

Written by Frederic Cui and Sim Winton

Last updated: 14/06/2021

Contents

[**What is Request Tracker? 3**](#_Toc105850026)

[**RT Ticketing System 3**](#_Toc105850027)

[**Access 3**](#_Toc105850028)

[**Hosting 3**](#_Toc105850029)

[**Connecting 3**](#_Toc105850030)

[**Pricing 4**](#_Toc105850031)

[**Connecting to RT-Test 4**](#_Toc105850032)

[**FQDN Implementation 6**](#_Toc105850033)

[**Installation Guide 7**](#_Toc105850034)

[**Dependences 7**](#_Toc105850035)

[**RT Installation 7**](#_Toc105850036)

[**Web Server Configuration 10**](#_Toc105850037)

[**CSRF Error 11**](#_Toc105850038)

[**Email Implementation 12**](#_Toc105850039)

[**Mail Account setup 12**](#_Toc105850040)

[**Application Password 13**](#_Toc105850041)

[**Postfix Installation 14**](#_Toc105850042)

[**RT User Creation 16**](#_Toc105850043)

[**Testing Outgoing Mail 17**](#_Toc105850044)

[**Fetching Incoming Mail 18**](#_Toc105850045)

[**Queue 20**](#_Toc105850046)

[**Customizing Templates 21**](#_Toc105850047)

[**Users 22**](#_Toc105850048)

[**SSL Certs 23**](#_Toc105850049)

[**Install SSL on RT 23**](#_Toc105850050)

# What is Request Tracker?

Request Tracker (RT) is an opensource ticketing system. The software allows users to create tickets which are then kept track in the system to help assist IT support teams. The software allows IT professional to keep track of incidents, to see what tasks have or have not been completed and to see who’s working on what so that incidents are efficiently solved.

# RT Ticketing System

## Access

The ticketing system is currently hosted on op-bit.nz DNS zone through Azure, giving the ticketing system the URL of rttest.ict.op.ac.nz. This is located within the OP-DNS resource group. It’s important that the IP address of 20.92.235.14 is continuously reserserved for the RT-Test virtual machine. If this IP address is changed, the value field on the op-bit.nz must be updated to reflect the new IP address, otherwise website will not be accessible via URL.

## Hosting

RT-Test is hosted on a Linux virtual machine called RT20.04, hosted within the student\_ops resource group, located in Australia East. This Virtual Machine is running Ubuntu 20.04 with the Standard B1ms offer, giving 2GB ram and 4GB of storage space.

## Connecting

There are two methods in which to access the RT20.04 virtual machine. These being the command line and Putty, however Putty is the recommended method.

1. Open Putty and enter 20.92.235.14 OR rttest.ict.op.ac.nz within the ‘Host Name’ field. Set the port to 22 and ensure the ‘Connection type’ is set to SSH.

Graphical user interface, text, application

Description automatically generated

Figure Use Putty Connect

1. On the ‘Category’ panel navigate through Connection -> SSH -> Auth. Under ‘Private key file for authentication:’ select browse and navigate to the folder where the private and public keys are stored on your system. These will be available for download on \_\_\_\_\_\_, alternatively if this is not available you’re able to generate new public and private keys and assign them via Azure under the ‘Reset Password’ tab for the virtual machine.

Graphical user interface, text, application

Description automatically generated

Figure Load public key

1. For admin access enter the Username RT-Admin and use the relevant password stored within the DevOps KeePass vault, ITPOperations.kdbx. Documentation for this vault can be found on the DevOps team group within the Files tab, under Documents > General > DevOps Documentation > Guides titled ‘Keepass credentials.docx’.

## Pricing

Pricing for the RT Ticketing system is calculated via each hour that the virtual machine is active for. Assuming that it’s running 24/7 this will cost $15.11 monthly, resulting in $181.33 annually. However, this can be cut down significantly by keeping the virtual machine off during the off-hours for school. As the school year encompasses 38 weeks in total, this allows the annual cost of the ticketing system to go down to $132.26 annually, assuming proper protocol is followed.

## Connecting to RT-Test

* On RT 20.04 the server requires an SSH key to connect. Reviewing the Docs provided me a way to connect via ssh, however the link to download the public key for the server is not present.
* Open **PuttyGen**, generate an RSA key. Copy and paste the public key under the reset password tab in the azure portal under the SSH-public key text area.



