Tech Startup System Documentation

- 🧩 Introduction
  - Document change log
- 🧩 System overview
  - 💦 System components
  - 💦 System network diagram
  - 💦 Hosts
  - 💦 Access List Rules (ACLs)
    - Pouta dashboard Security Groups
  - 💦 Users
  - 💦 System security configuration
    - SSH configuration
    - Automatic updates
    - Host firewall (ufw) configuration
  - 💦 Installed packages
  - 💦 Backup and disaster recovery
    - Snapshots
    - CMDB repository
- 🧩 Services
  - 🍃 Tip of the day service
    - 🌈 TotD - local file or NFS based
    - 🌈 TotD - HTTP static
    - 🌈 TotD - NodeJS application
    - 🌈 TotD 2.0 - NodeJS application with Postgres DB (docker containers)
  - 🍃 Apache Web server
  - 🍃 Let’s Encrypt & Certbot
  - 🍃 PM2
  - 🍃 NFS (Network File System)
- 🧩 CMDB
  - Configuration management database In the local system
  - Git repo URL for server configuration files
- 🧩 Quality and operation requirements

🧩 Introduction

This document contains the system operations documentation for the “TomatoTech” services (imaginery startup tech company).
The environment has been built in the Server Technologies 2023 course by Jingjing Yang.
It contains information required for operating the system and describes the network setup, hosts, services and users of the system.

Document change log

Version	Date	Author	Description
1.0	29.3.2023	Jingjing Yang	Initial version. Server setup with public IP and default user. No services, no other users.
2.0	01.4.2023	Jingjing Yang	Users-added. Default user disabled. Allow SSH key based access only. Enabled host firewall and allow SSH only. Enabled automatic updates.
2.1	01.4.2023	Jingjing Yang	Added details for local file based Totd service.
2.2	15.4.2023	Jingjing Yang	Added details for HTTP static Totd service, using apache.
2.3	18.4.2023	Jingjing Yang	Added details for Totd Node.js application, pm2.
2.4	23.4.2023	Jingjing Yang	Added details for apache Reverse-Proxy, Let’s Encrypt.
3.0	25.4.2023	Jingjing Yang	Added details for Totd service in docker containers, nfs service, nfs-based Totd.
3.1	28.4.2023	Jingjing Yang	Added CMDB, disaster recovery.
3.2	29.4.2023	Jingjing Yang	Added details for Security, ACLs.
3.3	02.5.2023	Jingjing Yang	Added details for System overview: diagram, installed packages, etc.
4.0	03.5.2023	Jingjing Yang	Finalised network diagram.

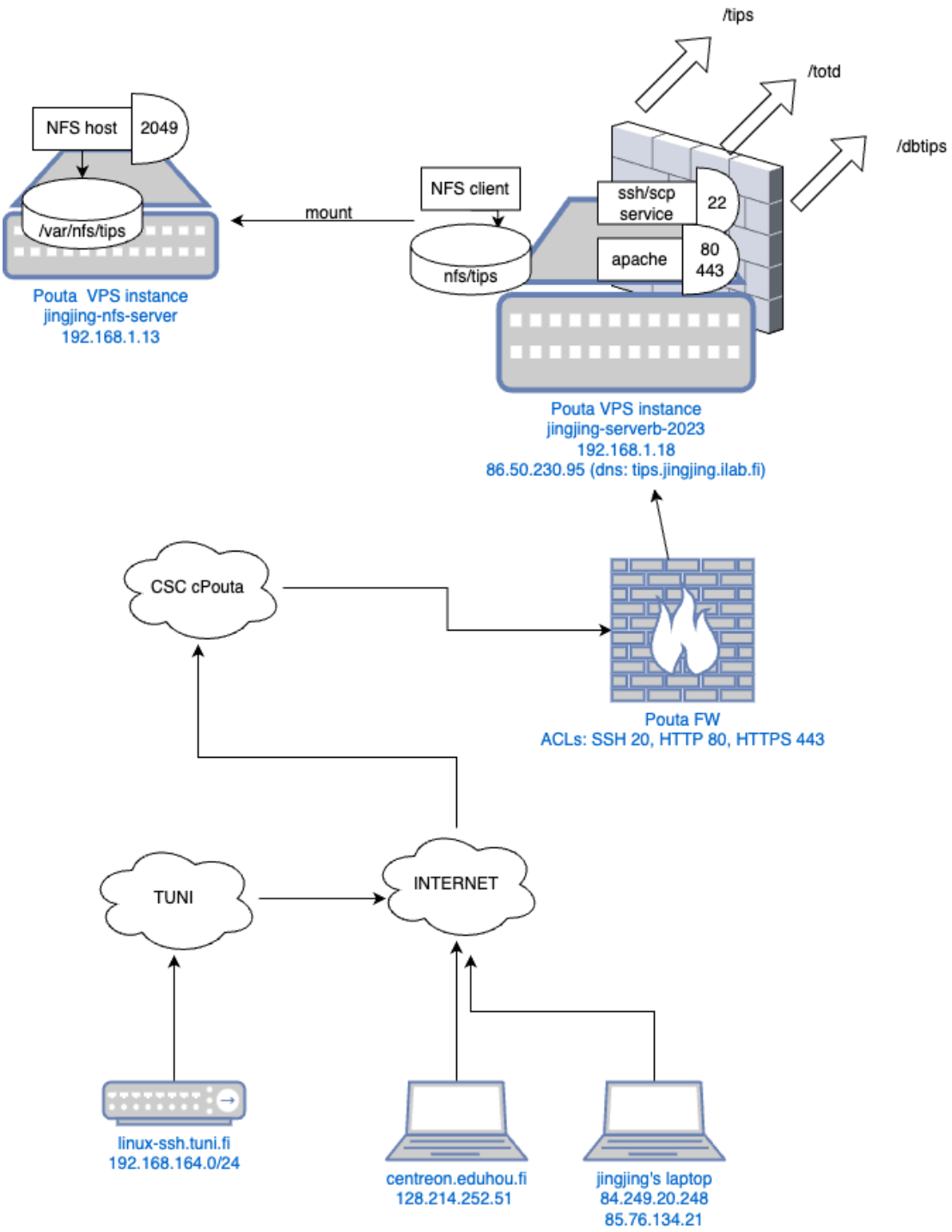
🧩 System overview

💦 System components

The system resides in the CSC cPouta IaaS cloud service in a single cPouta project.

cPouta uses the OpenStack technology.

💦 System network diagram



Hosts

Name	IP	OS	Description
jingjing-serverb-2023	192.168.1.18 Floating IPs: 86.50.230.95	Ubuntu 22.04	Main server. Facing to internet. Service TBD.
linux-ssh.tuni.fi	192.168.164.0/24	Red Hat (not our own server)	Tuni Linux server - login with Tuni-credentials. Use this to login to CSC Pouta server.

Name	IP	OS	Description
jingjing-nfs-server	192.168.1.13	Ubuntu 22.04	2nd server. Sharing a directory where tips files can be updated to and used from between users.

💦 Access List Rules (ACLs)

Enforcement	From	To	Ports	Description	
Pouta (default)	Internal	Internal	*	Internal traffic between servers allowed	
Pouta (Monitoring)	centreon.eduhou.fi 128.214.252.51 / 32	jingjing-serverb-2023	SSH, HTTP, HTTPS, ICMP:ALL	Course monitoring allowed	
Pouta (tuni eduVPN)	Tuni Linux Servers 193.166.164.0/ 24	jingjing-serverb-2023	SSH	Access from Tuni eduVPN, Linux and homepages servers	
Pouta (ssh login strict)	jingjing's laptop 84.249.20.248 / 32(home) 85.76.134.21 / 32(campus)	jingjing-serverb-2023	SSH	ssh login from my own laptop allowed	
Pouta (apache)	*	jingjing-serverb-2023	HTTP, HTTPS	apache full allowed	
Pouta (nfs sharing)	jingjing-serverb-2023 192.168.1.18 / 32	jingjing-nfs-server	2049, SSH, ICMP:ALL	NFS - the only service specific ACL !	

