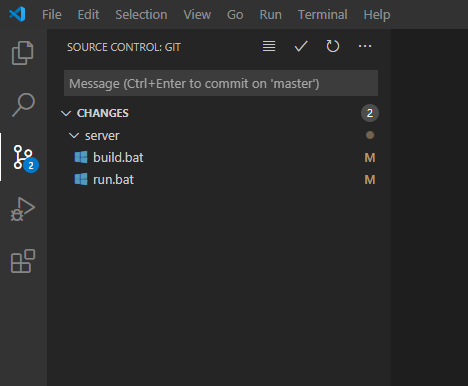
To setup your local database follow the steps below

1. Ensure that your local mongo database is running
   1. Open **Run**
   2. Enter **services.msc** and click OK
   3. Scroll to **MongoDB Server**
   4. Ensure that it is running
2. Open **MongoDB Compass**
3. Click on **Connect**. Do not change any connection information
4. Create a database with the name **fonebook**
5. Create the following collections within the database
   1. auth
   2. clients
   3. config
   4. stores
   5. tickets

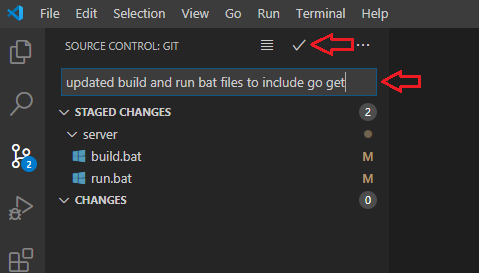
## Push Code to Bitbucket

To push the code to the Bitbucket repository follow the steps below.

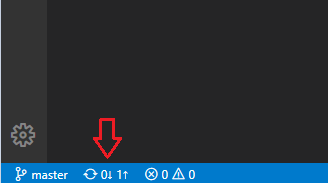
1. Have the project loaded as explained in [Client Side Development](#_3e9aqqrvlc1z) and [Server Side Development](#_22vex8874r0t).
2. Click on the source control tab to see all changes that have not been committed. Right click and **Stage** the files that you would like to commit.



1. Enter a commit message and click the commit check mark



1. Click on the sync button at the bottom to push the changes to Bitbucket



# New Instance Setup

In order to set up a new environment for a client you will need to buy a new cloud server with, at minimum, the specifications below.