Figure Reset password

Graphical user interface, text, application

Description automatically generated

Figure Reset SSH public key

* Save the generated key as a private key, then load up Putty. Navigate to Connection -> SSH -> AUTH. Select browse and navigate to your key file, load it in and connect.

Graphical user interface, text, application

Description automatically generated

Figure Set public key on Putty

**Inbound port rules**

In NSG set inbound port rules on port 22, 80 and 8080, select IP Addresses as source enter IP Address 202.49.0.0/21. To allow all the Otago Polytechnic address in.

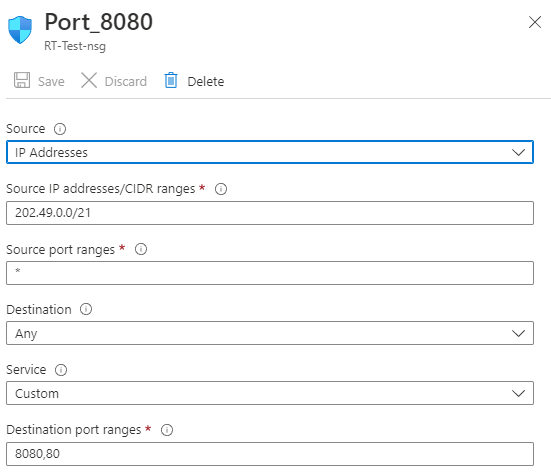
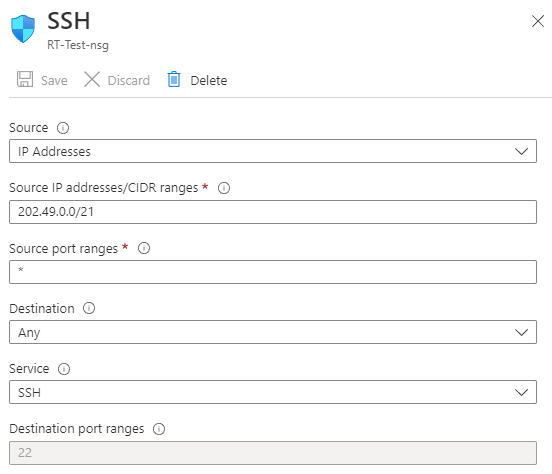


Figure 6 add Inbound rule on port 22 Figure 7 add Inbound rule on port 80 and 8080



Figure 8 Set NSG

## FQDN Implementation

Graphical user interface, application

Description automatically generatedSelect your VM. On the left panel, click on properties. find public IP/DNS then click on it.

Graphical user interface, application

Description automatically generated

Type in the FQDN you wish and click on “save”. The VM should now have a FQDN associated to it.

Graphical user interface, text, application, email

Description automatically generated

Figure Set DNS for RT

# Installation Guide

This installation guide is written for Ubuntu 20.04. RT requires a few dependencies, such as Perl 5.10.1, a supported SQL database and a webserver such as Apache 2.0 or later.

RT5 README Documentation: <https://docs.bestpractical.com/rt/5.0.2/README.html>

## Dependences

Firstly, RT requires some initial base packages:

**sudo apt install build-essential apache2 libapache2-mod-fcgid libssl-dev libexpat1-dev libmysqlclient-dev libcrypt-ssleay-perl liblwp-protocol-https-perl mariadb-server mariadb-client**

Then Install Perl modules:

**sudo /usr/bin/perl -MCPAN -e shell**

When prompted, input yes then q to quit.

Installing the required Perl modules:

**sudo cpan install HTML::FormatText HTML::TreeBuilder HTML::FormatText::WithLinks HTML::FormatText::WithLinks::AndTables DBD::mysql LWP::Protocol::https**

## RT Installation

CD into /tmp folder to download and install RT5

**cd /tmp**

Download the RT5 file in the /tmp directory:

**wget** [**https://download.bestpractical.com/pub/rt/release/rt-5.0.0.tar.gz**](https://download.bestpractical.com/pub/rt/release/rt-5.0.0.tar.gz)

then Extract the file:

**tar xzvf rt-5.0.0.tar.gz -C /tmp**

Run the configuration script:

**cd /tmp/rt-5.0.0**

**sudo ./configure**

Ensure the required perl and libraries are installed:

**sudo make testdeps**

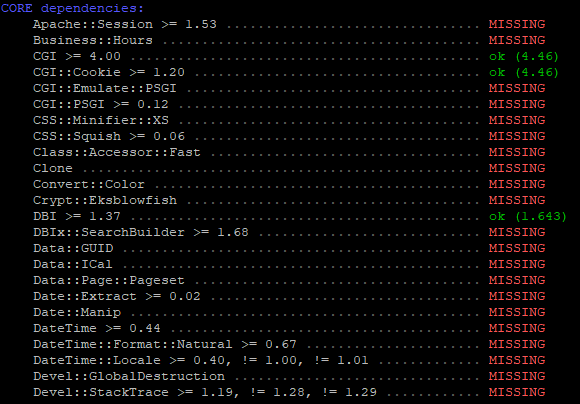


Figure Dependencies missing

if there are missing dependencies reported:

**sudo make fixdeps**

answer Y to all three prompted in the console. The script may take a while to complete,

once completed validate that all dependencies are present by running the testdeps script again.

**sudo make testdeps**

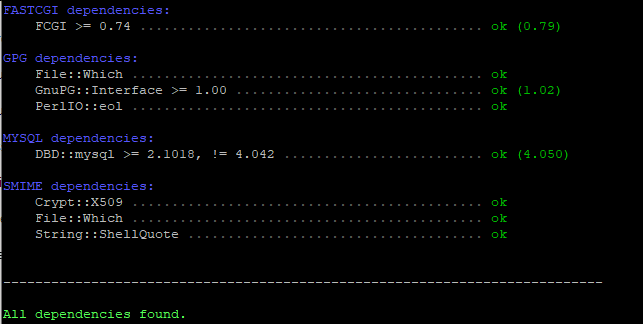


Figure All dependencies are installed

Run the make install command:

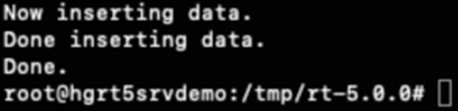
**sudo make install**

initialize the database for RT

**sudo make initialize-database**

*If the initialization fails run make dropdb and then rerun sudo make initialize-database*.

Once completed, you should see the following:



It is recommended to change the default password for the RT database user:

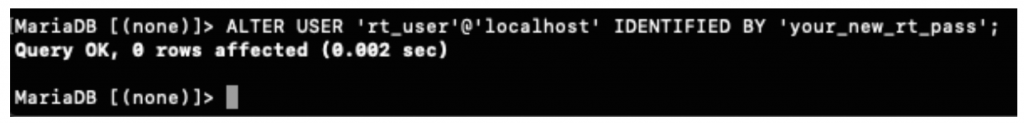
**sudo mysql -u root -p**

You’ll be prompted for a password, just hit the enter key and continue.

Changing the default user password:

**ALTER USER 'rt\_user'@'localhost' IDENTIFIED BY 'your\_new\_rt\_pass';**

**commit;**

**quit;**

Text

Description automatically generatedUpdate the RT\_config.pm config to reflect the password change we did for the RT database user:

**Original**

Text

Description automatically generated**Updated**

Confirm we have a working RT instance running with the standalone RT server:

**sudo /opt/rt5/sbin/rt-server -port 8080**

Open your web browser and navigate to your servers ip and port 8080. For example, http:// 20.92.235.14:8080/. You should see the login page.

## Web Server Configuration

Create a RT5 sites-available config file:

**sudo vi /etc/apache2/sites-available/rt5.conf**

populate this file with the following:

<VirtualHost \*:80>

### Optional apache logs for RT5 request tracker for HendGrow.com Demo server

# Ensure that your log rotation scripts know about these files

# ErrorLog /opt/rt5/var/log/apache2.error

# TransferLog /opt/rt5/var/log/apache2.access

# LogLevel debug

AddDefaultCharset UTF-8

ScriptAlias / /opt/rt5/sbin/rt-server.fcgi/

DocumentRoot "/opt/rt5/share/html"

<Location />

Require all granted

Options +ExecCGI

AddHandler fcgid-script fcgi

</Location>

</VirtualHost>

Next, edit the apache2.conf file with the following information below:

**sudo vim /etc/apache2/apache2.conf**

<Directory /opt/rt5/share/html>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

Enable the RT5 site:

**sudo a2ensite rt5**

disable the default site:

**sudo a2dissite 000-default**

check the config:

**sudo apachectl configtest**

restart apache:

**sudo systemctl restart apache2**

open your web browser and navigate to your server ip, for example: http://20.92.235.14 You should see the login page. The default username for admin access is root, default password is password. I advise you to change the default password immediately by going to Admin > Users > Select, select root and update the password.

