Maintainer’s guide – IMS Project

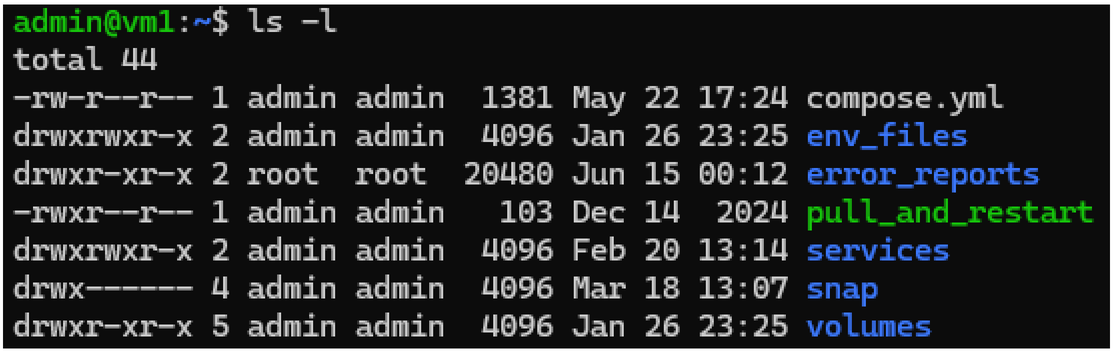
General information

- The project’s source code can be found in the following github repository: https://github.com/yuval-roth/ims_project
- The project’s domain is ims-project.cs.bgu.ac.il
- The SSL certificates used for encrypted communication are based on Let’s Encrypt
- The ‘watch’ and ‘game_server’ sub-projects depend on the ‘common’ sub-project as a library that holds common code used in the Kotlin projects
- Uncaught exceptions in the game server and the watch are logged and saved as error reports to a folder in the server

Server technical information

- The project’s services are ran on a server hosted in BGU’s cloud infrastructure on the Red DMZ network, managed by the department of computer science
- The server runs Ubuntu 22.04.5 LTS
- The server’s IP is 132.72.116.91
- The server can be accessed via SSH after connecting to BGU’s VPN
- The server has 8 open ports: 80, 443, 8640, 8641, 8642, 8643, 8644, 8645
- The server has 8 CPUs, 16GB ram and 300GB storage

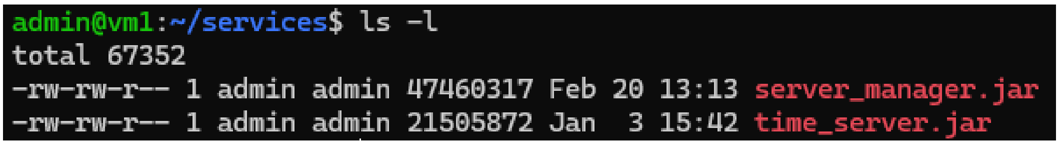
The app’s data is in the admin’s home folder on the ubuntu server



```
admin@vm1:~$ ls -l
total 44
-rw-r--r-- 1 admin admin  1381 May 22 17:24 compose.yml
drwxrwxr-x 2 admin admin  4096 Jan 26 23:25 env_files
drwxr-xr-x 2 root  root  20480 Jun 15 00:12 error_reports
-rwxr--r-- 1 admin admin   103 Dec 14  2024 pull_and_restart
drwxrwxr-x 2 admin admin  4096 Feb 20 13:14 services
drwx------ 4 admin admin  4096 Mar 18 13:07 snap
drwxr-xr-x 5 admin admin  4096 Jan 26 23:25 volumes
```

env_files: contains env files to be used in compose.yml
error_reports: contains the generated error reports for uncaught exceptions
services: contains binaries for the systemd services we use
volumes: docker containers’ volumes, as specified in compose.yml
pull_and_restart: script for pulling changes from DockerHub