Operating System: CentOS 7

Memory: 1 GB

Hard Drive: 20 GB SSD

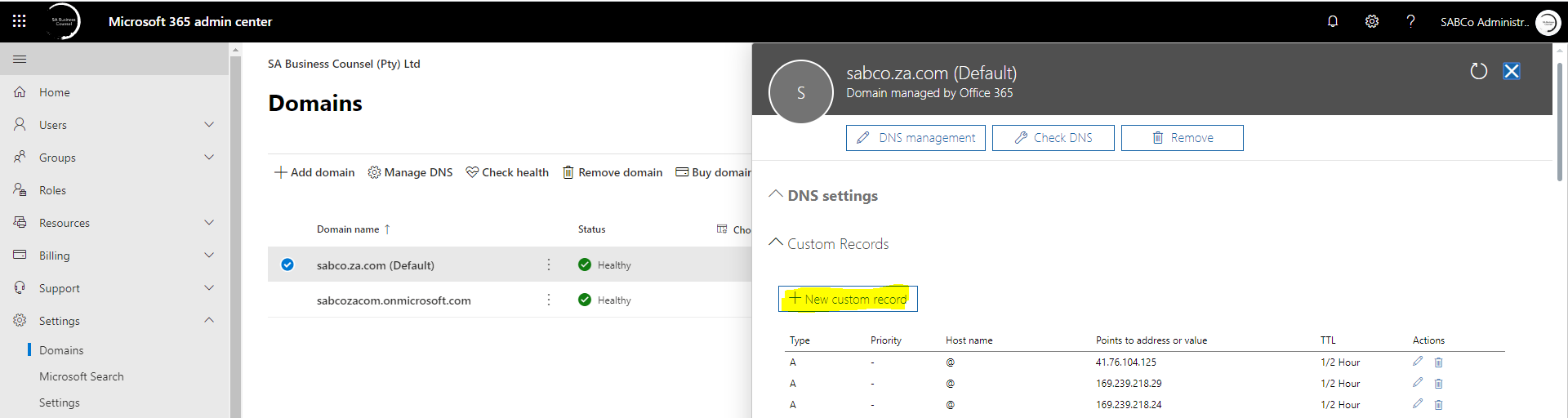
CPU: 1 Core Processor

Once the server is ready from the hosting company you should complete the setup as soon as possible to prevent it from being hacked and taken over. If it gets taken over by some hacker, you can simply reset the server then do the setup.

## DNS Entry

In order to assign a domain to the server you need to add a dns entry for the server. Since fonebook mainly uses subdomains to access different instances the DNS entry needs to be made under the main domain. To add the DNS entry, follow the steps below

1. Login into Office 365 admin portal ([https://portal.office.com/Adminportal/](https://portal.office.com/Adminportal/Home#/homepage))
2. Go to **Settings > Domains**
3. Click on **sabco.za.com** and click **New custom record**

****

1. Add the new DNS Entry
   1. DNS Type: A / Address
   2. Host Name: The new subdomain i.e. If the new subdomain is **newcollection**.sabco.za.com the hostname is **newcollection**
   3. Click **Save**

## Add Subdomain

The new server should be allocated it own domain. All fonebook instances runs on subdomains of sabco.za.com. To add a new subdomain, follow the steps below

1. Login to **domain.co.za**
2. Navigate to **Manage Account > Web Hosting**
3. Click on the **Manage** button on the sabco.za.com entry
4. Click on the **CPanel Login** button
5. Click on **Subdomains** and add the new subdomain i.e. **newcollection**.sabco.za.com
6. Manage the redirection for the new subdomain to redirect to https i.e **https://**newcollection.sabco.za.com

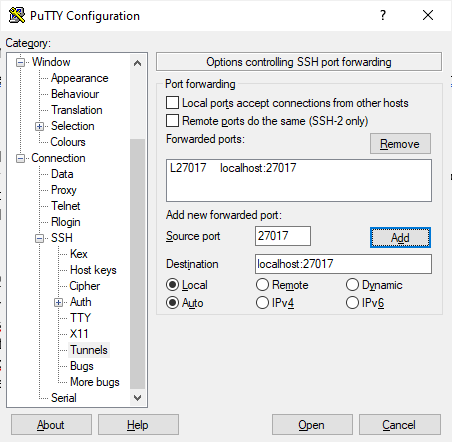
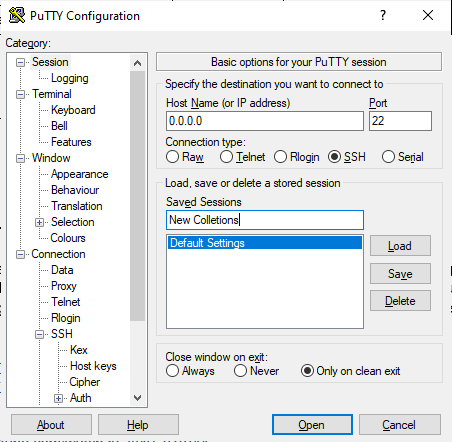
## Server Setup

### Server Connection

To connect to the server ensure that you have install PuTTY (<https://putty.org/>). If PuTTY is installed follow the steps below

1. Open PuTTY
2. In the category list navigate to **Connection > SSH > Tunnels**
3. Set source port to **27017** and destination to **localhost:27017** and click **Add** (This is to connect to the database later)
4. In the category list navigate to **Session**
5. Add the new server IP into the **Host Name** field
6. Add a new server name to the **Saved Sessions** field and click **Save**
7. Connect to the server by double clicking on the server name in the Saved Sessions list
8. Login with **root** user

NOTE: When logging into the server the password enter will not show any characters in the terminal. Just enter the password and press Enter

