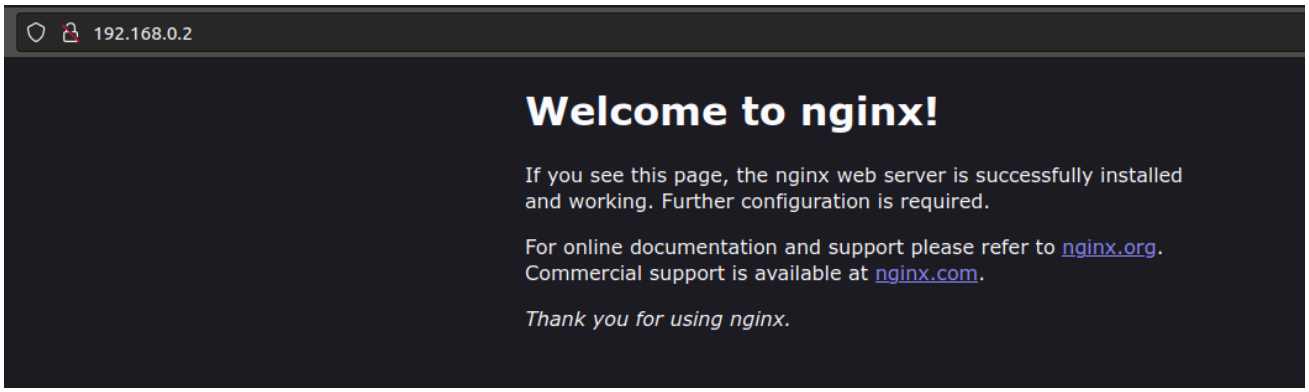
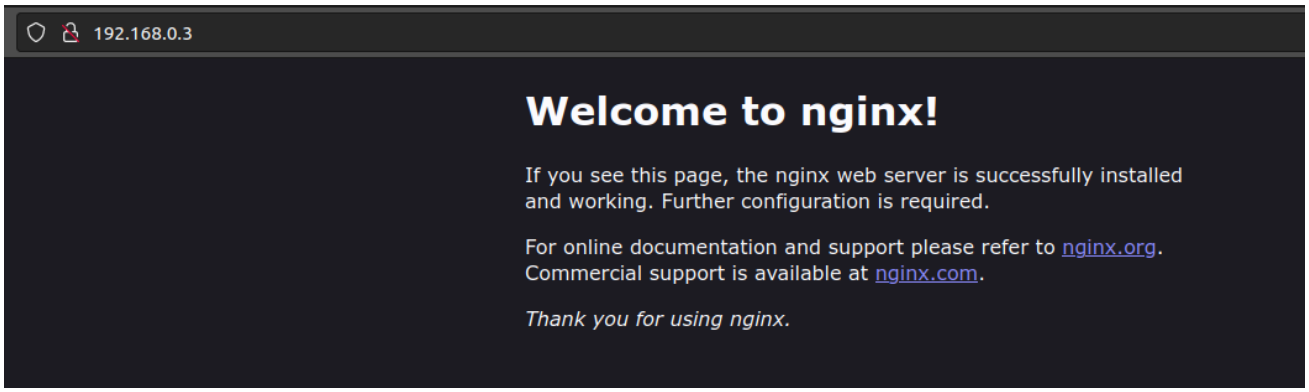
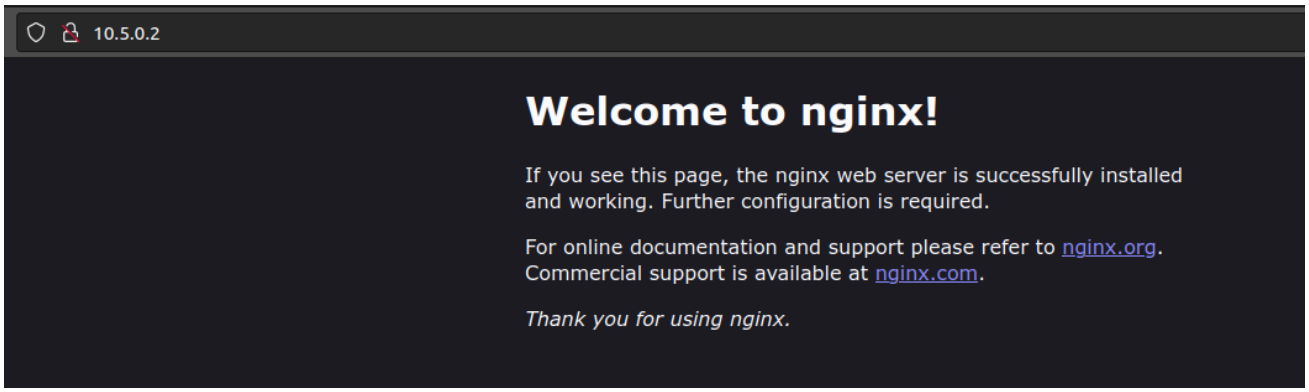
Problem 1:

Create a bridge network with subnet 192.168.0.0/24.
Run 2 containers and attach containers to this network.
Create another bridge network with subnet 10.5.0.0/24.
Run any container and attach it to the new network.
Make sure that the containers at different network can't ping each other

```
heba@heba-HP-ProBook-450-G4:~$ docker network create --subnet=192.168.0.0/24 --internal my-bridge-network
9921b849e9cc4f8e9fce4951f119a875e192094ad5c23c4bc2084d149124bff3
heba@heba-HP-ProBook-450-G4:~$ docker network create --subnet=10.5.0.0/24 --internal my-second-bridge-network
007319398beae2d6b01b3dba32447496905486083be9468eb4d638aec70584d8
heba@heba-HP-ProBook-450-G4:~$ docker run -d --name container1 --network=my-bridge-network nginx
f0cae3092b6db16c9abb333002dc42c859c04c1d6fb7801bf6253b074fdfb8c6
heba@heba-HP-ProBook-450-G4:~$ docker run -d --name container2 --network=my-bridge-network nginx
f07b155ed6b2e803af288e148c3b5533541d9b1c90eeae904e44ba654a50ed5d
heba@heba-HP-ProBook-450-G4:~$ docker run -d --name container3 --network=my-second-bridge-network nginx
68dfd7a1b664359085e570b82d0a0dcafe530ee3f49f2a056c43062359900e29
```







Problem 2:

Create static html file

Write Dockerfile to build image based on httpd to host the html file and specify the following

Copy the html file.

Copy a new configuration file to listen on port 9999 instead of 80

Open the port 9999 in the container

Add environment variable CONTAINER with value docker .

Add startup command to echo the variable

Docker file

Terminal

```
FROM httpd:latest

# Copy the static HTML file to the container
COPY index.html /usr/local/apache2/htdocs/

# Copy the new configuration file to listen on port 9999
COPY httpd.conf /usr/local/apache2/conf/httpd.conf

# Open port 9999 in the container
EXPOSE 9999

# Set environment variable CONTAINER
ENV CONTAINER docker

# Add startup command to echo the variable
CMD ["sh", "-c", "echo The container is running on $CONTAINER && httpd-foreground"]
~
```

HTML file

```
<!DOCTYPE html>
<html>
<head>
        <title>Hello, World!</title>
        <style>
                body {
                        background-color: #2E86C1;
                        background-image: linear-gradient(to bottom, #2E86C1, #4DB6AC);
                        color: #FFFFFF;
                        font-family: Arial, sans-serif;
                        display: flex;
                        align-items: center;
                        justify-content: center;
                        height: 100vh;
                }
                h1 {
                        font-size: 96px;
                        text-shadow: 2px 2px #000000;
                        animation: slide-in 1s ease-out;
                }
                @keyframes slide-in {
                        from {
                                transform: translateX(-100%);
                        }
                        to {
                                transform: translateX(0);
                        }
                }
        </style>
</head>
<body>
        <h1>Hello, World!</h1>
</body>
</html>
~
```