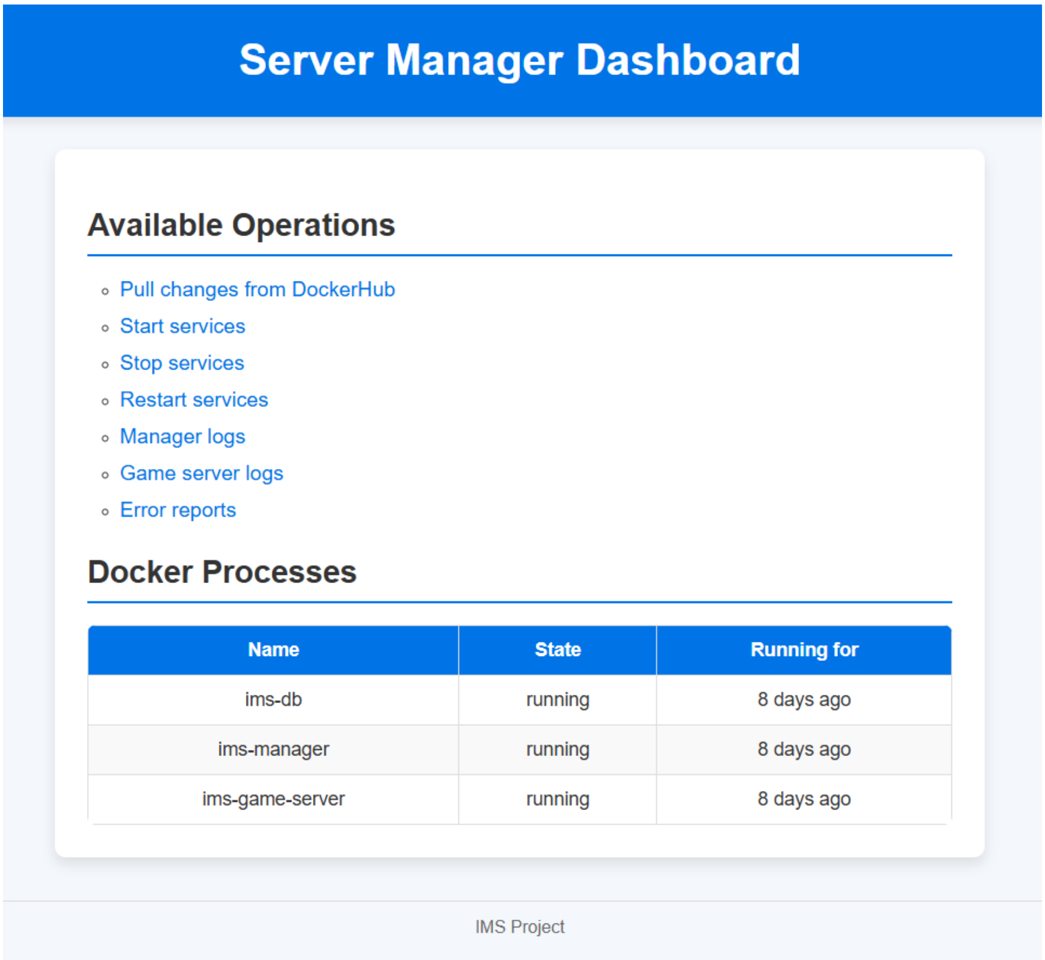
Systemd services



```
admin@vm1:~/services$ ls -l
total 67352
-rw-rw-r-- 1 admin admin 47460317 Feb 20 13:13 server_manager.jar
-rw-rw-r-- 1 admin admin 21505872 Jan  3 15:42 time_server.jar
```

Server manager (For development purposes)

- Running on port 8645/tcp
- Web interface to manage our services and allow control over the docker service. Also allows to see the manager and game server logs and view error reports.
- After 3 failed login attempts, your IP gets blocked for 30 minutes



Server Manager Dashboard

Available Operations

- Pull changes from DockerHub
- Start services
- Stop services
- Restart services
- Manager logs
- Game server logs
- Error reports

Docker Processes

Name	State	Running for
ims-db	running	8 days ago
ims-manager	running	8 days ago
ims-game-server	running	8 days ago

IMS Project

Time server

- Running on port 8641/udp
- Responds to basic “what is the time” requests
- Runs as a system service and not in docker because it needs a reliable and precise system clock to work