### Install Software

Once you have logged into the server you will be required to install some software. To install the software run the commands below

* yum -y install vim
* yum -y groupinstall "Development Tools"
* yum -y install iptables-services
* yum -y install epel-release
* yum -y install net-tools
* yum update
* reboot

### Configure YUM Repositories

#### Nginx

1. Run command **cd /etc/yum.repos.d/**
2. Run command **vim nginx.repo**
3. Edit the file
   1. Press the **i** button to enter insert mode
   2. Paste the following content in the file

*[nginx]*

*name=nginx repo*

*baseurl=https://nginx.org/packages/rhel/7/$basearch/*

*gpgcheck=0*

*enabled=1*

* 1. Press the **Esc** button to enter escape mode
  2. Type **:wq!** and press **Enter** to save and exit

### Install Additional Software

1. Run command **yum -y install yum-utils**
2. Run command **yum -y install nginx**
3. Run command **yum -y install certbot python2-certbot-nginx**
   * Go here: <https://certbot.eff.org/lets-encrypt/centos6-nginx>
   * Follow the instructions on the site. Have Certbot install and configure the files

### Edit Hostname

1. Run command **vim /etc/resolv.conf**
2. Edit the file
   1. Press the **i** button to enter insert mode
   2. Paste the following content in the file above the nameservers part

*domain thesystem.co.za*

* 1. Press the **Esc** button to enter escape mode
  2. Type :**wq!** and press **Enter** to save and exit

1. Run command **vim /etc/hostname**
2. Edit the file
   1. Press the **i** button to enter insert mode
   2. Enter the new subdomain name i.e. *newcollections*
   3. Press the **Esc** button to enter escape mode
   4. Type :**wq!** and press **Enter** to save and exit

### Add Security/Firewall

The steps below will add config for the firewall and only allow ports 22, 80, 443 accessible from the outside. Ports 12345 (Server-Side Application) and port 27017 (MongoDB) are only available on localhost.

1. Run command **vim /etc/sysconfig/iptables**
2. Edit the file
   1. Press the **i** button to enter insert mode
   2. Paste the following content in the file

*# sample configuration for iptables service*

*# you can edit this manually or use system-config-firewall*

*# please do not ask us to add additional ports/services to this default configuration*

*\*filter*

*:INPUT ACCEPT [0:0]*

*:FORWARD ACCEPT [0:0]*

*:OUTPUT ACCEPT [0:0]*

*-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT*

*-A INPUT -p icmp -j ACCEPT*

*-A INPUT -i lo -j ACCEPT*

*-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT*

*-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT*

*-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT*

*-A INPUT -p tcp -s localhost --dport 3306 -j ACCEPT*

*-A INPUT -p tcp -s localhost --dport 8081 -j ACCEPT*

*-A INPUT -p tcp --dport 8081 -j DROP*

*-A INPUT -j REJECT --reject-with icmp-host-prohibited*

*-A FORWARD -j REJECT --reject-with icmp-host-prohibited*

*COMMIT*

* 1. Press the **Esc** button to enter escape mode
  2. Type :**wq!** and press **Enter** to save and exit

1. Run command **vim /etc/sysconfig/ip6tables**
2. Edit the file
   1. Press the **i** button to enter insert mode
   2. Paste the following content in the file

*# sample configuration for ip6tables service*

*# you can edit this manually or use system-config-firewall*

*# please do not ask us to add additional ports/services to this default configuration*

*\*filter*

*:INPUT ACCEPT [0:0]*

*:FORWARD ACCEPT [0:0]*

*:OUTPUT ACCEPT [0:0]*

*-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT*

*-A INPUT -p ipv6-icmp -j ACCEPT*

*-A INPUT -i lo -j ACCEPT*

*-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT*

*-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT*

*-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT*

*-A INPUT -p tcp -s localhost --dport 3306 -j ACCEPT*

*-A INPUT -p tcp -s localhost --dport 8081 -j ACCEPT*

*-A INPUT -p tcp --dport 8081 -j DROP*

*-A INPUT -j REJECT --reject-with icmp6-adm-prohibited*

*-A FORWARD -j REJECT --reject-with icmp6-adm-prohibited*

*COMMIT*

* 1. Press the **Esc** button to enter escape mode
  2. Type **:wq!** and press **Enter** to save and exit