Pouta dashboard Security Groups

Name	Security Group ID	Description
default	cf507128-e697-474c-8359-c98be372e481	Default security group
nfs-share	14b20a4a-02e3-4c2f-b831-e8ea7fe63d8c	allow nfs mounting
ssh-strict	f90a5ace-2e97-49e2-8a41-6460e186c8f5	Allow limited SSH only

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group
Egress	IPv4	Any	Any	0.0.0.0/0	-
Egress	IPv6	Any	Any	::/0	-
Ingress	IPv4	TCP	Any	-	default
Ingress	IPv6	TCP	Any	-	default

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group
Egress	IPv4	Any	Any	0.0.0.0/0	-
Egress	IPv6	Any	Any	::/0	-
Ingress	IPv4	ICMP	Any	128.214.252.51/32	-
Ingress	IPv4	TCP	22 (SSH)	128.214.252.51/32	-
Ingress	IPv4	TCP	22 (SSH)	84.249.20.248/32	-
Ingress	IPv4	TCP	22 (SSH)	193.166.164.0/24	-
Ingress	IPv4	TCP	22 (SSH)	85.76.134.21/32	-
Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-
Ingress	IPv4	TCP	443 (HTTPS)	0.0.0.0/0	-

Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group
Egress	IPv4	Any	Any	0.0.0.0/0	-
Egress	IPv6	Any	Any	::/0	-
Ingress	IPv4	ICMP	Any	192.168.1.18/32	-
Ingress	IPv4	TCP	2049	0.0.0.0/0	-

💦 Users

jingjing@jingjing-serverb-2023:~$ ls /home/

emiliab jingjing joe22 phong sam22

jingjing@jingjing-serverb-2023:~$ ls /home/

emiliab jingjing joe22 phong sam22

User	userid	Sudo	Description
ubuntu	1000	YES	The default user in cPouta Ubuntu OS. DELETED.
joe22	2021	NO	Instructor account. Totd enabled.
sam22	2031	ufw - nopasswd	Instructor account. TOtd disabled.
jingjing	1400	YES	Jingjing Yang's username
emiliab	1420	NO	peer student in group 24

```
jingjing@jingjing-serverb-2023:~$ ls /home/
emiliab  jingjing  joe22  phong  sam22

root@jingjing-serverb-2023:~# egrep -v "(#|^$)" /etc/sudoers.d/91-stec22-users
jingjing ALL=(ALL) ALL
sam22 ALL=(ALL) NOPASSWD: /usr/sbin/ufw
```

💦 System security configuration

SSH configuration

- Disable root login via SSH

```
root@jingjing-serverb-2023:~# echo "PermitRootLogin no" >> /etc/ssh/sshd_config
root@jingjing-serverb-2023:~# grep -i '^PermitRootLogin' /etc/ssh/sshd_config
PermitRootLogin no
root@jingjing-serverb-2023:~# sudo systemctl restart ss
```

- Use key-based authentication instead of password-based authentication

```
jingjing@jingjing-serverb-2023:~$ grep -i '^PasswordAuthentication' /etc/ssh/sshd_config
PasswordAuthentication no
```

- Set up a firewall to only allow SSH access from trusted IP addresses (refer to ufw below)

Automatic updates

👉 Enable automatic reboot for security updates:

`/etc/apt/apt.conf.d/50unattended-upgrades`

```
jingjing@jingjing-serverb-2023:~$ grep '^Unattended-Upgrade::Automatic-Reboot' /etc/apt/apt.conf.d/50unatt
ended-upgrades
Unattended-Upgrade::Automatic-Reboot "false";
Unattended-Upgrade::Automatic-Reboot-WithUsers "true";
Unattended-Upgrade::Automatic-Reboot-Time "02:00";
```

Host firewall (ufw) configuration

```
jingjing@jingjing-serverb-2023:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       193.166.164.0/24           (log)
22                         ALLOW       128.214.252.51             (log)
Apache Full                ALLOW       Anywhere
22                         ALLOW       84.249.20.248              (log)
22                         ALLOW       85.76.134.21               (log)
Apache Full (v6)           ALLOW       Anywhere (v6)
```

```
jingjing@jingjing-nfs-server:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
2049                       ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
2049 (v6)                  ALLOW       Anywhere (v6)
```

💦 Installed packages

`$ sudo apt update; sudo apt upgrade`

`$ sudo apt install <package>`

`$ sudo npm install -g pm2` or shorthand command `sudo npm i -g pm2`

Packages	Host	Description
apache2	jingjing-serverb-2023	Apache Web server

Packages	Host	Description
nodejs	jingjing-serverb-2023	Node.js along with the Node Package Manager (npm
docker	jingjing-serverb-2023	run applications in containers
docker-compose	jingjing-serverb-2023	run multi-container Docker applications
pm2	jingjing-serverb-2023	start the node.js application automatically
certbot	jingjing-serverb-2023	obtain a free SSL certificate for for Apache web server
python3-certbot-apache	jingjing-serverb-2023	integrates Certbot with Apache
nfs-common	jingjing-serverb-2023	NFS client
nfs-kernel-server	jingjing-nfs-server	NFS server
git	ALL	fetch from gitlab TOTD repo; version control of gitlab CMDB repo; clone github express-tips-version-2 to local.
net-tools	ALL	ifconfig, netstat
tree	ALL	display the contents of a directory (eg. cmdb) in tree-like format

```
jingjing@jingjing-serverb-2023:~$ apache2 -v
Server version: Apache/2.4.52 (Ubuntu)
Server built:   2023-03-08T17:32:01

jingjing@jingjing-serverb-2023:~$ node -v
v18.16.0
jingjing@jingjing-serverb-2023:~$ npm -v
9.6.5

jingjing@jingjing-serverb-2023:~$ pm2 -v
5.3.0

jingjing@jingjing-serverb-2023:~$ certbot --version
certbot 1.21.0

jingjing@jingjing-serverb-2023:~$ docker -v
Docker version 20.10.21, build 20.10.21-0ubuntu1~22.04.3
jingjing@jingjing-serverb-2023:~$ docker-compose -v
docker-compose version 1.29.2, build unknown

jingjing@jingjing-serverb-2023:~$ dpkg -l nfs-common | grep ii
ii  nfs-common     1:2.6.1-1ubuntu1.2 amd64        NFS support files common to client and server

jingjing@jingjing-serverb-2023:~$ git --version
git version 2.34.1
jingjing@jingjing-serverb-2023:~$ ip -V
ip utility, iproute2-5.15.0, libbpf 0.5.0
jingjing@jingjing-serverb-2023:~$ tree --version
tree v2.0.2 (c) 1996 - 2022 by Steve Baker, Thomas Moore, Francesc Rocher, Florian Sesser, Kyosuke Tokoro
```

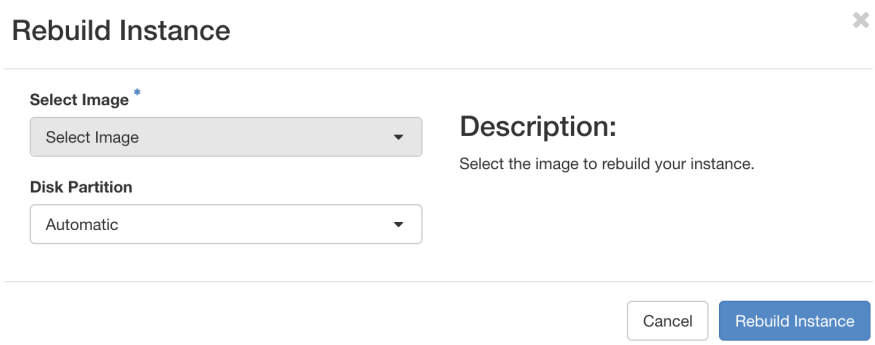
```
jingjing@jingjing-nfs-server:~$  dpkg -l nfs-kernel-server | grep ii
ii  nfs-kernel-server 1:2.6.1-1ubuntu1.2 amd64        support for NFS kernel server
```

```
jingjing@jingjing-nfs-server:~$ git --version
git version 2.34.1
jingjing@jingjing-nfs-server:~$ ip -V
ip utility, iproute2-5.15.0, libbpf 0.5.0
jingjing@jingjing-nfs-server:~$ tree --version
tree v2.0.2 (c) 1996 - 2022 by Steve Baker, Thomas Moore, Francesc Rocher, Florian Sesser, Kyosuke Tokoro
```

💦 Backup and disaster recovery

Snapshots

A snapshot is an image which preserves the disk state of a running instance. You can rebuild instance based on a specific snapshot from Pouta Dashboard for system recovery.



Snapshot (description in naming)	Date	Server
jingjing-serverB-2023-monitors added	4/7/23	jingjing-serverb-2023
jingjing-serverB-2023-before proxy	4/16/23	jingjing-serverb-2023
jingjing-serverB-2023-pm2 done	4/17/23	jingjing-serverb-2023
jingjing-serverB-2023-before nfs	4/22/23	jingjing-serverb-2023
jingjing-serverB-2023-before docker	4/22/23	jingjing-serverb-2023
jingjing-serverB-2023-totd2.0 done	4/28/23	jingjing-serverb-2023
jingjing-nfs-server-tips mount done	5/1/23	jingjing-nfs-server

CMDB repository

A Configuration Management Database or CMDB is a valuable repository that stores lists of information and relationships between individual Configuration Items (CIs). With this centralized location, getting a comprehensive view of the organization’s components becomes easier.