Configuration file

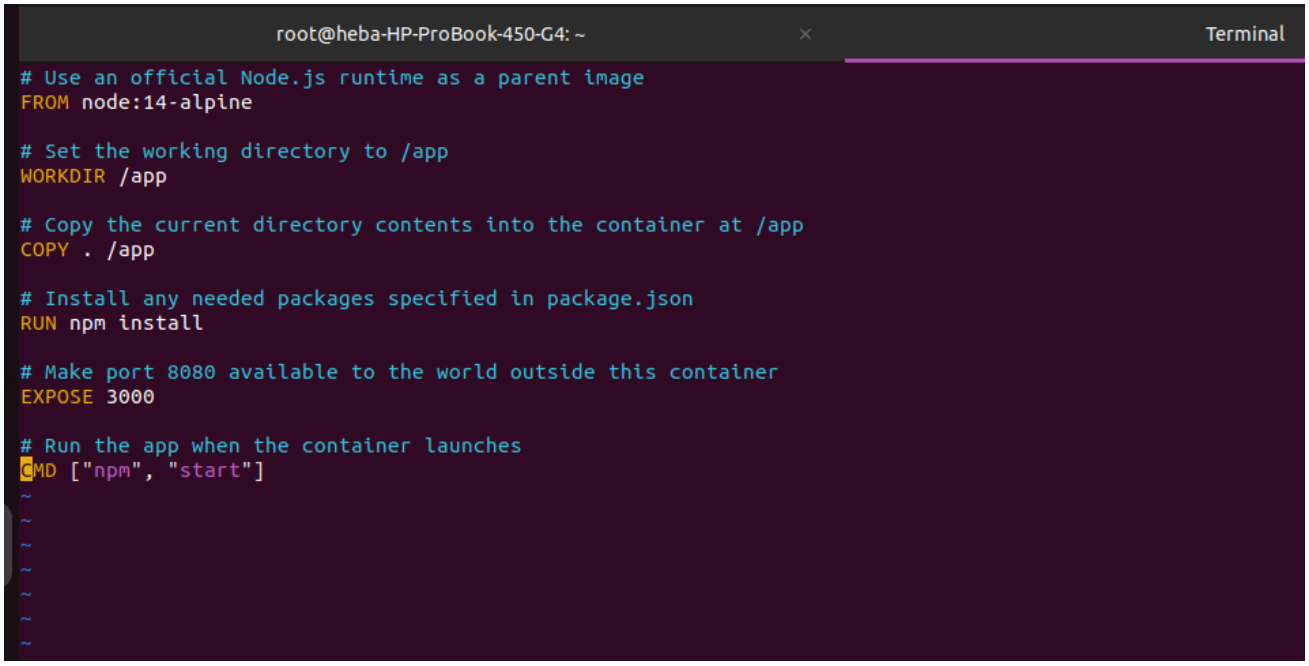
```
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used.  If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/usr/local/apache2"

#
# Mutex: Allows you to set the mutex mechanism and mutex file directory
# for individual mutexes, or change the global defaults
#
# Uncomment and change the directory if mutexes are file-based and the default
# mutex file directory is not on a local disk or is not appropriate for some
# other reason.
#
# Mutex default:logs

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:9999
Listen 9999
```

```
heba@heba-HP-ProBook-450-G4:~/dockerplay2$ vim Dockerfile
heba@heba-HP-ProBook-450-G4:~/dockerplay2$ docker build -t my-web-server .
[+] Building 5.3s (8/8) FINISHED
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 489B
 => [internal] load metadata for docker.io/library/httpd:latest
 => [1/3] FROM docker.io/library/httpd:latest@sha256:1bb3f7669a85713906e695599d29c58ab40d4e6409907946609d92a428e95b49
 => [internal] load build context
 => => transferring context: 20.90kB
 => CACHED [2/3] COPY index.html /usr/local/apache2/htdocs/
 => [3/3] COPY httpd.conf /usr/local/apache2/conf/httpd.conf
 => exporting to image
 => => exporting layers
 => => writing image sha256:e082c16f74acb56388ae9ec613440da907273841011ed94806c3d0091296daf4
 => => naming to docker.io/library/my-web-server
heba@heba-HP-ProBook-450-G4:~/dockerplay2$ docker run -d -p 9999:9999 my-web-server
cdaf2d04c5433dc2d1dc9afddc403580e7f89aa7301df09b2172cd0a1743ee27
heba@heba-HP-ProBook-450-G4:~/dockerplay2$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                                                  NAMES
cdaf2d04c543   my-web-server  "sh -c 'echo The con…"   8 seconds ago    Up 6 seconds    80/tcp, 0.0.0.0:9999->9999/tcp, :::9999->9999/tcp      pensive_gauss
```