To enable the firewall to start at boot and start it now

1. Run command **systemctl enable iptables**
2. Run command **systemctl enable ip6tables**
3. Run command **systemctl restart iptables**
4. Run command **systemctl restart ip6tables**

### Add Users

#### Adrian – Sudo User

1. Run command **useradd -m -d /home/darryll -s /bin/bash adrian**
2. Run command **passwd adrian**
3. Enter a strong password (shared with Adrian)

#### Application User

1. Run command **useradd -m -d /home/appuser -s /bin/bash appuser**
2. Run command **passwd appuser**
3. Enter this password: **YSxws9G32gwsHE$J7p!wYWUGuJpXxwnf** (this is for conformity across all instances)

### Performance Tuning

1. Run command **vim /etc/sysctl.conf**
2. Edit the file
   1. Press the **i** button to enter insert mode
   2. Add *fs.file-max=200000* to the file
   3. Press the **Esc** button to enter escape mode
   4. Type **wq!** and press **Enter** to save and exit
3. Run command **vim /etc/security/limits.conf**
4. Edit the file
   1. Press the **i** button to enter insert mode
   2. Paste the below at the bottom of the file

*appuser soft nofile 65535*

*appuser hard nofile 68535*

*appuser soft nproc 65535*

*appuser hard nproc 68535*

*mongod soft nofile 65535*

*mongod hard nofile 68535*

*mongod soft nproc 65535*

*mongod hard nproc 68535*

*nginx soft nofile 65535*

*nginx hard nofile 68535*

*nginx soft nproc 65535*

*nginx hard nproc 68535*

* 1. Press the **Esc** button to enter escape mode
  2. Type :**wq!** and press **Enter** to save and exit

### Configure Nginx

1. Run command **vim /etc/nginx/nginx.conf**
2. Edit the file
   1. Press the **i** button to enter insert mode
   2. Replace the content of the file with the below

*user nginx;*

*worker\_processes 1;*

*error\_log /var/log/nginx/error.log warn;*

*pid /var/run/nginx.pid;*

*events {*

*worker\_connections 1024;*

*}*

*http {*

*include /etc/nginx/mime.types;*

*default\_type application/octet-stream;*

*log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '*

*'$status $body\_bytes\_sent "$http\_referer" '*

*'"$http\_user\_agent" "$http\_x\_forwarded\_for"';*

*access\_log /var/log/nginx/access.log main;*

*#large\_client\_header\_buffers 16 64k;*

*#client\_max\_body\_size 100M;*

*ssl\_protocols TLSv1.1 TLSv1.2;*

*ssl\_prefer\_server\_ciphers On;*

*ssl\_ciphers ECDH+AESGCM:ECDH+CHACHA20:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS;*

*server\_tokens off;*

*sendfile on;*

*#tcp\_nopush on;*

*keepalive\_timeout 65;*

*gzip on;*

*gzip\_disable "msie6";*

*gzip\_comp\_level 6;*

*gzip\_min\_length 1100;*

*gzip\_buffers 16 8k;*

*gzip\_proxied any;*

*gzip\_types*

*text/plain*

*text/css*

*text/js*

*text/xml*

*text/javascript*

*application/javascript*

*application/x-javascript*

*application/json*

*application/xml*

*application/rss+xml*

*image/svg+xml;*

*include /etc/nginx/conf.d/\*.conf;*

*}*

* 1. Press the **Esc** button to enter escape mode
  2. Type :**wq!** and press **Enter** to save and exit
  3. Run **nginx -t** to check config

1. Run command **cd /etc/nginx/conf.d**
2. Run command **vim default.conf**
3. Edit the file
   1. Press the **i** button to enter insert mode
   2. Paste the below in the file. Change the yellow parts to the new subdomain