## CSRF Error

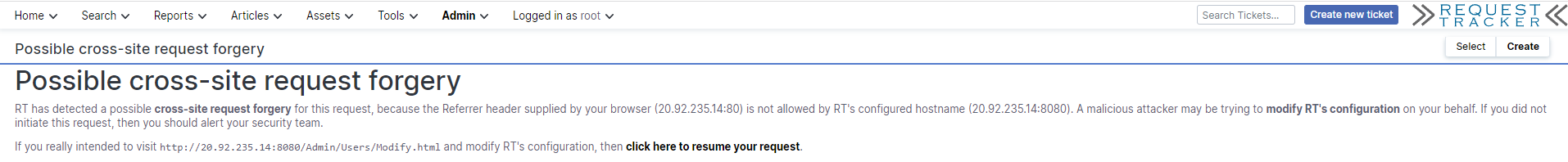


Figure CSRF Error

When RT is installed, you may come across the above Cross-Site Request Forgery(CSRF) message when navigating through the site. The reason causes the CSRF, is an attack that forces authenticated users to submit a request to a Web application against which they are currently authenticated. Basically, CSRF attacks exploit the trust a Web application has in an authenticated user.

To fix this issue, we need to add our local host, IP address, port number and domain to the white host, to get special recognition, privileges, or access. By edit the RT\_SiteConfig.pm file, located in **/opt/rt5/local/etc/**. In the bottom of the file, add the following line with your IP address and if applicable your FQDN:

**Set (@ReferrerWhitelist, qw(“***example IP”, “example FQDN”);*

Text

Description automatically generated

Figure add ip address with port 80 to the whitelist

Don’t forget to save the file and restart the Apache server to apply the configs with the following command:

**sudo systemctl restart apache2**

## Email Implementation

One of the main functionalities in RT is to send and receive emails. To be able to do this, we’ll need to configure a mail server for outgoing email and pull incoming mail into RT’s mail gateway, which will internally update or create a ticket. To achieve this, we’ll be setting up postfix and fetchmail. Postfix is a mailserver application that is relatively easy to setup in linux.

### **Mail Account setup**

Before postfix is configured, a mail account needs to be created for your RT instance. For RT, we’re using an outlook email account. Navigate to outlook.com, under settings go to Mail Sync email > POP and IMAP. Select “let devices and apps use POP” and save the changes.

The reason we use sync (**synchronization**), because it will sync outlook data with RT website.  These data basically is the email and it will be sync to the RT as a ticket. Once they are fetched onto RT page, they are deleted from the email inbox.

There are three settings in SYNC email: POP, IMAP and SMTP.

* POP (Post Office Protocol) works by contacting your email service and downloading all of your new messages from it.
* SMPT (Simple Mail Transfer Protocol): allows you to configure RT website to send emails using your Outlook account.
* IMAP (short for Internet Message Access Protocol): lets you sync your email inbox across multiple devices.

Graphical user interface, application

Description automatically generated

Figure Key email setting

### **Application Password**

An application password is also required for authentication. To create an app password, two factor authentications is required to be setup on the outlook account. Head to outlook.com, select your profile and click on “My Microsoft account”. Here you may be asked to setup 2FA.

Graphical user interface, application

Description automatically generated

Figure Set the key email

Navigate to your account’s security dashboard, click on the security dropdown and select “additional security options”.

Graphical user interface, text, application, email

Description automatically generated

Figure Change the password

Scroll down to find App passwords, click on “Create a new app password”. Take note of that password.

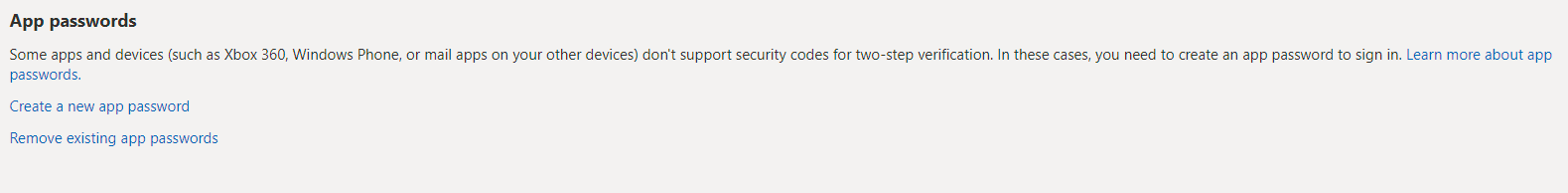


Figure Create New Password

## Postfix Installation

Now that the mail account is setup, the next step is to install and configure postfix. Run the following command:

**sudo apt-get install postfix**

The installation process will open a series of prompts. Select internet with smarthost.