Problem 3:

Create a docker compose to up mysql container, and
https://github.com/sabreensalama/dockerize-node-app-task which depend on mysqldb.
Add volume for mysqldb

Docker file

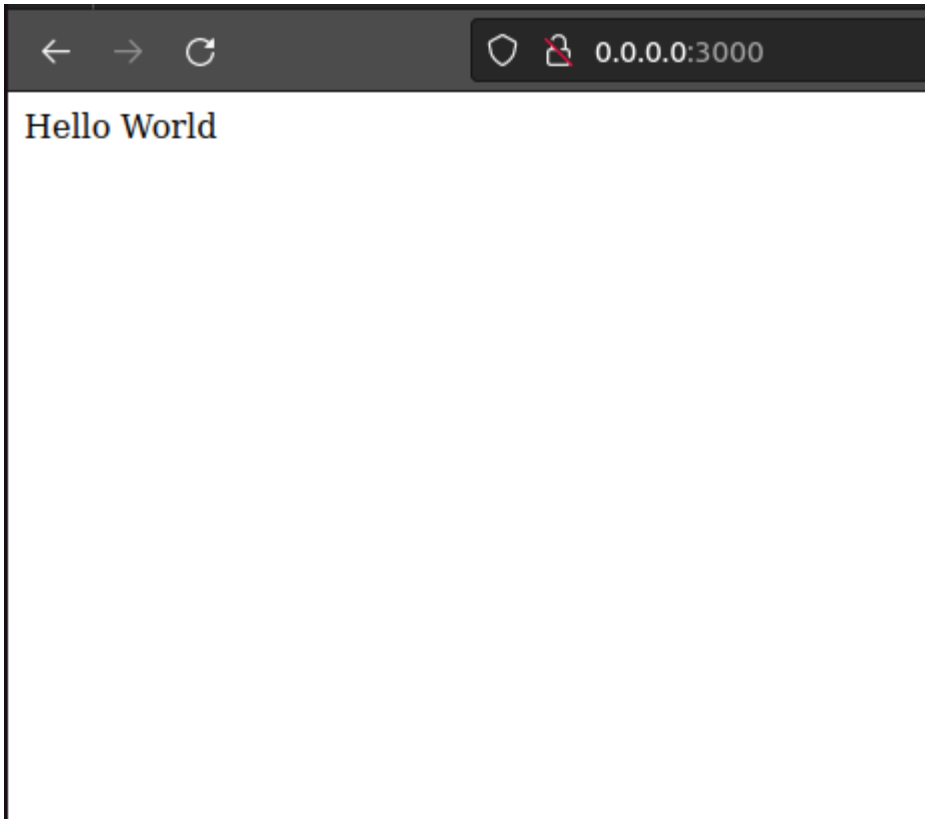


Docker compose file

root@heba-HP-ProBook-450-G4: ~

```yaml
version: '3.9'

services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
    depends_on:
      db:
        condition: service_healthy
    environment:
      DB_HOST: db
      DB_NAME: mydb
      DB_PASSWORD: mypassword
      DB_USER: myuser
    networks:
      - app-network
    ports:
      - "3000:3000"

  db:
    image: mysql:8.0
    environment:
      MYSQL_DATABASE: mydb
      MYSQL_USER: myuser
      MYSQL_PASSWORD: mypassword
      MYSQL_ROOT_PASSWORD: myrootpassword
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
      interval: 10s
      timeout: 5s
      retries: 3
    networks:
      - app-network
    ports:
      - "3306:3306"
    volumes:
      - db_data:/var/lib/mysql

networks:
  app-network:
    driver: bridge

volumes:
  db_data:
~
~
```

```
heba@heba-HP-ProBook-450-G4:~/docker-compose/dockerize-node-app-task$ docker-compose up
[+] Running 2/0
 ✔ Container dockerize-node-app-task-db-1   Created
 ✔ Container dockerize-node-app-task-app-1  Created
Attaching to dockerize-node-app-task-app-1, dockerize-node-app-task-db-1
dockerize-node-app-task-db-1    | 2023-05-30 16:29:00+00:00 [Note] [Entrypoint]: Entrypoint script for MyS
dockerize-node-app-task-db-1    | 2023-05-30 16:29:01+00:00 [Note] [Entrypoint]: Switching to dedicated us