*server {*

*listen 80 default\_server;*

*#listen [::]:80 default\_server ssl;*

*server\_name thesystem.co.za;*

*#ssl\_certificate /etc/letsencrypt/live/thesystem.co.za/fullchain.pem;*

*#ssl\_certificate\_key /etc/letsencrypt/live/thesystem.co.za/privkey.pem;*

*expires 0;*

*add\_header Cache-Control public;*

*add\_header Cache-Control no-store;*

*add\_header Cache-Control no-cache;*

*add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;*

*add\_header X-Frame-Options "SAMEORIGIN";*

*add\_header X-Content-Type-Options nosniff;*

*add\_header 'Referrer-Policy' 'origin';*

*add\_header X-XSS-Protection "1; mode=block";*

*add\_header Set-Cookie "HTTPOnly; HttpOnly; Secure";*

*location / {*

*root /usr/share/nginx/html/thesystem;*

*try\_files $uri /index.html;*

*}*

*location /integration/ {*

*charset\_types application/json;*

*charset UTF-8;*

*proxy\_pass* [*http://localhost:8081/*](http://localhost:8081/)*api/;*

*}*

*}*

*server {*

*listen 80;*

*#listen [::]:80;*

*server\_name uat.thesystem.co.za;*

*#ssl\_certificate /etc/letsencrypt/live/thesystem.co.za/fullchain.pem;*

*#ssl\_certificate\_key /etc/letsencrypt/live/thesystem.co.za/privkey.pem;*

*expires 0;*

*add\_header Cache-Control public;*

*add\_header Cache-Control no-store;*

*add\_header Cache-Control no-cache;*

*add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;*

*add\_header X-Frame-Options "SAMEORIGIN";*

*add\_header X-Content-Type-Options nosniff;*

*add\_header 'Referrer-Policy' 'origin';*

*add\_header X-XSS-Protection "1; mode=block";*

*add\_header Set-Cookie "HTTPOnly; HttpOnly; Secure";*

*location / {*

*root /usr/share/nginx/html/thesystem/uat;*

*try\_files $uri /index.html;*

*}*

*location /integration/ {*

*charset\_types application/json;*

*charset UTF-8;*

*proxy\_pass* [*http://localhost:8083/*](http://localhost:8083/)*api/;*

*}*

*}*

*server {*

*listen 80;*

*#listen [::]:80;*

*server\_name sit.thesystem.co.za;*

*#ssl\_certificate /etc/letsencrypt/live/thesystem.co.za/fullchain.pem;*

*#ssl\_certificate\_key /etc/letsencrypt/live/thesystem.co.za/privkey.pem;*

*expires 0;*

*add\_header Cache-Control public;*

*add\_header Cache-Control no-store;*

*add\_header Cache-Control no-cache;*

*add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;*

*add\_header X-Frame-Options "SAMEORIGIN";*

*add\_header X-Content-Type-Options nosniff;*

*add\_header 'Referrer-Policy' 'origin';*

*add\_header X-XSS-Protection "1; mode=block";*

*add\_header Set-Cookie "HTTPOnly; HttpOnly; Secure";*

*location / {*

*root /usr/share/nginx/html/thesystem/sit;*

*try\_files $uri /index.html;*

*}*

*location /integration/ {*

*charset\_types application/json;*

*charset UTF-8;*

*proxy\_pass* [*http://localhost:8082/*](http://localhost:8082/)*api/;*

*}*

*}*

*server {*

*listen 80 default\_server;*

*#listen [::]:443 default\_server ssl;*

*server\_name thesystem.co.za;*

*#ssl\_certificate /etc/letsencrypt/live/thesystem.co.za/fullchain.pem;*

*#ssl\_certificate\_key /etc/letsencrypt/live/thesystem.co.za/privkey.pem;*

*expires 0;*

*add\_header Cache-Control public;*

*add\_header Cache-Control no-store;*

*add\_header Cache-Control no-cache;*

*add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;*

*add\_header X-Frame-Options "SAMEORIGIN";*

*add\_header X-Content-Type-Options nosniff;*

*add\_header 'Referrer-Policy' 'origin';*

*add\_header X-XSS-Protection "1; mode=block";*

*add\_header Set-Cookie "HTTPOnly; HttpOnly; Secure";*

*location / {*

*root /usr/share/nginx/html/thesystem;*

*}*

*location /integration/ {*

*charset\_types application/json;*

*charset UTF-8;*

*proxy\_pass* [*http://localhost:8081/*](http://localhost:8081/)*;*

*proxy\_set\_header Proxy "";*

*}*

*}*

* 1. Press the **Esc** button to enter escape mode
  2. Type **wq!** and press **Enter** to save and exit
  3. Run **nginx -t** to check config

1. Run command **chkconfig nginx on**
2. Run command **service nginx restart**
3. Run command **reboot**

### Create Certificates

1. Run command **certbot certonly –nginx**
2. Run command **cd /etc/nginx/conf.d**
3. Run command **vim default.conf**
4. Edit the file
   1. Press the **i** button to enter insert mode
   2. Remove the green parts, and replace the blue part with *443 default\_server ssl;*