Nginx

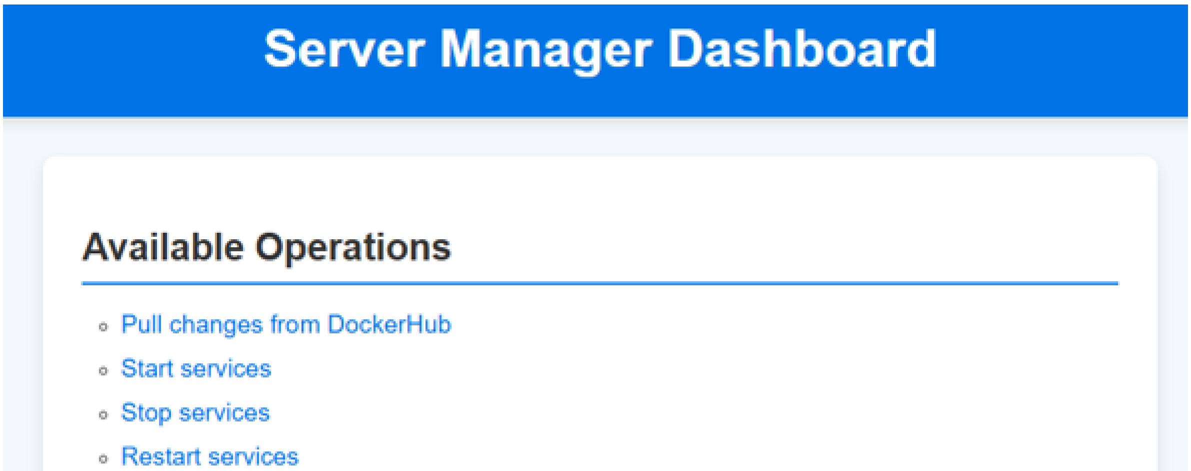
- Used as reverse proxy for SSL termination

Deployment

Server side

The Manager Console, Game Server and PostgreSQL services are deployed to the server using docker compose and DockerHub.

We upload the docker images to DockerHub and then we pull the images and deploy them to the server.



Server Manager Dashboard

Available Operations

- Pull changes from DockerHub
- Start services
- Stop services
- Restart services

We created an automation for deploying a new version to the server. When clicking on “Pull changes from DockerHub” in the server manager dashboard, it spawns a process that calls the “**pull_and_restart”** script that shuts down the containers, pulls the new images from DockerHub and then starts the containers up again

The docker services can also be started, stopped and restarted with a single click of a button.

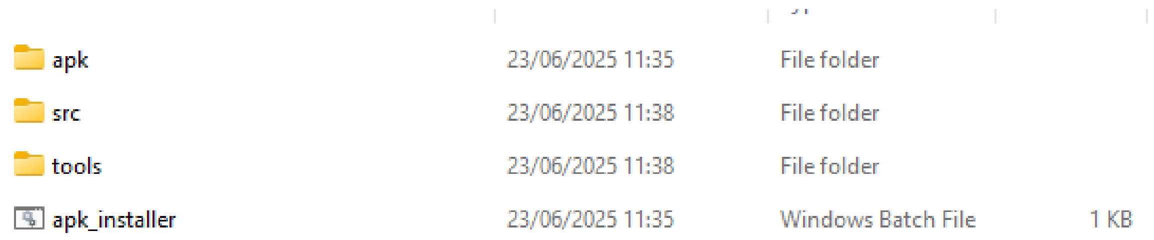
Watch side

Before installing the app on the watch you will need to perform the following steps:

1) Enable developer mode in the android settings
2) Enable developer mode for samsung’s health platform (Health Platform’s Developer Mode | Samsung Developer)
3) Disable touch bezel (Settings, tap General > Touch bezel, and disable the feature)

After you’ve completed these steps, you can continue to the instructions of installing the app in the next page

There is a folder in the repo called “apk_installer_bundle” that contains the following files:

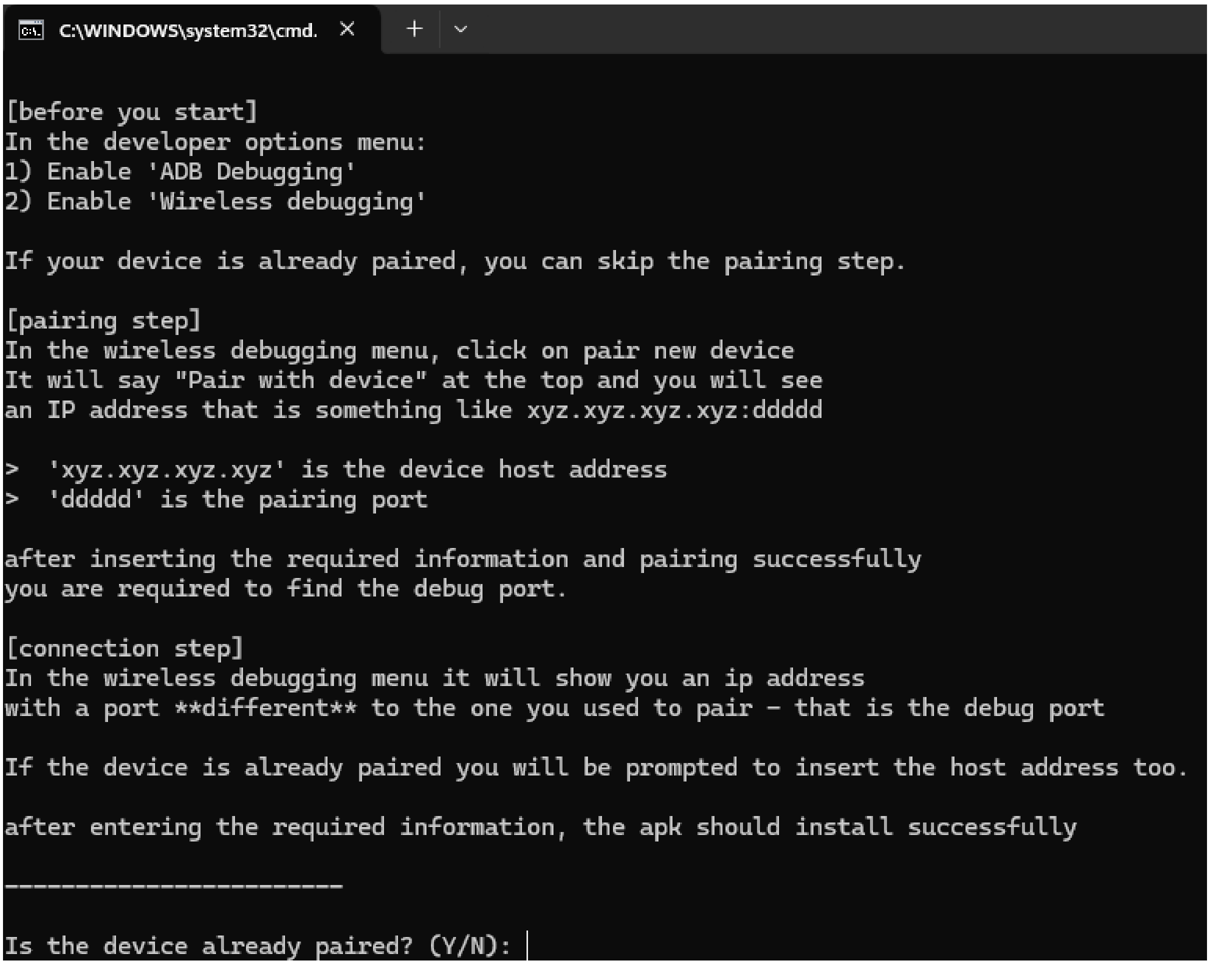


apk	23/06/2025 11:35	File folder	
src	23/06/2025 11:38	File folder	
tools	23/06/2025 11:38	File folder	
apk_installer	23/06/2025 11:35	Windows Batch File	1 KB

src: contains the source code for the installer bundle
tools: contains android dev tools (ADB) needed for installing

The android apk file should be placed inside the “apk” folder and **must be named** “ims.apk”

Upon running the “apk_installer.bat” the installation will commence, and you will need to follow the instructions on the screen



```
[before you start]
In the developer options menu:
1) Enable 'ADB Debugging'
2) Enable 'Wireless debugging'

If your device is already paired, you can skip the pairing step.

[pairing step]
In the wireless debugging menu, click on pair new device
It will say "Pair with device" at the top and you will see
an IP address that is something like xyz.xyz.xyz.xyz:ddddd

>  'xyz.xyz.xyz.xyz' is the device host address
>  'ddddd' is the pairing port

after inserting the required information and pairing successfully
you are required to find the debug port.

[connection step]
In the wireless debugging menu it will show you an ip address
with a port **different** to the one you used to pair - that is the debug port

If the device is already paired you will be prompted to insert the host address too.

after entering the required information, the apk should install successfully

------------------------

Is the device already paired? (Y/N):
```

After installing the app for the first time **you need to open it, accept the body sensor permission, and then close the app completely and start it back again for the sensors to work** due to an issue with samsung’s health platform