Configuration information about hardware, software, systems, facilities, and even personnel can be stored within the CMDB. It is one common location to store information and can help provide visibility and keep things organized.

```
jingjing@jingjing-serverb-2023:~$ nslookup tips.jingjing.ilab.fi
Server:   127.0.0.53
Address:  127.0.0.53#53

Non-authoritative answer:
tips.jingjing.ilab.fi canonical name = jingjing.ilab.fi.
Name: jingjing.ilab.fi
Address: 86.50.230.95
```

```
jingjing@jingjing-serverb-2023:~$ nslookup www.jingjing.ilab.fi
Server:   127.0.0.53
Address:  127.0.0.53#53

Non-authoritative answer:
www.jingjing.ilab.fi  canonical name = jingjing.ilab.fi.
Name: jingjing.ilab.fi
Address: 86.50.230.95
```

🧩 Services

🍃 Tip of the day service

▼ 🌈 TotD - local file or NFS based

Installation

Clone the repo: https://gitlab.tamk.cloud/server-tech-2023b-jingjing-yang/tip-of-the-day-1.0-bash-script to the service directory.

This bash script displays a different "Tip of the day" each time it is run.

- Round robin for the tips: 1st, 2nd … last, 1st …
- Prompts for showing more tips

Start / stop

To deactivate the totd functionality for a single user, please create a file `.notips` in the home directory of that user - `$ touch ~/.notips`.

Service directory

`/opt/totd/`

```
jingjing@jingjing-serverb-2023:~$ ls /opt/totd/
README.md  linuxtips.sh  tips

jingjing@jingjing-serverb-2023:~$ ls /opt/totd/tips/
1.txt  10.txt  2.txt  3.txt  4.txt  5.txt  6.txt  7.txt  8.txt  9.txt
```

Configuration

- Put the script in `/etc/profile.d/` by creating symbolic link, so that the totd functionality is activited globally for all users in a host.

```
jingjing@jingjing-serverb-2023:/opt$ cd /etc/profile.d/
jingjing@jingjing-serverb-2023:/etc/profile.d$ ln -s /opt/totd/linuxtips.sh
```

- Alternatively, you can add line `sh /full/path/to/linuxtip.sh` to each `~/.bashrc` file separately.

> 👉 Replace `/full/path/to/linuxtip.sh` with the path of your script location.

- Create an environment variable `TIPS_REPO` specifies the nfs mounted tips directory.

```
jingjing@jingjing-serverb-2023:~$ printenv | grep "TIPS"
TIPS_REPO=/nfs/tips
```

- Replace `scriptDir` in the `linuxtips.sh` script with your service directory.

```
jingjing@jingjing-serverb-2023:~$ cat /opt/totd/linuxtips.sh
#!bin/bash

tty -s || exit # check if we have interactive shell

# the path for script and tips, modify this if you have different ones!
scriptDir=/opt/totd

if [ ! "$TIPS_REPO" ]; then
  # Default value, access tips from local disk
  tipsDir=$scriptDir/tips
else
  # access from NFS server
  # TIPS_REPO=/nfs/tips   (AN ENV VAR)
  tipsDir=$TIPS_REPO
fi

# if the totd functionality is not deactivated (~/.notips does not exit)
if [ ! -f ~/.notips ]; then

          if test -f $scriptDir/.curtip; then
                    n=$(cat $scriptDir/.curtip)  # get the current tip id if .curtip exits
          else
                    echo 1 > $scriptDir/.curtip  # create file .curtip to store initial tip id
                    n=1
          fi

          # calculate total number of tips
          N=$(ls $tipsDir/*.txt | wc -l)

          date # display the current time
          # show that tip
          echo "           Tip of the Day!"

          next="y"

          # show more tips as user want
          while [ "$next" = "y" ]
          do
                    # show the tip
                    echo "-----------------Tip $n----------------------"
                    cat $tipsDir/$n.txt
                    echo "-----------------------------------------------"

                    # Prompt for skip this tip
                    # read -p "Skip this tip permanently? (y/n) " skip

                    # Prompt for showing another tip
                    read -p "Show next tip? (y/n) " next

                    # update the id for next tip
            n=$(($n%$N+1))
          done

          echo $n > $scriptDir/.curtip # store this tip id
fi
```

Testing with every new terminal session

```
jingjing@Ubuntu-MacBookPro:~$ ssh -i .ssh/id_rsa_stec23 jingjing@jingjing.ilab.fi
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-71-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue May  2 15:59:44 UTC 2023

  System load:                      0.0224609375
  Usage of /:                       7.3% of 77.35GB
  Memory usage:                     40%
  Swap usage:                       0%
  Processes:                        116
  Users logged in:                  0
  IPv4 address for br-8b959b60f93b: 172.18.0.1
  IPv4 address for docker0:         172.17.0.1
  IPv4 address for ens3:            192.168.1.18

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

10 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Tue May  2 15:59:45 2023 from 84.249.20.248
Tue May  2 16:00:03 UTC 2023
           Tip of the Day!
-----------------Tip 10----------------------
`cut`
-d (--delimiter)  to specify a delimiter to use instead of the default TAB delimiter.
-f (--fields=LIST)  Select using a specified field, a field set, or a field range.
$ cut -d ' ' -f 2 file.txt
-b (--bytes=LIST)          -c (--characters=LIST)
To extract the first five bytes from each input line:
$ cut -b 5 file.txt
---------------------------------------------
Show next tip? (y/n) y
-----------------Tip 1----------------------
`date`
1)  displays the system date and time
2)  print the time in different formats
%D – Display date as mm/dd/yy
%T – Display date as HH:MM:SS
%A – Full weekday name (e.g., Friday)
…
3)  sets the system date and time
4)  calculate future and past dates
date --date="2 year ago"
date --date="next monday"
---------------------------------------------
Show next tip? (y/n) y
-----------------Tip 2----------------------
`paste`
to join files horizontally/parallel merging
by outputting lines consisting of lines from each file specified to the standard output
$ paste file1 file2 file3 ...
separated by tab as default delimiter
```

```
set another using -d (delimiter)
-----------------------------------------
Show next tip? (y/n) n
jingjing@jingjing-serverb-2023:~$
```

🌈 TotD - HTTP static

Please refer to section “🍃 Apache Web server” below.

🌈 TotD - NodeJS application

Installation

`$ sudo apt-get install nodejs`

Clone the repo: https://gitlab.tamk.cloud/server-tech-2023b-jingjing-yang/tip-of-the-day-1.0-bash-script to the service directory.

This is a sample implementation for the TotD scenario.

Start / stop

First install the necessary dependencies: `$ npm install`

Then starting Node.js application: `$ node <app-name>`

> 👉 NOTE: run the commands in the directory where the `app.js` file resides.
>
> `/home/jingjing/tla-express-tips/nodeapp`

To stop a running Node.js application, press `Ctrl + C` in the terminal or command prompt where the Node.js application is running. This will send an interrupt signal to the Node.js process, which will cause it to stop running.

Alternatively, you can use a process manager like PM2 to manage the Node.js application. Please refer to the section “🍃 PM2” below.

Service directory

`/home/jingjing/tla-express-tips`

```
jingjing@jingjing-serverb-2023:~$ ls tla-express-tips/
README.md  bin  nodeapp  tips
jingjing@jingjing-serverb-2023:~$ ls tla-express-tips/nodeapp/
README.md  app.js  modules  node_modules  package-lock.json  package.json  public
```

Configuration

- Enabling Necessary Apache Modules

`$ sudo a2enmod proxy`

`$ sudo a2enmod proxy_http`

- Modifying the Default Configuration to Enable Reverse Proxy

```
jingjing@jingjing-serverb-2023:~$ grep "totd" /etc/apache2/sites-available/tips.jingjing.ilab.fi-le-ssl.conf
  # Node.js application for totd
```

```
   ProxyPass "/totd"  "http://localhost:3000/totd"
         ProxyPassReverse "/totd"  "http://localhost:3000/totd"
```

- Restart or reload apache

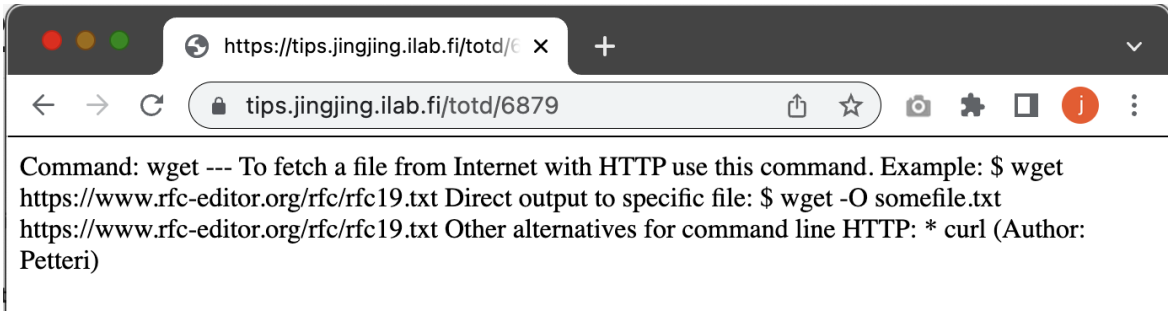
`$ sudo systemctl reload apache2.service`

Testing

```
jingjing@jingjing-serverb-2023:~$ curl localhost:3000/totd/$RANDOM
Topic: wc

wc - counts the number of characters and lines.
Example: calculate nr of lines:

$ wc -l file.tx
```



▼ 🌈 TotD 2.0 - NodeJS application with Postgres DB (docker containers)

Installation

`$ sudo apt-get install docker docker-compose`

Clone the repo: https://github.com/petteri-forks/express-tips-version-2 to the service directory.