*server {*

*listen 443 default\_server;*

*listen [::]:443 default\_server ssl;*

*server\_name thesystem.co.za;*

*#ssl\_certificate /etc/letsencrypt/live/thesystem.co.za/fullchain.pem;*

*#ssl\_certificate\_key /etc/letsencrypt/live/thesystem.co.za/privkey.pem;*

*ssl on;*

*ssl\_certificate /etc/nginx/certs/cert\_chain.crt;*

*ssl\_certificate\_key /etc/nginx/certs/private.key;*

*expires 0;*

*add\_header Cache-Control public;*

*add\_header Cache-Control no-store;*

*add\_header Cache-Control no-cache;*

*add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;*

*add\_header X-Frame-Options "SAMEORIGIN";*

*add\_header X-Content-Type-Options nosniff;*

*add\_header 'Referrer-Policy' 'origin';*

*add\_header X-XSS-Protection "1; mode=block";*

*add\_header Set-Cookie "HTTPOnly; HttpOnly; Secure";*

*location / {*

*root /usr/share/nginx/html/thesystem;*

*try\_files $uri /index.html;*

*}*

*location /integration/ {*

*charset\_types application/json;*

*charset UTF-8;*

*proxy\_pass* [*http://localhost:8081/*](http://localhost:8081/)*api/;*

*}*

*}*

*server {*

*listen 443;*

*listen [::]:443;*

*server\_name uat.thesystem.co.za;*

*#ssl\_certificate /etc/letsencrypt/live/thesystem.co.za/fullchain.pem;*

*#ssl\_certificate\_key /etc/letsencrypt/live/thesystem.co.za/privkey.pem;*

*ssl on;*

*ssl\_certificate /etc/nginx/certs/uat/cert\_chain.crt;*

*ssl\_certificate\_key /etc/nginx/certs/uat/private.key;*

*expires 0;*

*add\_header Cache-Control public;*

*add\_header Cache-Control no-store;*

*add\_header Cache-Control no-cache;*

*add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;*

*add\_header X-Frame-Options "SAMEORIGIN";*

*add\_header X-Content-Type-Options nosniff;*

*add\_header 'Referrer-Policy' 'origin';*

*add\_header X-XSS-Protection "1; mode=block";*

*add\_header Set-Cookie "HTTPOnly; HttpOnly; Secure";*

*location / {*

*root /usr/share/nginx/html/thesystem/uat;*

*try\_files $uri /index.html;*

*}*

*location /integration/ {*

*charset\_types application/json;*

*charset UTF-8;*

*proxy\_pass* [*http://localhost:8083/*](http://localhost:8083/)*api/;*

*}*

*}*

*server {*

*listen 443;*

*listen [::]:443;*

*server\_name sit.thesystem.co.za;*

*#ssl\_certificate /etc/letsencrypt/live/thesystem.co.za/fullchain.pem;*

*#ssl\_certificate\_key /etc/letsencrypt/live/thesystem.co.za/privkey.pem;*

*ssl on;*

*ssl\_certificate /etc/nginx/certs/sit/cert\_chain.crt;*

*ssl\_certificate\_key /etc/nginx/certs/sit/private.key;*

*expires 0;*

*add\_header Cache-Control public;*

*add\_header Cache-Control no-store;*

*add\_header Cache-Control no-cache;*

*add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;*

*add\_header X-Frame-Options "SAMEORIGIN";*

