## Relevant files on reasoning-technology-server-1 (Debian 10):

# find / -name 'nginx'

/usr/share/doc/nginx # no docs here

/usr/share/nginx # (\*1) use /var/www instead

/usr/sbin/nginx

/usr/lib/nginx # holds modules directory

/etc/logrotate.d/nginx

/etc/default/nginx # extra options for starting/stopping service

/etc/ufw/applications.d/nginx #firewall rules, gives it 80 and 443

/etc/init.d/nginx

/etc/nginx # (\*2) config files

/var/lib/nginx # body, fastcgi, proxy, scgi, uwsgi directories

/var/log/nginx # access and error logs

(\*1) /usr/share/nginx holds module and html directories. These are files that come with the nginx distribution and will not be modified later. By convention and by standard, data in the /usr tree is not supposed to be modified by users (ironic isn’t it?). The user modifiable version of this directory is /var/www which also has an html subdirectory.

(\*2) One or more config files in this directory are modified to point to our website code.. One would suppose that directives (blocks) in such config files will point

Into the /var/www tree.

Main config file for nginx: /etc/nginx/nginx.conf

The wsgi parms template: /etc/nginx/uwsgi\_params

## Server Status and Control

# systemctl enable|start|restart|status|stop nginx

# nginx -t

# ufw status

**Test That The Server is Running**

To see the ngnix welcome panel after the server is running though it only has a default configuration, we must use the IP address, not the hostname. http:35.194.71.194 replies with the panel (see the journal entry) In contrast, <http://reasoningtechnogy.com> replies with ‘refused to connect’ even though the DNS records are correct. The access log shows the attempt to connect, but it is not considered an error, and there is no entry in the error log.

**Fix “Failed to parse PID from file /run/nginx”** shows on sysctl status nginx

(note: <https://www.cloudinsidr.com/content/heres-fix-nginx-error-failed-read-pid-file-linux/>)

# umask

0022

# mkdir /etc/systemd/system/nginx.service.d

# printf "[Service]\nExecStartPost=/bin/sleep 0.1\n" > /etc/systemd/system/nginx.service.d/override.conf

# systemctl daemon-reload

# systemctl restart nginx

**Install and run ufw**

Note that when a port is specified it should be followed by a protocol. E.g. 22/tcp. Here I take advantage of the ufw shortcut symbols.

# ufw status

Status: inactive

…

# ufw reset

# ufw default deny

Default incoming policy changed to 'deny'

(be sure to update your rules accordingly)

# ufw allow "WWW Full"

Rules updated

Rules updated (v6)

# ufw allow ssh

Rules updated

# ufw enable

…

# systemctl enable ufw

…

**Consider adding a new rule to change the ssh port**, it would look like this:

“There is a SSH rule in ufw-loginserver but it defaults to port 22. I can either change it or create a new rule. The latter seems better and will probably survive new versions of ufw better. So I add a file called ufw-loginserver-custom and copy the ssh rule with just a change to the port:”

[CUSTOMSSH]

title=SSH server

description=SSH server

ports=667/tcp

Then

# ufw delete allow ssh

# ufw allow CUSTOMSSH

Perhaps add rules to allow ssh from specific ips.

**Directory name in /var/www/html must be identical to the domain name**

The python scripts for lets encrypt require the name to be the same as the domain name. In our case reasoningtechnology.com. /var/www/html/reasoningtechnology.com

**Install SSL Certificate**

See doc on this topic..

# apt install python3-acme python3-certbot python3-mock python3-openssl python3-pkg-resources python3-pyparsing python3-zope.interface

# apt install python3-certbot-nginx

# systemctl reload nginx

# sudo certbot --nginx -d reasoningtechnology.com -d www.reasoningtechnology.com

Conf files:

<https://docs.nginx.com/nginx/admin-guide/web-server/serving-static-content/>

Nginx variables:

<http://nginx.org/en/docs/varindex.html>