SimpleTimeService - Deployment Guide

The following documentation will guide you through the steps needed to clone the repository, set up the environment, and run the container.

Table of Contents

Prerequisites

Before you can deploy the **SimpleTimeService** container, ensure you have the following tools installed on your local machine:

1. **Docker**: Docker will be used to build and run the container.
2. **Git**: Git will be used to clone the repository.

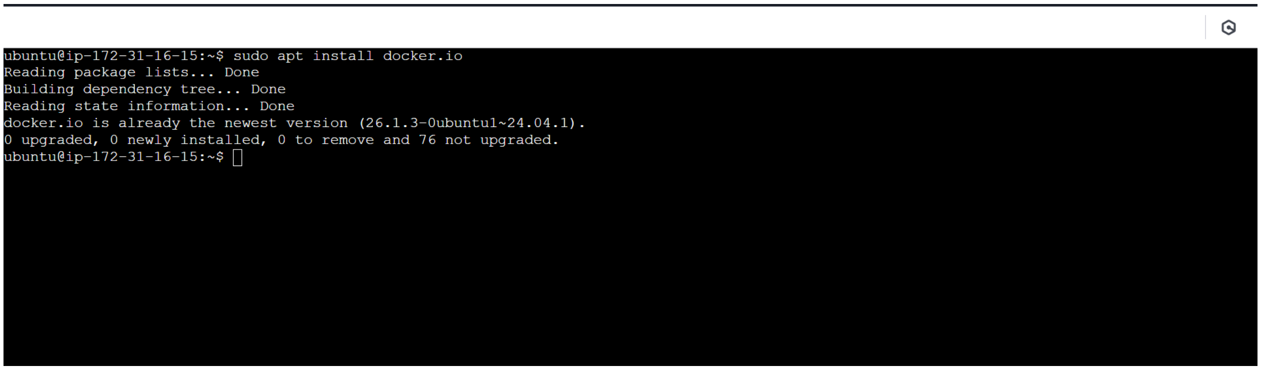
Tools Installation

1. Install Docker

Follow the instructions to install Docker:

- **Docker Installation Guide**: https://docs.docker.com/get-docker/
- **If you are using Ubuntu OS: sudo apt update && apt install docker.io**

  (Or simply “sudo snap install docker”)

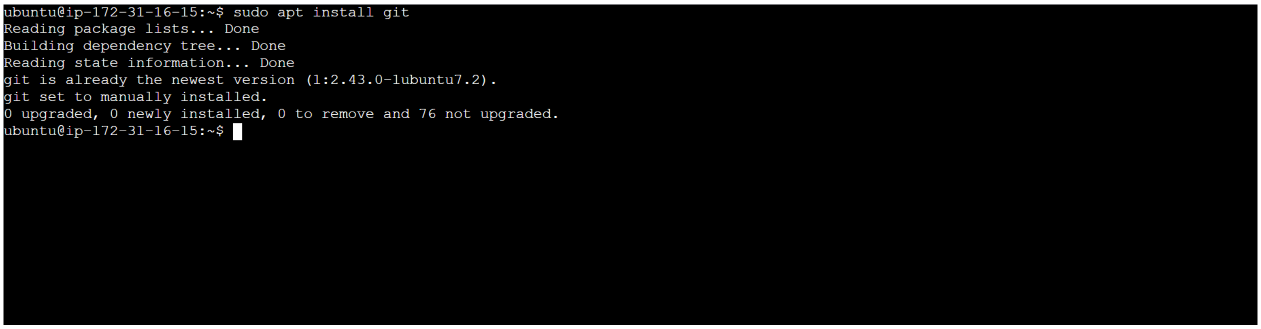


```
ubuntu@ip-172-31-16-15:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (26.1.3-0ubuntu1~24.04.1).
0 upgraded, 0 newly installed, 0 to remove and 76 not upgraded.
ubuntu@ip-172-31-16-15:~$
```

2. Install Git

Follow the instructions to install Git:

- **If you are using Ubuntu OS: sudo apt install git**
- **Git Installation Guide**: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

