HT1 Deployment Plan

Roles for deployment

* System administrator
  + Make the configuration changes to load all code/programs
* Training administrator
  + Administer the training materials to the users/clinic staff
* Network administrator
  + Make the firewall and server changes to comply with sponsor firewall rules, and allow an external connection to the web app.

Equipment needed for installation

* CentOS linux distribution
  + Sql server and apache web server
  + Storage to store needed application code and data

Equipment needed for operation

* CentOS linux distribution
  + Sql server and apache web server
  + At least 2GB of RAM
  + 30 GB of storage

Depending on the scale of use the RAM and storage should be changed to fit the needs in the future

Procedure for installing the system

Preconditions: Web server and Sql server are already installed

Part 1: Get frontend installed to new server

* Installing nodejs

curl -sL https://rpm.nodesource.com/setup\_10.x | sudo bash -

Sudo yum install node js

When prompted to import the repository GPG key, type y, and press Enter

Verify node.js is installed with the command

node --version

To install Node.js and npm using NVM on your CentOS system, follow these steps:

### **1. Install NVM (Node Version Manager)**

To download the nvm install script run the following command:

curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh | bash

### **2. Install Node.js using NVM**

Now that the nvm tool is installed we can install the latest available version of Node.js, by typing:

nvm install node

Verify the Node.js version, by typing:

node --version

* Inside /var/www/html

/etc/httpd/conf/httpd.conf

Add the following



Copy the contents of the folder “BUILD” to the /var/www/html folder

Part 2: Get backend installed to new server

1. Installing MySQL

wget <http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm>

sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm

yum update

sudo yum install mysql-server

sudo systemctl start mysqld

1. Dump and export current database

mysqldump -u root -p senior\_ht1 > d:\{path}\db\_script.sql

mysql -u root -p senior\_ht1\_backup < d:\db\senior\_ht1.sql

mysqldump -u root -p --databases senior\_ht1 > d:\db\db.sql

1. Dump current database

wget <http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm>

1. Dump current database

wget <http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm>

1. Create new database on new server

CREATE DATABASE `senior\_ht1`.

USE `senior\_ht1`;

1. Create new database on new server

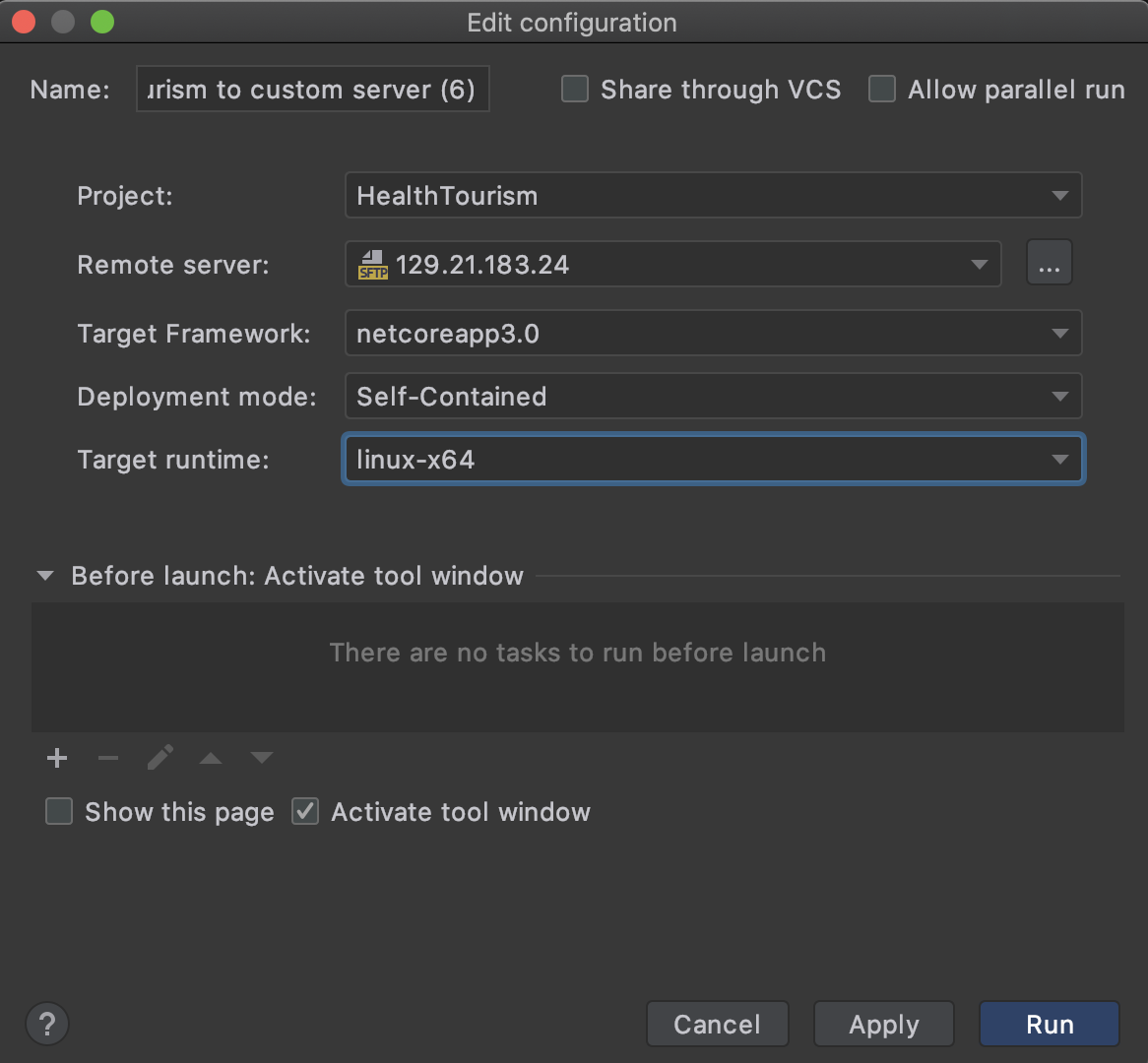
CREATE DATABASE `senior\_ht1`.