*add\_header X-Content-Type-Options nosniff;*

*add\_header 'Referrer-Policy' 'origin';*

*add\_header X-XSS-Protection "1; mode=block";*

*add\_header Set-Cookie "HTTPOnly; HttpOnly; Secure";*

*location / {*

*root /usr/share/nginx/html/thesystem/sit;*

*try\_files $uri /index.html;*

*}*

*location /integration/ {*

*charset\_types application/json;*

*charset UTF-8;*

*proxy\_pass* [*http://localhost:8082/*](http://localhost:8082/)*api/;*

*}*

*}*

*server {*

*listen 443 default\_server ssl;*

*listen [::]:443 default\_server ssl;*

*server\_name thesystem.co.za;*

*#ssl\_certificate /etc/letsencrypt/live/thesystem.co.za/fullchain.pem;*

*#ssl\_certificate\_key /etc/letsencrypt/live/thesystem.co.za /privkey.pem;*

*expires 0;*

*add\_header Cache-Control public;*

*add\_header Cache-Control no-store;*

*add\_header Cache-Control no-cache;*

*# Added by Willie - HTTP Security Headers*

*add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;*

*add\_header X-Frame-Options "SAMEORIGIN";*

*add\_header X-Content-Type-Options nosniff;*

*add\_header 'Referrer-Policy' 'origin';*

*add\_header X-XSS-Protection "1; mode=block";*

*add\_header Set-Cookie "HTTPOnly; HttpOnly; Secure";*

*location / {*

*root /usr/share/nginx/html/thesystem;*

*try\_files $uri /index.html;*

*}*

*location /integration/ {*

*charset\_types application/json;*

*charset UTF-8;*

*proxy\_pass http://localhost:8081/;*

*}*

*}*

* 1. Press the **Esc** button to enter escape mode
  2. Type **wq!** and press **Enter** to save and exit

1. Run command **cd /etc/nginx/conf.d**
2. Run command **vim http.conf**
3. Edit the file
   1. Press the **i** button to enter insert mode
   2. Paste the below in the file. Change the yellow parts to the new subdomain

*server {*

*listen 80 default\_server;*

*listen [::]:80 default\_server;*

*server\_name thesystem.co.za;*

*return 301 https://$server\_name$request\_uri;*

*}*

*server {*

*listen 80;*

*listen [::]:80;*

*server\_name uat.thesystem.co.za;*

*return 301 https://$server\_name$request\_uri;*

*}*

*server {*

*listen 80;*

*listen [::]:80;*

*server\_name sit.thesystem.co.za;*

*return 301 https://$server\_name$request\_uri;*

*}*

* 1. Press the **Esc** button to enter escape mode
  2. Type :**wq!** and press **Enter** to save and exit

1. Run command **service nginx restart**

### Enable MySQL:

<https://www.linode.com/docs/databases/mysql/how-to-install-mysql-on-centos-7/>

create user 'thesyste\_prodsqluser'@'localhost' identified by '7W3#R$zu^$nbxF';

grant all on thesyste\_cws\_admin.\* to 'thesyste\_prodsqluser'@'localhost';

grant all on thesyste\_cws\_business.\* to 'thesyste\_prodsqluser'@'localhost';

flush privileges;