dockerize-node-app-task-db-1    | 2023-05-30 16:29:01+00:00 [Note] [Entrypoint]: Entrypoint script for MyS
dockerize-node-app-task-db-1    | '/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
dockerize-node-app-task-db-1    | 2023-05-30T16:29:01.698865Z 0 [Warning] [MY-011068] [Server] The syntax
ost_cache_size=0 instead.
dockerize-node-app-task-db-1    | 2023-05-30T16:29:01.700774Z 0 [System] [MY-010116] [Server] /usr/sbin/my
dockerize-node-app-task-db-1    | 2023-05-30T16:29:01.767097Z 1 [System] [MY-013576] [InnoDB] InnoDB initi
dockerize-node-app-task-db-1    | 2023-05-30T16:29:02.994122Z 1 [System] [MY-013577] [InnoDB] InnoDB initi
dockerize-node-app-task-db-1    | 2023-05-30T16:29:04.517249Z 0 [Warning] [MY-010068] [Server] CA certific
dockerize-node-app-task-db-1    | 2023-05-30T16:29:04.517879Z 0 [System] [MY-013602] [Server] Channel mysq
dockerize-node-app-task-db-1    | 2023-05-30T16:29:04.635880Z 0 [Warning] [MY-011810] [Server] Insecure co
Consider choosing a different directory.
dockerize-node-app-task-db-1    | 2023-05-30T16:29:04.718756Z 0 [System] [MY-011323] [Server] X Plugin rea
dockerize-node-app-task-db-1    | 2023-05-30T16:29:04.718826Z 0 [System] [MY-010931] [Server] /usr/sbin/my
 MySQL Community Server - GPL.
dockerize-node-app-task-app-1   |
dockerize-node-app-task-app-1   | > docker_web_app@1.0.0 start /app
dockerize-node-app-task-app-1   | > node server.js
dockerize-node-app-task-app-1   |
dockerize-node-app-task-app-1   | Running on http://0.0.0.0:3000
```



Problem5:

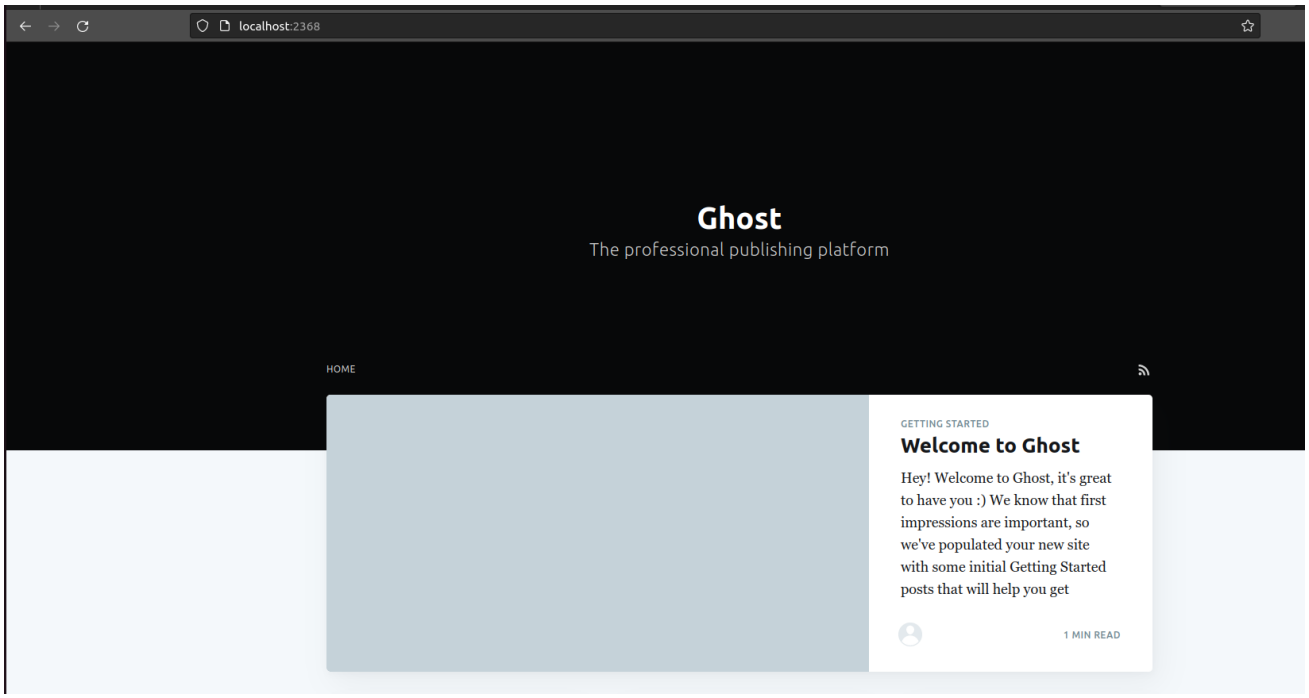
Use docker compose to deploy ghost platform (image: ghost:1-alpine)(Ghost is a free and open source blogging platform written in JavaScript)

Use mysql database instead of sqlite

root@heba-HP-ProBook-450-G4: ~ Terminal

```
version: '3'

services:
  ghost:
    image: ghost:1-alpine
    ports:
      - 2368:2368
    environment:
      - url=http://localhost:2368
      - database__client=mysql
      - database__connection__host=db
      - database__connection__user=root
      - database__connection__password=heba
      - database__connection__database=ghost
    networks:
      - docker-compose1-network
    depends_on:
      - db


  db:
    image: mysql:5.7
    environment:
      - MYSQL_DATABASE=ghost
      - MYSQL_ROOT_PASSWORD=heba
    networks:
      - docker-compose1-network
    volumes:
      - db_data:/var/lib/mysql

networks:
  docker-compose1-network:
    driver: bridge

volumes:
  db_data:
~
```

```
heba@heba-HP-ProBook-450-G4:~/docker-compose1$ docker-compose up
[+] Running 3/3
 ✔ Network docker-compose1_docker-compose1-network  Created
 ✔ Container docker-compose1-db-1                   Recreated
 ✔ Container docker-compose1-ghost-1                Recreated
Attaching to docker-compose1-db-1, docker-compose1-ghost-1
docker-compose1-db-1     | 2023-05-30 21:21:57+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.42-1.el7
docker-compose1-db-1     | 2023-05-30 21:21:58+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
docker-compose1-db-1     | 2023-05-30 21:21:58+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.42-1.el7
docker-compose1-db-1     | '/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
docker-compose1-db-1     | 2023-05-30T21:21:59.056768Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Pl
ore details).
docker-compose1-db-1     | 2023-05-30T21:21:59.058412Z 0 [Note] mysqld (mysqld 5.7.42) starting as process 1 ...
docker-compose1-db-1     | 2023-05-30T21:21:59.061671Z 0 [Note] InnoDB: PUNCH HOLE support available
```


localhost:2368
Ghost
The professional publishing platform
HOME
GETTING STARTED
Welcome to Ghost
Hey! Welcome to Ghost, it's great to have you :) We know that first impressions are important, so we've populated your new site with some initial Getting Started posts that will help you get
1 MIN READ