USE `senior\_ht1`;

1. Import sql file

mysql -u root -p senior\_ht1 < c:\tmp\db.sql

1. Publish the project into the folder on your local machine by using the “Publish” option in your IDE; or publish and upload the project directly to the server



1. Install the dotnet and supervisor tools on the server

sudo rpm --import <https://packages.microsoft.com/keys/microsoft.asc>

sudo sh -c 'echo -e "[packages-microsoft-com-prod]\nname=packages-microsoft-com-prod \nbaseurl= https://packages.microsoft.com/yumrepos/microsoft-rhel7.3-prod\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/keys/microsoft.asc" > /etc/yum.repos.d/dotnetdev.repo'

sudo yum update -y

sudo yum install libunwind libicu -y

sudo yum install dotnet-sdk-2.1.4 -y

sudo yum install supervisor -y

1. Edit supervisor config file

cd /etc/supervisord.d

sudo nano HealthTourism.conf

Populate the file:

[program:HealthTourism]

command=dotnet HealthTourism.dll

directory=/opt/ht-root

environment=ASPNETCORE\_\_ENVIRONMENT=Production

user=root

stopsignal=INT

autostart=true

autorestart=true

startsecs=1

stderr\_logfile=/var/log/HealthTourism.err.log

stdout\_logfile=/var/log/HealthTourism.out.log

Save and quit

1. Modify the default supervisord config file to include the config file you have created:

sudo cp /etc/supervisord.conf /etc/supervisord.conf.bak

sudo nano /etc/supervisord.conf

Find the last line:

files = supervisord.d/\*.ini

Replace it:

files = supervisord.d/\*.conf

Save and quit

1. Start Supervisor and set it to automatically start at system startup:

sudo systemctl start supervisord.service

sudo systemctl enable supervisord.service

1. Load the new Supervisor settings:

sudo supervisorctl reread

sudo supervisorctl update

1. Use the following command to show the app's status:

sudo supervisorctl status

Training Materials and Plans

Materials

* The following materials will be given to the deployment team
  + User Guide
  + Walk through video

Plan

* To ensure our Web App is utilized to its full potential, we will stress as much user training up-front in an effort to create a positive user experience for all. Upon deployment the training administrator will have the following tasks
  + Distribute the User Guide to the users
  + Play the Web App walk through video
  + Walk the users through using the web app

Deployment roll-out plan

Two-day implementation

A two-day implementation separated into two phases:

* Phase 1 - Installing the software
* Phase 2 - Training of the end users

|  |  |
| --- | --- |
| *Phase 1 - Day 1*   * Software installation / setup * Discuss project maintenance processes * Discuss and set up system admin and maintenance roles | *Phase 2 - Day 2*   * Overview of the software * Discuss workflow process, and demonstrate the functionalities * Hands-on practice |

Future releases plan

If necessary, there is an open possibility to continue working on the project for a certain fee which will have to be discuss and arranged with sponsors in a timely manner.

Support team

* Marko Cacic - Backend Developer
  + [mxc9700@rit.edu](mailto:mxc9700@rit.edu)
* Ulziiorshikh Batdorj - Frontend Developer
  + [uxb6317@rit.edu](mailto:uxb6317@rit.edu)
* Andrew Villella - System Administrator
  + [axv9484@rit.edu](mailto:axv9484@rit.edu)
* Ana Peric - Project Manager
  + [axp5428@rit.edu](mailto:axp5428@rit.edu)