This application is the db-based version of the Tip of the Day (TotD) application. The applications's basic use is for storing and displaying short flashcards in command line for self learning.

Each time user logs in to Linux command line, new tip of the day is displayed. The application here is the API service for delivering the data through HTTP/REST. This application contains also methods for updating the data.

Start / stop

Starting docker containers: `$ docker compose up -d`

> 👉 NOTE: run the command in the directory where the `docker-compose.yml` file resides.
> `/home/jingjing/express-tips-version-2`

Stopping containers: `$ docker compose down`

Service directory

`/home/jingjing/express-tips-version-2`

```
jingjing@jingjing-serverb-2023:~$ ls express-tips-version-2/
README-dev.md  README.md  bin  client  data  database  docker-compose.yml  nodeapp  screenshots  scrip
ts  tips
```

Configuration

```
jingjing@jingjing-serverb-2023:~$ cat express-tips-version-2/docker-compose.yml
version: '3.9'
services:
  db:
    build:
      context: ./database
      dockerfile: Dockerfile
    environment:
      POSTGRES_USER: user
      POSTGRES_PASSWORD: password
      POSTGRES_DB: tips
    ports:
      - "5432:5432"
    restart: always
    volumes:
      - ./data:/var/lib/postgresql/data
  nodeapp:
    build:
      dockerfile: Dockerfile
      context: ./nodeapp
    ports:
      - "3432:3432"
    volumes:
      - /app/node_modules
      - ./nodeapp:/app
    environment:
      - DB_PORT=5432
      - DB_HOST=db
      - POSTGRES_USER=user
      - POSTGRES_PASSWORD=password
      - POSTGRES_DB=tips
      - NODE_PORT=3432
```

- Modifying the Default Configuration to Enable Reverse Proxy

```
jingjing@jingjing-serverb-2023:~$ grep "dbtips" /etc/apache2/sites-available/tips.jingjing.ilab.fi-le-
ssl.conf
  # Node.js aplication with PostgresDB for dbtips
  ProxyPass "/dbtips"  "http://localhost:3432/plain"
        ProxyPassReverse "/dbtips"  "http://localhost:3432/plain"
```

- Restart or reload apache

`$ sudo systemctl reload apache2.service`

Logs and Diagnosis

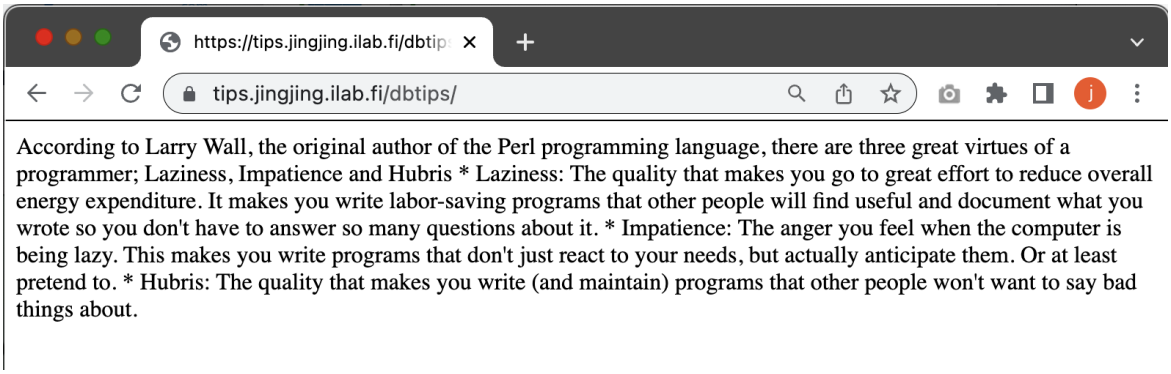
👉 Fetch the logs of a container.
Usage: `$ docker logs [OPTIONS] CONTAINER`

```
jingjing@jingjing-serverb-2023:~/express-tips-version-2$ sudo docker-compose ps
             Name                           Command               State                    Ports
-------------------------------------------------------------------------------------------------------------
express-tips-version-2_db_1        docker-entrypoint.sh postgres    Up      0.0.0.0:5432->5432/tcp,:::
5432-
                                                                            >5432/tcp
express-tips-version-2_nodeapp_1   docker-entrypoint.sh node  ...   Up      0.0.0.0:3432->3432/tcp,:::
3432-
                                                                            >3432/tcp
```

```
jingjing@jingjing-serverb-2023:~/express-tips-version-2$ sudo docker logs -f express-tips-version-2_no
deapp_1
...
```

Testing

```
jingjing@jingjing-serverb-2023:~$ curl localhost:3432/getall
{"tips":[{"id":1,"description":"Topic: chage - user account aging\n\nTo get to aging info of user test
1:\n\n  # chage -l test1\n\nSet the MAXIMUM period between password change to 500 days:\n\n  $\tsudo c
hage -M 500 test1"},{"id":2,"description":"Command: date\n\n--- With date command it is easy to format
a string with date and time information.\n$ date \"+%Y:%m:%d-%H%M%S-DATEDIR\"\n\n--- This can be used
 as part of a variable as well:\n$ export MYVAR=\"HALOO-\"date \"+%F-%T-DATEDIR\"\n$ echo $MYVAR\n\n(A
uthor: Petteri)"},{"id":3,"description":"Topic: find\n\nFind files from current location (.) with prec
ise name. For each hit, run a command - this time grep from the found file ({}) - end the command with
semicolon ... escape it to make part of parameter\n\n$ find . -name \"wisdom.txt\" -exec grep -i vote
 {} ;"},{"id":4,"description":"Topic: for loop (bash)\n\nExample one liner:\n\n$ for file in ls; do ec
ho $file; done"},{"id":5,"description":"Topic: gpasswd\n\nWith gpasswd command a user can be promoted
 to a group admin.\nSet user u1 as admin to group engineers:\n\n$ gpasswd -A u1 engineers\n\nDemote us
er u2 from engineers group admins:\n\n$ gpasswd -d u2 engineers\n\nSee the admins of the group:\n\n$ g
rep engineers /etc/gshadow"},{"id":6,"description":"Topic: LVM - Logical Volume Management\n\nLVM is a
technique for creating dynamically expandable storage volumes. A mounted volume can be expanded on the
fly with no need for restarting services using it.\n\nLogically steps for creating a LVM device are:\n
- Create a (logical) volume group and add physical devices to it.\n- Create a logical partition using
 the volume group\n- Make a file system (like ext4) to the partition and mount it\n- Afterwards the lo
gical partition can be expanded to the limit of free space in volume group and volume group by adding
 physical devices\n\nSteps to create a brand new LVM device:\n\n# lsblk  (diagnose)\n# pvcreate /dev/x
vdc /dev/xvde         (not needed necessarily)\n# vgcreate vg200 /dev/xvdc /dev/xvde   (Create the poo
l)\n# lvcreate -n vision200 -L 2G vg200    (Create a mountable partition)\n# mkfs.xfs /dev/vg200/visio
n200        (Make filesystem)\n# mount /dev/vg200/vision200 /vision200"},{"id":7,"description":"Topic:
LVM - Logical Volume Management\n\nTo expand the existing LVM partition:\n\n# lvextend -L +1G /dev/vg2
00/vision200\n# xfs_growfs /vision\n... OR:\n# lvresize -l +100%FREE /dev/vg200/vision"},{"id":8,"desc
ription":"Command: useradd\n\nExample: Create a user with userid \"userjohn\" with bash shel (-s). Cre
ate home directory (-m) automatically and provide natural name (-c). Uidnumber will be automatically g
enerated.\n\n  $ useradd -m -c \"User John\" -s /bin/bash userjohn"},{"id":9,"description":"Topic: wc
\n\nwc - counts the number of characters and lines.\nExample: calculate nr of lines:\n\n$ wc -l file.t
xt"},{"id":10,"description":"Command: wget\n\n--- To fetch a file from Internet with HTTP use this com
mand.\n\nExample:\n\n$ wget https://www.rfc-editor.org/rfc/rfc19.txt\n\nDirect output to specific fil
e:\n\n$ wget -O somefile.txt https://www.rfc-editor.org/rfc/rfc19.txt\n\n\nOther alternatives for comm
and line HTTP:\n* curl\n\n\n(Author: Petteri)"},{"id":11,"description":"According to Larry Wall, the o
riginal author of the Perl programming language, there are three great virtues of a programmer; Lazine
ss, Impatience and Hubris\n\n* Laziness: The quality that makes you go to great effort to reduce overa
ll energy expenditure. It makes you write labor-saving programs that other people will find useful and
document what you wrote so you don't have to answer so many questions about it.\n* Impatience: The ang
er you feel when the computer is being lazy. This makes you write programs that don't just react to yo
ur needs, but actually anticipate them. Or at least pretend to.\n* Hubris: The quality that makes you
 write (and maintain) programs that other people won't want to say bad things about."},{"id":12,"descr
iption":"I well remember when this realization first came on me with full force.\nThe EDSAC was on the
top floor of the building and the tape-punching and\nediting equipment one floor below. ...\nIt was on
one of my journeys between the EDSAC room and the punching equipment\nthat \"hesitating at the angles
 of stairs\" the realization came over me with\nfull force that a good part of the remainder of my lif
e was going to be spent\nin finding errors in my own programs.\n\nMaurice Wilkes. Memoirs of a compute
r pioneer. "}]}
```