Text

Description automatically generated

On the system mail name prompt, Press OK. Leave the default entry in the system mail name.

Graphical user interface, text, application

Description automatically generated

On the next prompt when asked to specify a SMTP relay, enter **smtp.office365.com.**

Graphical user interface, text, email

Description automatically generated

Press enter and let postfix install. Next, we’ll need to edit the main config file for postfix and add the following lines. Add your email address and application password into the applicable fields. The config file is found under **/etc/postfix/main.cf.**

smtp\_sasl\_password\_maps = static:***”your email address”***:***”your app password”***

smtp\_sasl\_auth\_enable = yes

smtp\_tls\_security\_level = may

smtp\_sasl\_security\_options = noanonymous

relayhost = [smtp.office365.com]:587

Text

Description automatically generatedBelow is an example of the lines added to the config file.

Figure postfix config file

Restart postfix with **sudo service postfix restart** to apply changes within the config file.

## RT User Creation

Login to RT as root and create a new user to use the email address. This user will be names support and will be used to respond to tickets. Select Admin > Users > Create from the RT navigation menu.

Graphical user interface, text, application, email

Description automatically generated

Figure User creation page

Next, navigate to Admin > Queues > Select. Select the general queue.

Graphical user interface, application

Description automatically generated

Figure Go to the queue page

Click on “User Rights” in the top right corner.

Graphical user interface, application

Description automatically generated

Figure Click User right

Select your user you wish to grant rights to. I ticked all the General rights and Staff rights for the support user, select save changes to apply these permissions.

Graphical user interface, text, application

Description automatically generated

Figure Give user privilege

## Testing Outgoing Mail

Finally, we’ll test that outgoing mail works properly. We can do this by logging in as the support user and creating a ticket. Click on Create new ticket and simply create a ticket for testing.

Graphical user interface, text, application

Description automatically generated

Figure Testing Outgoing Mail

On the RT server we can check the mail logs to verify that the mail was sent. The logs are found under **/var/log/mail.log**

**tail -20f /var/log/mail.log**

Graphical user interface, text, application

Description automatically generatedLook for status=sent to verify mail delivery. You can also verify mail delivery by logging into the mail account and viewing sent items.

Figure Verify mail delivery

## Fetching Incoming Mail

Tickets can be generated through incoming email. Users can simply email the helpdesk address rather than logging into the RT website to submit a ticket. To enable this, we need to fetch the emails from the address and then forward them in RT’s system, we’ll do this by installing fetch mail. Run the command below to get started.

**sudo apt-get install fetchmail**

stop the fetchmail service after installation.:

**sudo fetchmail stop**

now create our fetchmail config file:

**sudo vim /etc/fetchmailrc**

populate the file with these lines, change the parameters applicable to you:

**poll outlook.office365.com**

**protocol pop3**

**user** ***"your mail address"***

**password *"your app password"***

**mda '/opt/rt5/bin/rt-mailgate --url** ***“FQDN here”* --queue General –action correspond'**

example fetchmailrc config file:

Figure fetchmailrc config

restart fetchmail with:

**sudo systemctl restart fetchmail**

navigate back to RT, login as root. Select the general queue under Admin > Queues > Select. Click on group rights at the top menu.



Graphical user interface, text, application

Description automatically generatedSelect “Everyone” under system and tick “Create tickets”. Save changes. Now, send an email to the address configured for fetch mail so that fetch mail has content it can fetch. I sent an email from my school account to op-rt-test@outlook.com.

Figure Select privilege to Everyone, who can create ticket.

Set permissions on the fetchmail rc file:

**sudo chmod 0700 /etc/fetchmailrc**

test fetchmail manually with this command:

**sudo fetchmail -f /etc/fetchmailrc**

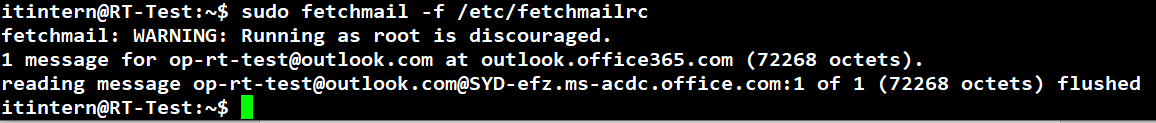
you should see the following output:

Figure Manual fetch mail from inbox

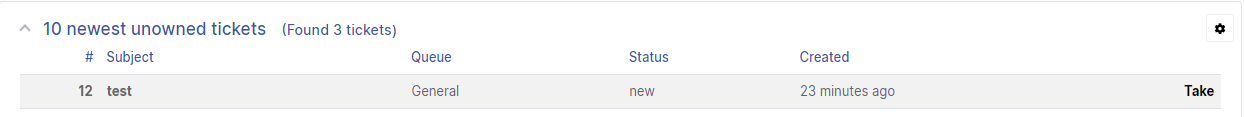
if you navigate back to the RT dashboard you should see a new ticket in the general queue:

Figure New ticket has been generated

Finally edit crontab to ensure fetch mail checks the inbox every 5 minutes or to whatever frequency you desire:

**sudo crontab -e**

add the following line to your crontab file:

**\*/5 \* \* \* \* fetchmail -f /etc/fetchmailrc**

Graphical user interface, application

Description automatically generated

Figure Check and fetch mail from inbox every 5 mins

Now the RT system can send and receive emails through the general queue.

## Queue

Queues are how different sectors of the ticketing system are defined. You can think of them like user groups. Users in one queue won’t get alerts or be expected to work on tickets assigned to a different queue. Ideally the number of queues shouldn’t fluctuate at all and you’re expected to keep it to as keep it to as much of a minimum as you can. As of current, in the RT website, we have Ops-Team for the Ops project group as well as groups A through E, these are used in Operations Engineering 2 for the students work groups.

Graphical user interface, text, application, email

Description automatically generated

Figure Create a new queue

Graphical user interface, application

Description automatically generated

Figure All queues

When creating a new queue, it’s important that you follow the conventions shown in the picture below. This is to ensure that the postfix emailing service will route alerts through to the appropriate groups.

Graphical user interface, text

Description automatically generated

Figure Set up queue on aliases file

Once you’ve created a new group, you need to edit the aliases file on the virtual machine, this is to create aliases for the addresses you just entered in. This is to allow RT to properly pipe it through to PostFix and then out to the user. This document can be edited with the command;

**sudo vim /etc/aliases**

The convention should be self-explanatory, simply copy what already exists, each line should be functionally identical aside from the prefix. For each queue you’ll need to add the following with the-queue-name substituted out for whatever you named your queue:

The-Queue-Name "|/opt/rt5/bin/rt-mailgate --queue The-Queue-Name --action correspond --url http://rttest.ict.op.ac.nz"

The-Queue-Name-Comment: "|/opt/rt5/bin/rt-mailgate --queue The-Queue-Name --action comment --url http://rttest.ict.op.ac.nz "

## Customizing Templates

Ticket templates can be customized globally or for each individual queue. Global templates are found under Global > Templates > Select. Individual templates for each queue can be found under templates when viewing an individual queue under Admin > Queues. Here you can find and create custom ticket email templates. These email templates are used to notify RT users when actions happen within RT like a ticket being created or deleted.

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, application

Description automatically generated

Figure Customizing Templates

### **Users**

Graphical user interface, text, application, email

Description automatically generatedUsers will need to be created at the start of semester for each new student added to the Ops group. The process for this is relatively simple and straight forward to understand. The important parts here are ensuring that the username and email are correct as well as assigning the correct timezone (Pacific/Auckland +1200). I would recommend giving each user a password using in keePass. Once created, navigate to the Admin tab -> global -> user rights. Enter the username of the account you just created on the text box on the left hand side of the page, then under the ‘Rights for Staff’ tab check the value ‘ModifySelf’ and hit save changes. This will allow users to then change their password to whatever they wish.

## SSL Certs

### **Install SSL on RT**

Install SSL certification on RT, gained SSL files from Teams to our relevant DNS. Manually rewrote RT5 config files to enable site security.

In teams file, under Documents>General>Keys>SSL Certs, I find files specific for our DNS ict.op.ac.nz, these being a ca-bundle, crt, key and pem file alongside instructions for how to go about it.