1. Run command **chkconfig mongod on**
2. Run command **service mongod restart**

### Addition Security and Sudo Access

1. Run command **vim /etc/sudoers**
2. Edit the file
   1. Press the **i** button to enter insert mode
   2. Add *adrian ALL=(ALL) ALL* below the line *root ALL=(ALL) ALL*
   3. Press the **Esc** button to enter escape mode
   4. Type **!wq** and press **Enter** to save and exit
3. Run command **vim /etc/ssh/sshd\_config**
4. Edit the file
   1. Press the **i** button to enter insert mode
   2. Find and replace *#PermitRootLogin yes* with *#PermitRootLogin no*
   3. Press the **Esc** button to enter escape mode
   4. Type **!wq** and press **Enter** to save and exit
5. Run command **service sshd restart**
6. Open a second terminal to the server via PuTTY and login with user *adrian*
7. Run command **sudo su -**
8. If step 7 is successful then continue

#### Create directories and change ownership

1. Run command **mkdir -p /opt/apps**
2. Run command **chown -R appuser.appuser /opt/apps**
3. Run command **mkdir -p /opt/scripts**
4. Run command **mkdir -p /opt/backups**
5. Run command **mkdir -p /usr/share/nginx/html/thesystem**
6. Run command **mkdir -p /usr/share/nginx/html/errors**
7. Run command **chown -R appuser.appuser /usr/share/nginx/html/thesystem**
8. Run command **chown -R appuser.appuser /usr/share/nginx/html/errors**
9. Run command **reboot**

### Enable Server-Side Application on Boot

1. Login to the server with user *adrian*
2. Run command **login to new server**
3. Run command **su - appuser**
4. Run command **crontab -e**
5. Edit the file
   1. Press the **i** button to enter insert mode
   2. Add @reboot sleep 60 && /opt/apps/fonebook/startstop.sh start to the file
   3. Press the **Esc** button to enter escape mode
   4. Type **!wq** and press **Enter** to save and exit

### Enable Certificate Renewal

1. Login to the server with user *Adrian*
2. echo "0 0,12 \* \* \* root python -c 'import random; import time; time.sleep(random.random() \* 3600)' && /usr/local/bin/certbot-auto renew -q" | sudo tee -a /etc/crontab > /dev/null
3. Run command **vim /etc/crontab**
4. Edit the file
   1. Press the **i** button to enter insert mode
   2. Add the below to the file

# Auto renew certificate

0 0,12 \* \* \* root certbot renew

* 1. Press the **Esc** button to enter escape mode
  2. Type **!wq** and press **Enter** to save and exit

### Configure Backups

1. Login to the server with user *adrian*
2. Run command **crontab -e**
3. Edit the file
   1. Press the **i** button to enter insert mode
   2. Add *55 23 \* \* \* /opt/scripts/dump\_db.sh* the below to the file
   3. Press the **Esc** button to enter escape mode
   4. Type **!wq** and press **Enter** to save and exit

This creates local backups only in **/opt/backups**.

These files should be shipped to a remote location

# Application Deployment

## Client Side

1. Open the project that you would like to deploy in VS Code as explained in the development section
2. In the terminal run command **npm run build**
3. Connect to the server that you would like to deploy to via SFTP (you can use FileZilla)
4. Copy the contents of **C:\Source\<selected application>\Client\Build** to **/usr/share/nginx/html/fonebook**

## Server Side

1. Open the project that you would like to deploy in VS Code as explained in the development section
2. In the terminal run command **build.bat**
3. Connect to the server that you would like to deploy to via SFTP (you can use FileZilla)
4. Copy the files below from **C:\Source\<selected application>\Server** to **/opt/apps/fonebook**
   1. fonebook
   2. startstop.sh
5. Connect to the server via PuTTY and run command **/opt/apps/fonebook/startstop.sh restart**

## Database

1. Ensure that your local mongo database is stopped
   1. Open **Run**
   2. Enter **services.msc** and click OK
   3. Scroll to **MongoDB Server**
   4. Ensure that it is stopped
2. Open **MongoDB Compass**
3. Click on **Connect**. Do not change any connection information
4. Create a database with the name **fonebook**
5. Create the following collections within the database
   1. auth
   2. clients
   3. config
   4. stores
   5. tickets

NOTE: You can connect to another server to make backups from that database to restore on the new instance.