▼ 🍃 Apache Web server

The Apache Web server is a powerful and flexible HTTP server that can be used to host websites and web applications. It is configured through a series of configuration files, which control various aspects of the server's behaviour.

Installation

`$ sudo apt-get install apache2`

Start / stop

Starting Apache web server: `$ sudo systemctl start apache2`

Stopping Apache web server: `$ sudo systemctl stop apache2`

Checking the status of Apache web server: `$ sudo systemctl status apache2`

Service directory

`/var/www/tips.jingjing.ilab.fi/`

```
jingjing@jingjing-serverb-2023:~$ ls /var/www/tips.jingjing.ilab.fi/
index.html  tips
```

Configuration

- Creating the Directory Structure and Granting Permissions

`$ sudo mkdir -p /var/www/tips.jingjing.ilab.fi`

`$ sudo chown -R jingjing:jingjing /var/www/tips.jingjing.ilab.fi`

- Creating Default Pages for Each Virtual Host

```
jingjing@jingjing-serverb-2023:~$ grep -v "^$" /var/www/tips.jingjing.ilab.fi/index.html
<!doctype html>
```

```
<html lang="en" data-bs-theme="dark">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Server Technologies 2023 - jingjing</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="s
tylesheet"
        integrity="sha384-GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigi
n="anonymous">
</head>
<body>
    <div class="container-fluid">
        <h1> Server Technologies 2023 - Jingjing Yang </h1>
        <p>
        <ul>
            <li> <a href="https://pouta.csc.fi" target="_blank"> Pouta </a> </li>
            <li> <a href="https://csc.fi" target="_blank"> CSC </a> </li>
      <li> <a href="http://tips.jingjing.ilab.fi/tips"> Tips </a> </li>
        </ul>
        <h2>Imaginery startup tech company</h2>
        <p>
      Tomato Tech
        </p>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-w76AqPfDkMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwxhTBTkF7CXvN" crossorigin="anony
mous"></script>
</body>
</html>
```

- Creating New Virtual Host Files, and configure virtual server tips.<server_name>.ilab.fi

  `/etc/apache2/sites-available/tips.jingjing.ilab.fi.conf`

```
jingjing@jingjing-serverb-2023:~$ egrep -v "(#|^$)" /etc/apache2/sites-available/tips.jingjing.ilab.f
i.conf
<VirtualHost *:80>
  ServerName tips.jingjing.ilab.fi
  ServerAlias www.jingjing.ilab.fi jingjing.ilab.fi
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/tips.jingjing.ilab.fi
  ErrorLog ${APACHE_LOG_DIR}/tips.jingjing.error.log
  CustomLog ${APACHE_LOG_DIR}/tips.jingjing.access.log combined
RewriteEngine on
RewriteCond %{SERVER_NAME} =www.jingjing.ilab.fi [OR]
RewriteCond %{SERVER_NAME} =jingjing.ilab.fi [OR]
RewriteCond %{SERVER_NAME} =tips.jingjing.ilab.fi
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>
```

- Enabling the New Virtual Host Files

`$ sudo a2ensite /etc/apache2/sites-available/tips.jingjing.ilab.fi.conf`

`$ sudo systemctl reload apache2`

- Secure the website using Let’s Encrypt & Certbot

Logs and Diagnosis

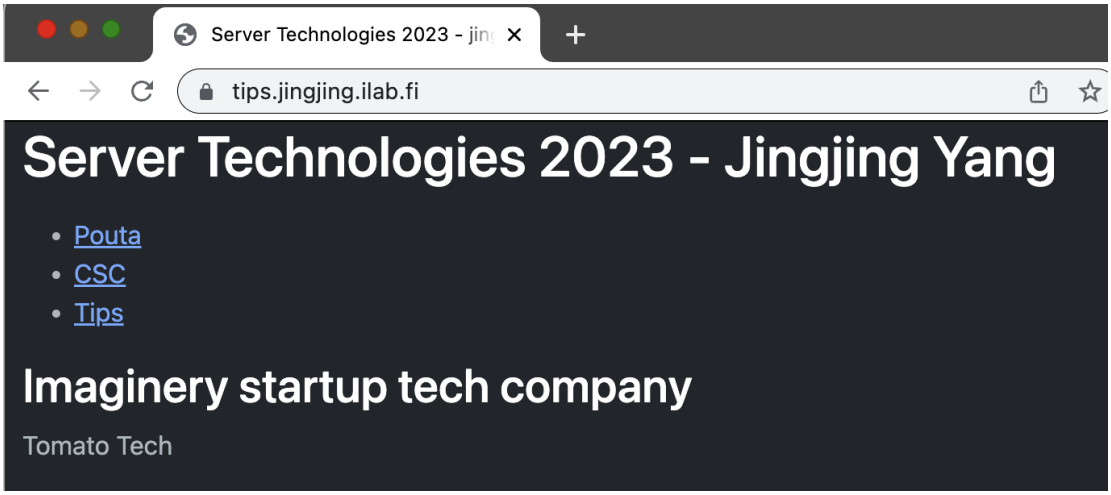
The main log file for apache is `/var/log/apache2/access.log` for access logs and `/var/log/apache2/error.log` for error logs.

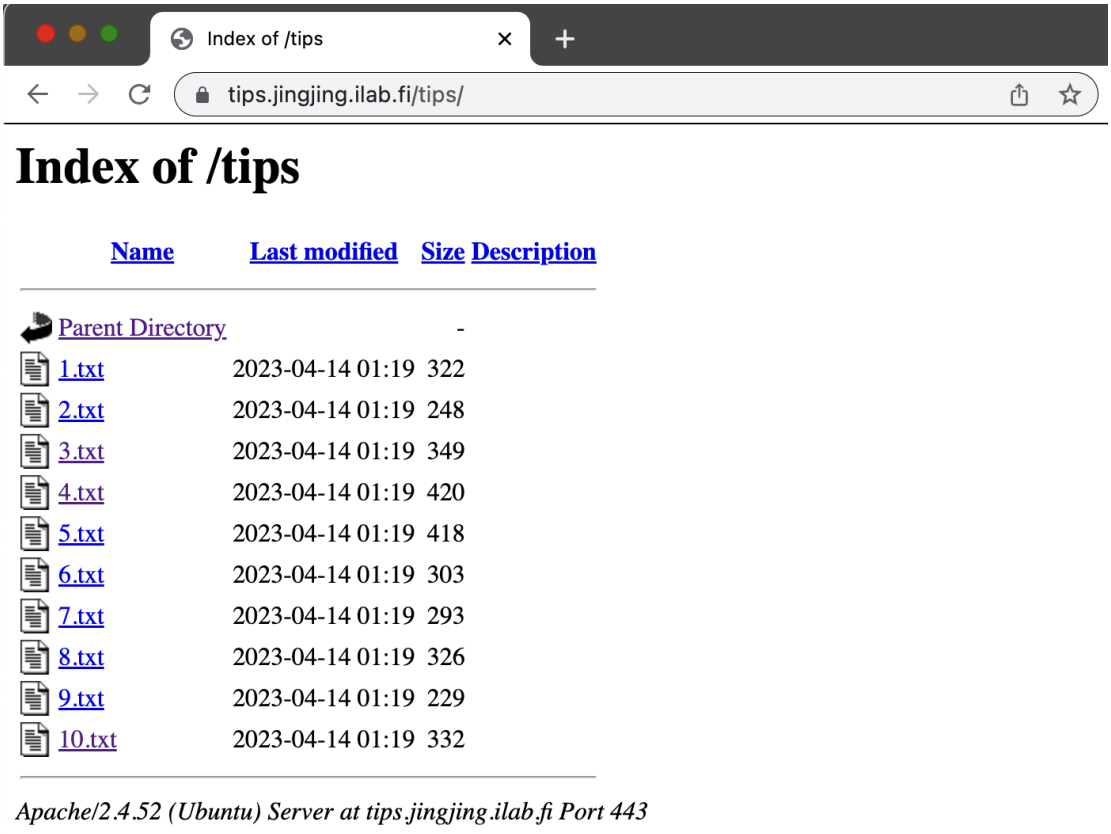
To diagnose any issues, you can check the error log to see if any errors are reported and then adjust your configuration accordingly.