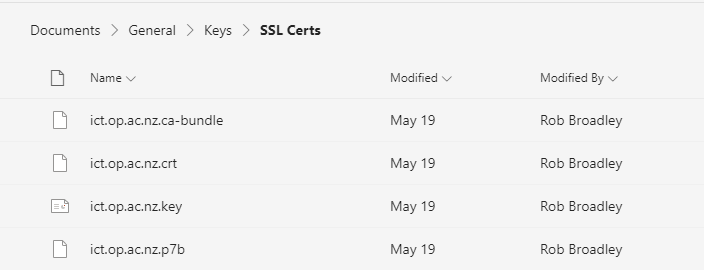


Figure SSL Certs file

First, we needed to upload the SSL files locally to Linux Ubuntu, I recommend use WinSCP to achieve this. Also, save the files to a random folder, that is not secure. However, there is a designated location in Linux for storing SSL certificates. **/etc/ssl/certs/** is the right location to store the SSL files.

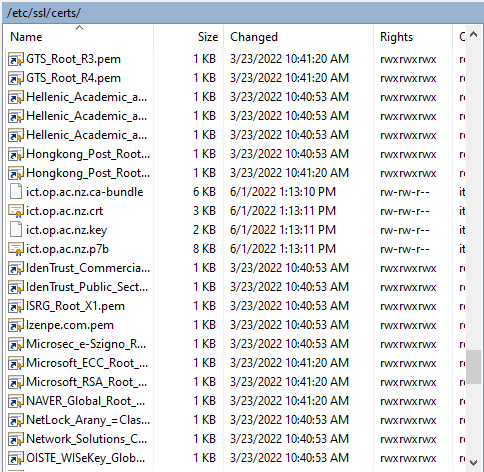


Figure Location to store certs file

After upload SSL certificate to the VM, we also need to enable the SSL within the rt5.conf file within Apache. However, we can locate the file location by querying the command **sudo find /etc -name rt5\***

C:\Users\cuip1\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\1F52C0C9.tmp

Figure Open the config file

Also, we need to add some clause to make the RT website can use the SSL certificate. Add SSLCertificateFile and SSLCertificateKeyFile to enable the SSL.

In addition, we enable the SSL on port 443, because HTTPS is secure and is on port 443, while HTTP is unsecured and available on port 80. Port 443 encrypts network data packets before data transmission takes place. The security over port 443 is used by the SSL protocol (secure socket layer).

But we still keep port 80, because allowing port 80 doesn't introduce a larger attack surface on server, one of the main reasons to keep port 80 open is to continue to redirect traffic from HTTP to HTTPS.

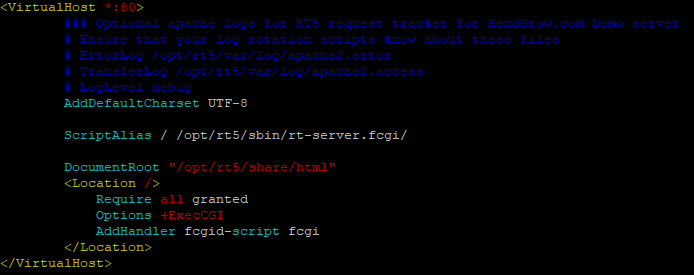


Figure Before config

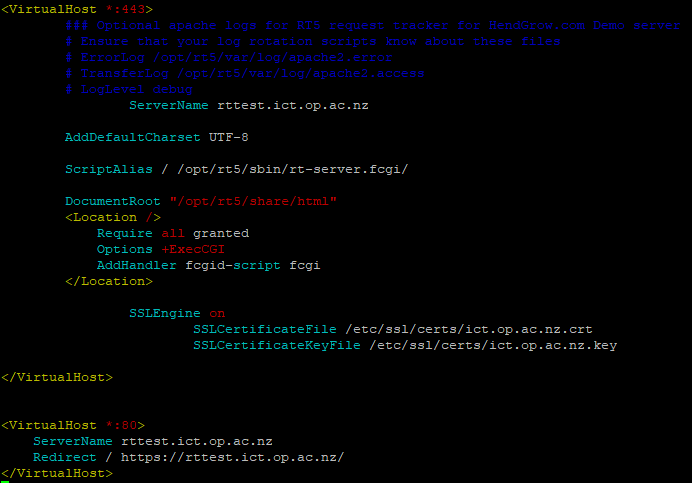


Figure After config

However, there is an error when restart the apache2 module. Because the SSL module haven't enabled yet. To fix this error by running: **sudo a2nmod ssl**

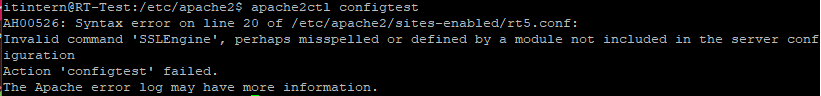


Figure error can't restart apache2

We also need to open the port 443 on NSG. Because, the port that’s responsible for handling all unencrypted HTTP web traffic is port 80. When we use a SSL certificate, the communication channel between the browser and the server gets encrypted to protect all sensitive data exchanges. All such secure transfers are done using port 443.