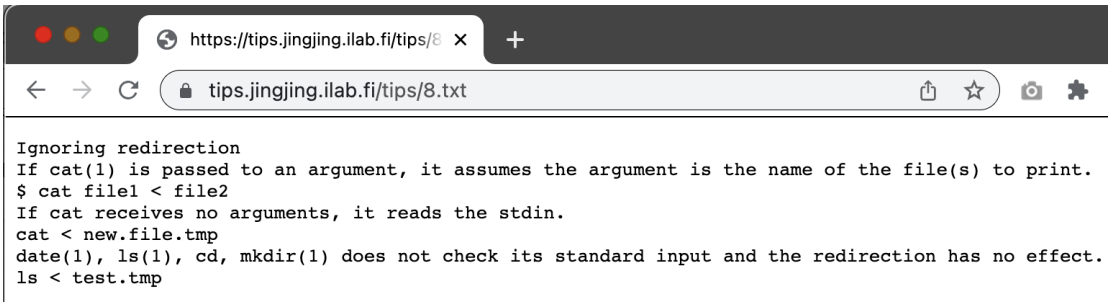
```
root@jingjing-serverb-2023:~# ls /var/log/apache2/ | grep .log$
access.log
error.log
other_vhosts_access.log
tips.jingjing.access.log
tips.jingjing.error.log
```

👉 ErrorLog ${APACHE_LOG_DIR}/tips.jingjing.error.log
CustomLog ${APACHE_LOG_DIR}/tips.jingjing.access.log combined

Testing







▼ 🍃 Let’s Encrypt & Certbot

Let's Encrypt is a Certificate Authority that provides free SSL/TLS certificates.
Certbot is an open-source tool that automates the process of obtaining and renewing these certificates. It can work together with the Apache plugin for Python 3 to secure websites running on Apache web servers.

Installation

```
$ sudo apt-get install certbot python3-certbot-apache
```

Start / stop

To start the Certbot service: `$ sudo systemctl start certbot.timer`

To enable the Certbot service to automatically start at boot time: `$ sudo systemctl enable certbot.timer`

To stop the Certbot service permanently, you should first disable the timer associated with the Certbot service using `$ sudo systemctl disable certbot.timer`, so that it cannot be automatically activated. Then you can stop the Certbot service using `$ sudo systemctl stop certbot.service`.

Configuration

- Installing Certbot `$ sudo apt install certbot python3-certbot-apache`
- Checking your Apache Virtual Host Configuration
- Allowing HTTPS Through the Firewall
- Obtaining an SSL Certificate `$ sudo certbot --apache`

```
jingjing@jingjing-serverb-2023:~$ sudo certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
 (Enter 'c' to cancel): jingjing.yang@tuni.fi

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: Y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: N
Account registered.

Which names would you like to activate HTTPS for?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: jingjing.ilab.fi
2: tips.jingjing.ilab.fi
3: www.jingjing.ilab.fi
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
Requesting a certificate for jingjing.ilab.fi and 2 more domains

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/jingjing.ilab.fi/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/jingjing.ilab.fi/privkey.pem
This certificate expires on 2023-07-15.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for jingjing.ilab.fi to /etc/apache2/sites-available/tips.jingjing.ilab.fi-le-ssl.conf
Successfully deployed certificate for tips.jingjing.ilab.fi to /etc/apache2/sites-available/tips.jingjing.ilab.fi-le-ssl.conf
Successfully deployed certificate for www.jingjing.ilab.fi to /etc/apache2/sites-available/tips.jingjing.ilab.fi-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://jingjing.ilab.fi, https://tips.jingjing.ilab.fi, and https://www.jingjing.ilab.fi

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
jingjing@jingjing-serverb-2023:~$ egrep -v "(#|^$)" /etc/apache2/sites-available/tips.jingjing.ilab.fi
-le-ssl.conf
<IfModule mod_ssl.c>
<VirtualHost *:443>
  ServerName tips.jingjing.ilab.fi
  ServerAlias www.jingjing.ilab.fi jingjing.ilab.fi
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/tips.jingjing.ilab.fi
  ErrorLog ${APACHE_LOG_DIR}/tips.jingjing.error.log
  CustomLog ${APACHE_LOG_DIR}/tips.jingjing.access.log combined
  ProxyPass "/totd"  "http://localhost:3000/totd"
        ProxyPassReverse "/totd"  "http://localhost:3000/totd"
  ProxyPass "/dbtips"  "http://localhost:3432/plain"
        ProxyPassReverse "/dbtips"  "http://localhost:3432/plain"
Include /etc/letsencrypt/options-ssl-apache.conf
SSLCertificateFile /etc/letsencrypt/live/jingjing.ilab.fi/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/jingjing.ilab.fi/privkey.pem
</VirtualHost>
</IfModule>
```

- Verifying Certbot Auto-Renewal `$ sudo systemctl status certbot.timer`

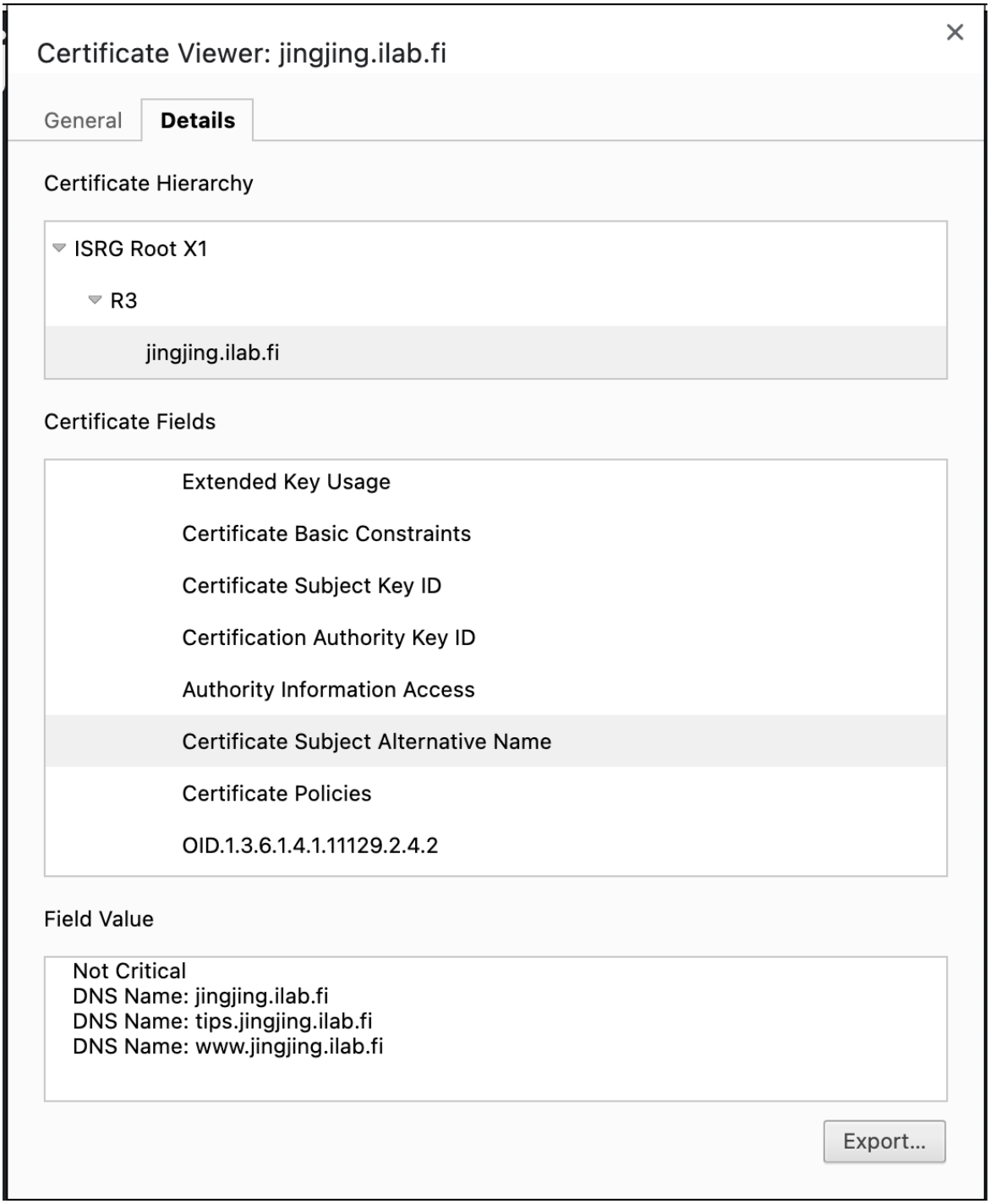
```
jingjing@jingjing-serverb-2023:~$ sudo systemctl status certbot.timer
● certbot.timer - Run certbot twice daily
     Loaded: loaded (/lib/systemd/system/certbot.timer; enabled; vendor preset: enabled)
     Active: active (waiting) since Tue 2023-05-02 22:35:21 UTC; 4s ago
    Trigger: Wed 2023-05-03 00:40:07 UTC; 2h 4min left
   Triggers: ● certbot.service

May 02 22:35:21 jingjing-serverb-2023 systemd[1]: Started Run certbot twice daily.

jingjing@jingjing-serverb-2023:~$ sudo systemctl stop certbot.service
Warning: Stopping certbot.service, but it can still be activated by:
  certbot.timer
```

Testing

```
jingjing@jingjing-serverb-2023:~$ curl http://tips.jingjing.ilab.fi
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://tips.jingjing.ilab.fi/">here</a>.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at tips.jingjing.ilab.fi Port 80</address>
</body></html>
```



▼ PM2

PM2 is a process manager for Node.js applications that allows you to easily manage and deploy Node.js applications in production environments. It provides features such as process monitoring, logging, and automatic restarts in case of crashes or errors.
PM2 can be used to manage Node.js applications that are running on a server or in a containerized environment.

Installation

```
$ sudo apt-get install pm2
```

Start / stop

Starting pm2 service: `$ pm2 start <app.js>`

Stopping pm2 service: `$ pm2 stop <app.js>`

👉 `/home/jingjing/tla-express-tips/nodeapp/app.js`

Configuration

- `$ pm2 startup` This will generate a command that you need to run as root.
  After running the command, you should see a message indicating that PM2 has been installed as a service.

```
jingjing@jingjing-serverb-2023:~$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u jingjing --hp /home/
jingjin
```

```
jingjing@jingjing-serverb-2023:~$ sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 start
up systemd -u jingjing --hp /home/jingjing
[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target

[Service]
Type=forking
User=jingjing
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/ga
mes:/snap/bin:/usr/bin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
Environment=PM2_HOME=/home/jingjing/.pm2
PIDFile=/home/jingjing/.pm2/pm2.pid
Restart=on-failure

ExecStart=/usr/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/usr/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/usr/lib/node_modules/pm2/bin/pm2 kill

[Install]
WantedBy=multi-user.target

Target path
/etc/systemd/system/pm2-jingjing.service
Command list
[ 'systemctl enable pm2-jingjing' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-jingjing.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-jingjing...
[PM2] [v] Command successfully executed.
+---------------------------------------+
[PM2] Freeze a process list on reboot via:
$ pm2 save
```

```
[PM2] Remove init script via:
$ pm2 unstartup system
```

- `$ pm2 start app.js` This will start the application with PM2.
  If everything is working correctly, you should see your application listed as `online`.

```
jingjing@jingjing-serverb-2023:~$ pm2 start tla-express-tips/nodeapp/app.js
[PM2] Applying action restartProcessId on app [app](ids: [ 0 ])
[PM2] [app](0) ✓
[PM2] Process successfully started
┌────┬────────────────────┬──────────┬────┬───────────┬──────────┬──────────┐
│ id │ name               │ mode     │ ↺  │ status    │ cpu      │ memory   │
├────┼────────────────────┼──────────┼────┼───────────┼──────────┼──────────┤
│ 0  │ app                │ fork     │ 0  │ online    │ 0%       │ 12.4mb   │
└────┴────────────────────┴──────────┴────┴───────────┴──────────┴──────────┘
```

- `$ pm2 save` This save the current state of the application and will ensure that your application is automatically started when the server is rebooted.

```
jingjing@jingjing-serverb-2023:~$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/jingjing/.pm2/dump.pm2
```

Logs and Diagnosis

To display all logs for all PM2-managed processes: `$ pm2 logs`

👉 Tailing last 15 lines for [all] processes (change the value with --lines option)

`/home/jingjing/.pm2/pm2.log`

`/home/jingjing/.pm2/logs/app-error.log`

`/home/jingjing/.pm2/logs/app-out.log`

```
jingjing@jingjing-serverb-2023:~$ ls .pm2
dump.pm2  module_conf.json  pids     pm2.pid   rpc.sock
logs      modules           pm2.log  pub.sock  touch

jingjing@jingjing-serverb-2023:~$ ls .pm2/logs
app-error.log  app-out.log
```

To display logs for a specific process: `$ pm2 logs <process_name>`

👉 The process name is the name of the application or script that you are running with PM2. For example, if you are running an application called `app.js` with PM2, the process name will be `app`.

Testing

```
jingjing@jingjing-serverb-2023:~$ while true;do date;ls;sleep 1;done
Tue May  2 00:10:31 UTC 2023
express-tips-version-2  nodesource_setup.sh  tla-express-tips
Tue May  2 00:10:32 UTC 2023
express-tips-version-2  nodesource_setup.sh  tla-express-tips
Tue May  2 00:10:33 UTC 2023
express-tips-version-2  nodesource_setup.sh  tla-express-tips
^C
```

▼ 🍃 NFS (Network File System)

NFS is a distributed file system that allows remote file access over a network. It is commonly used in environments where multiple computers need to access the same files, such as in a cluster or cloud computing environment.

NFS works by mounting a remote directory on a local computer, allowing users to access files on the remote machine as if they were stored locally.

Installation

on NFS host server(jingjing-nfs-server): `$ sudo apt-get install nfs-kernel-server`

on NFS client server(jingjing-serverb-2023): `$ sudo apt-get install nfs-common`

Start / stop

Starting NFS kernel service: `$ sudo systemctl start nfs-kernel-server`

Enable NFS kernel service: `$ sudo systemctl enable nfs-kernel-server`

Stopping NFS kernel service: `$ sudo systemctl stop nfs-kernel-server`

remote directory mounted

`/nfs/tips`

Configuration

- Create a directory on the server that you want to share over NFS.
- Edit the /etc/exports file on the server to specify which directories to share and which clients are allowed to access them.

```
jingjing@jingjing-nfs-server:~$ egrep -v "#" /etc/exports
/var/nfs/tips    192.168.1.0/24(rw,sync,no_subtree_check)
```

- Export the shared directory by running the `exportfs` command on the server.

```
jingjing@jingjing-nfs-server:~$ sudo exportfs
/var/nfs/tips   192.168.1.0/24
```

- On the client machine, create a mount point directory where the shared directory will be mounted. If the directory exists, make sure that it is empty.

- Mount the shared directory on the client machine using the "mount" command.

```
jingjing@jingjing-serverb-2023:~$ sudo mount -v jingjing-nfs-server:/var/nfs/tips /nfs/tips
mount.nfs: timeout set for Mon May  1 23:14:58 2023
mount.nfs: trying text-based options 'vers=4.2,addr=192.168.1.13,clientaddr=192.168.1.18'

jingjing@jingjing-serverb-2023:~$ df -h | grep tips
jingjing-nfs-server:/var/nfs/tips   78G  2.1G   76G   3% /nfs/ti
```

- Configure file `/etc/fstab` to make the mounting permanent.

```
jingjing@jingjing-serverb-2023:~$ cat /etc/fstab | grep tips
jingjing-nfs-server:/var/nfs/tips  /nfs/tips  nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

Logs and Diagnosis

The `/var/log/syslog` file contains system logs, including NFS logs.
You can view the file by running the command: `$ sudo tail -f /var/log/syslog`. This will display the last 10 lines of the file, and will continue to display new lines as they are added to the file.
If you want to see more lines, you can change the number after `tail` to the desired number of lines. For example, to see the last 20 lines, run the command: `$ sudo tail -n 20 -f /var/log/syslog`. This will display the last 20 lines of the file and continue to display new lines as they are added to the file.

Another way to view NFS logs is to use the `$ nfsstat` command. The `$ nfsstat` command displays statistics about the NFS client and server. To view the NFS logs, you can run the command: `$ sudo nfsstat -c`. This command will display client-side statistics for NFS operations, including the number of read, write, and access operations performed. You can also use the `-s` option to view server-side statistics.

Another command you could use is `$ sudo tcpdump -i any port nfs`. This command will capture all NFS traffic on the network interface and display it in real-time. This can be useful if you need to troubleshoot NFS performance issues or diagnose problems with NFS mounts.

Unmounting an NFS Remote Share

`client: $ sudo umount /nfs/tips`

Testing