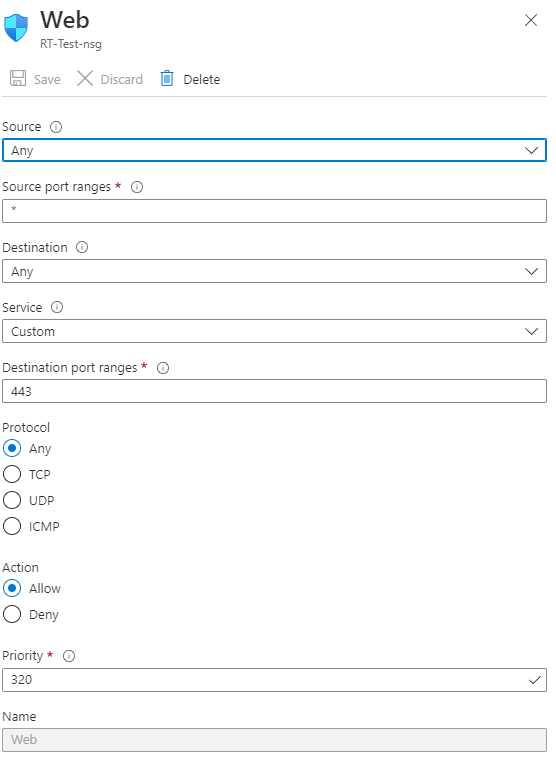
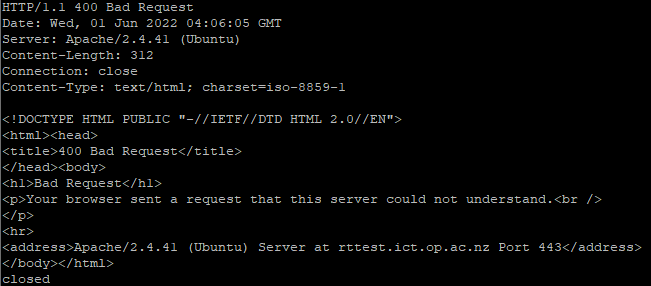


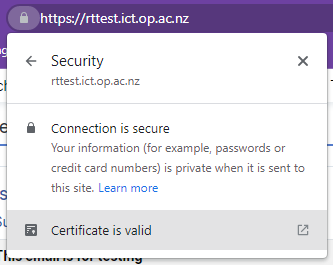
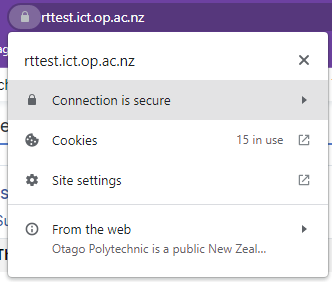
Figure Open port 443

One of the handiest tools in the OpenSSL toolbox is **s\_client**. We can quickly view lots of details about the SSL certificates installed on a particular server and diagnose problems. For example, use this command to look at RT SSL certificates: **openssl s\_sclient –connect rttest.ict.op.ac.nz:443**



The way to check SSL is installed or not?

1. Click the padlock icon in the address bar for the website.
2. Click on Certificate (Valid) in the pop-up.
3. Check the Valid from dates to validate the SSL certificate is current



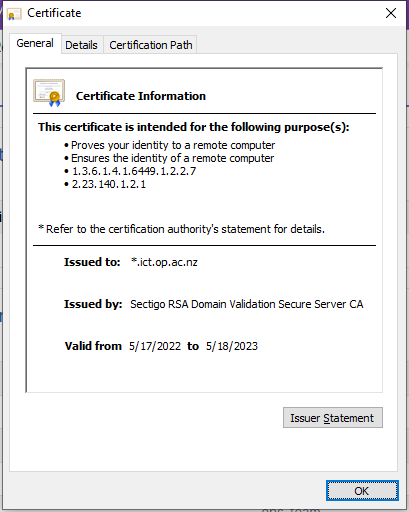


Figure SSL has been installed on RT website

However, the CSRF has shown again, this is because we still using port 80, but after we install SSL on RT website the communication channel between the browser and the server gets encrypted to protect all sensitive data exchanges. All such secure transfers are done using port 443. So, we need to change the port to 443 in RT\_Config file.