```
jingjing@jingjing-nfs-server:~$ hostname > /nfs/tips/test-jingjing.txt
jingjing@jingjing-nfs-server:~$ cat /nfs/tips/test-jingjing.txt
jingjing-nfs-server

jingjing@jingjing-nfs-server:~$ cd /nfs/tips/
jingjing@jingjing-nfs-server:/nfs/tips$ cd /var/nfs/tips/
jingjing@jingjing-nfs-server:/var/nfs/tips$
```

CMDB

Configuration management database In the local system

`/data/cmdb-server-tech-2023-jingjing-yang/` on main server

```
jingjing@jingjing-serverb-2023:~$ tree /data/cmdb-server-tech-2023-jingjing-yang/
/data/cmdb-server-tech-2023-jingjing-yang/
├── README.md
├── docs
│   ├── jingjing-nfs-server
│   │   └── apt_installed.txt
│   └── jingjing-serverb-2023
│       └── apt_installed.txt
├── jingjing-nfs-server
│   ├── etc
│   │   ├── exports
│   │   ├── sudoers.d
│   │   │   └── 91-stec-nfs
│   │   └── ufw
│   │       ├── ufw.conf
│   │       ├── user.rules
│   │       └── user6.rules
│   └── var
│       └── spool
│           └── cron
│               └── crontabs
│                   └── root
└── jingjing-serverb-2023
    ├── etc
    │   ├── apache2
    │   │   ├── apache2.conf
    │   │   └── sites-available
    │   │       ├── 000-default.conf
    │   │       ├── default-ssl.conf
    │   │       └── tips.jingjing.ilab.fi.conf
    │   ├── apt
    │   │   └── apt.conf.d
    │   │       └── 50unattended-upgrades
    │   ├── bash.bashrc
    │   ├── fstab
    │   ├── hosts
    │   ├── inputrc
    │   ├── letsencrypt
    │   │   ├── cli.ini
    │   │   ├── options-ssl-apache.conf
    │   │   ├── renewal
    │   │   │   └── jingjing.ilab.fi.conf
    │   │   └── renewal-hooks
    │   │       ├── deploy
    │   │       ├── post
    │   │       └── pre
    │   ├── passwd
    │   ├── pm2-jingjing.service
    │   ├── ssh
    │   │   └── sshd_config
    │   ├── sudoers.d
    │   │   ├── 90-cloud-init-users
    │   │   ├── 91-stec22-users
    │   │   └── README
    │   ├── systemd
    │   │   └── system
    │   │       └── timers.target.wants
    │   │           └── certbot.timer
    │   └── ufw
    │       ├── ufw.conf
    │       ├── user.rules
    │       └── user6.rules
    ├── lib
    │   └── systemd
    │       └── system
    │           └── certbot.timer
    ├── opt
```

```
    │   └── totd
    │       ├── README.md
    │       ├── linuxtips.sh
    │       └── tips
    │           ├── 1.txt
    │           ├── 10.txt
    │           ├── 2.txt
    │           ├── 3.txt
    │           ├── 4.txt
    │           ├── 5.txt
    │           ├── 6.txt
    │           ├── 7.txt
    │           ├── 8.txt
    │           └── 9.txt
    └── var
        ├── spool
        │   └── cron
        │       └── crontabs
        │           └── root
        └── www
            ├── html
            │   └── index.html
            └── tips.jingjing.ilab.fi
                ├── files -> /opt/stec/tip-of-the-day-1.0-bash-script/tips
                ├── index.html
                ├── tips -> /opt/stec/tip-of-the-day-1.0-bash-script/tips
                └── txt -> /opt/stec/tip-of-the-day-1.0-bash-script/tips

45 directories, 47 files
```

`/home/jingjing/cmdb-server-tech-2023-jingjing-yang/` one nfs server

```
jingjing@jingjing-nfs-server:~$ tree cmdb-server-tech-2023-jingjing-yang/
cmdb-server-tech-2023-jingjing-yang/
├── README.md
├── docs
│   ├── jingjing-nfs-server
│   │   └── apt_installed.txt
│   └── jingjing-serverb-2023
│       └── apt_installed.txt
├── jingjing-nfs-server
│   ├── etc
│   │   ├── exports
│   │   ├── sudoers.d
│   │   │   └── 91-stec-nfs
│   │   └── ufw
│   │       ├── ufw.conf
│   │       ├── user.rules
│   │       └── user6.rules
│   └── var
│       └── spool
│           └── cron
│               └── crontabs
│                   └── root
└── jingjing-serverb-2023
    ├── etc
    │   ├── apache2
    │   │   ├── apache2.conf
    │   │   └── sites-available
    │   │       ├── 000-default.conf
    │   │       ├── default-ssl.conf
    │   │       └── tips.jingjing.ilab.fi.conf
    │   ├── apt
    │   │   └── apt.conf.d
    │   │       └── 50unattended-upgrades
    │   ├── bash.bashrc
    │   ├── fstab
    │   ├── hosts
```

```
    │   ├── inputrc
    │   ├── letsencrypt
    │   │   ├── cli.ini
    │   │   ├── options-ssl-apache.conf
    │   │   └── renewal
    │   │       └── jingjing.ilab.fi.conf
    │   ├── passwd
    │   ├── pm2-jingjing.service
    │   ├── ssh
    │   │   └── sshd_config
    │   ├── sudoers.d
    │   │   ├── 90-cloud-init-users
    │   │   ├── 91-stec22-users
    │   │   └── README
    │   ├── systemd
    │   │   └── system
    │   │       └── timers.target.wants
    │   │           └── certbot.timer
    │   └── ufw
    │       ├── ufw.conf
    │       ├── user.rules
    │       └── user6.rules
    ├── lib
    │   └── systemd
    │       └── system
    │           └── certbot.timer
    ├── opt
    │   └── totd
    │       ├── README.md
    │       ├── linuxtips.sh
    │       └── tips
    │           ├── 1.txt
    │           ├── 10.txt
    │           ├── 2.txt
    │           ├── 3.txt
    │           ├── 4.txt
    │           ├── 5.txt
    │           ├── 6.txt
    │           ├── 7.txt
    │           ├── 8.txt
    │           └── 9.txt
    └── var
        ├── spool
        │   └── cron
        │       └── crontabs
        │           └── root
        └── www
            ├── html
            │   └── index.html
            └── tips.jingjing.ilab.fi
                ├── files -> /opt/stec/tip-of-the-day-1.0-bash-script/tips
                ├── index.html
                ├── tips -> /opt/stec/tip-of-the-day-1.0-bash-script/tips
                └── txt -> /opt/stec/tip-of-the-day-1.0-bash-script/tips

38 directories, 50 files
```

Git repo URL for server configuration files

📎 https://gitlab.tamk.cloud/server-tech-2023b-jingjing-yang/cmdb-server-tech-2023-jingjing-yang

▼ 🧩 Quality and operation requirements

Enable automatic updates

All operating systems have the ability to apply updates automatically, and it is easy to turn this on.

No Mail Servers

It is very easy to configure a mail server so that it can be used by spammers. So, please instead use an existing SMTP server outside the cloud.

Upgrade your kernel

Some updates, such as kernel upgrades, require rebooting the virtual machines. Please schedule this into your regular maintenance.

Subscribe to security announcements for your OS

If there is a security problem in your operating system, you need to find it out as soon as possible. Find the appropriate mailing list and keep an eye out for anything that requires urgent action.

Run a restrictive firewall

Your instances should be configured so that they allow the minimum access required to run the service. Please use a host-based firewall, in conjunction with the cloud-provided firewall to manage access.

Disable/remove unneeded accounts

Keep an eye on the user accounts enabled in your system. Some applications create default accounts which are insecure. An ideal scenario might be three accounts: root (with ssh disabled), a user account for a sysadmin (key login only) and a user-level account for a service (login disabled).

Disable password login – use keys

Passwords can be, with enough time and compute power, attacked with brute force. The average SSH server deals with thousands of such attacks every week, so use keys to have one less worry.

Do not store keys on the image

The cloud provides a metadata service so that you can download keys on boot. This is recommended. It ensures that if your key is compromised, not all running instances of that image are compromised.

Use tools like denyhosts

Tools such as *denyhosts*, which look at log files for attempted breaches and then firewall out IP addresses, can take your security approach to a more active level.

Disable unneeded services

Know what services run on your image and disable the unnecessary ones before you upload it. This reduces the attack surface.

Use encrypted communications

Wherever possible, use encrypted communications to avoid attacks which intercept data.

Use the best practices for logging

Make sure that the services are logging to a secure location, that is as tamper-proof as possible. Keep the logs for a reasonably long. Consider logging to a remote server as well.