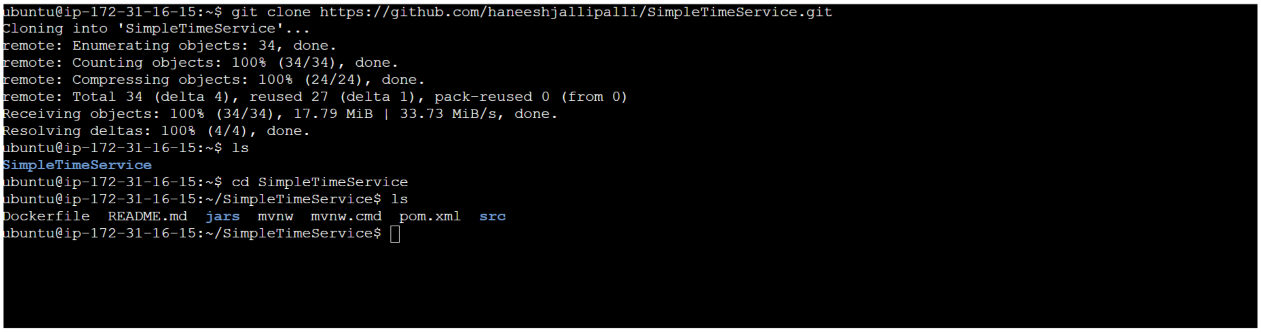


```
ubuntu@ip-172-31-16-15:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 76 not upgraded.
ubuntu@ip-172-31-16-15:~$
```

Repository Setup

Clone the repository to your local machine:

git clone https://github.com/haneeshjallipalli/SimpleTimeService.git
cd SimpleTimeService



```
ubuntu@ip-172-31-16-15:~$ git clone https://github.com/haneeshjallipalli/SimpleTimeService.git
Cloning into 'SimpleTimeService'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 34 (delta 4), reused 27 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 17.79 MiB | 33.73 MiB/s, done.
Resolving deltas: 100% (4/4), done.
ubuntu@ip-172-31-16-15:~$ ls
SimpleTimeService
ubuntu@ip-172-31-16-15:~$ cd SimpleTimeService
ubuntu@ip-172-31-16-15:~/SimpleTimeService$ ls
Dockerfile  README.md  jars  mvnw  mvnw.cmd  pom.xml  src
ubuntu@ip-172-31-16-15:~/SimpleTimeService$ 
```

This will pull the project files and move you into the project directory.

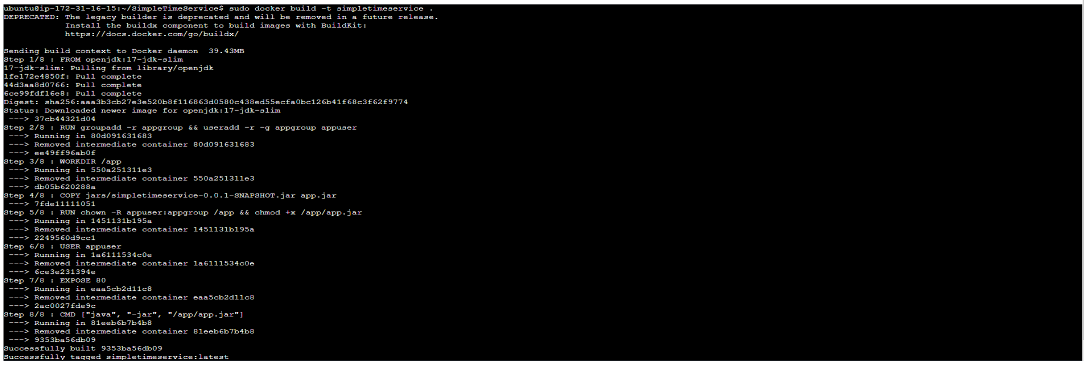
Running the Docker Container

Once the repository is set up locally, you can run the **SimpleTimeService** container.

1. Build the Docker Image

If you have a custom Dockerfile or want to ensure you have the latest build, use the following command to build the Docker image locally:

sudo docker build -t simpletimeservice .



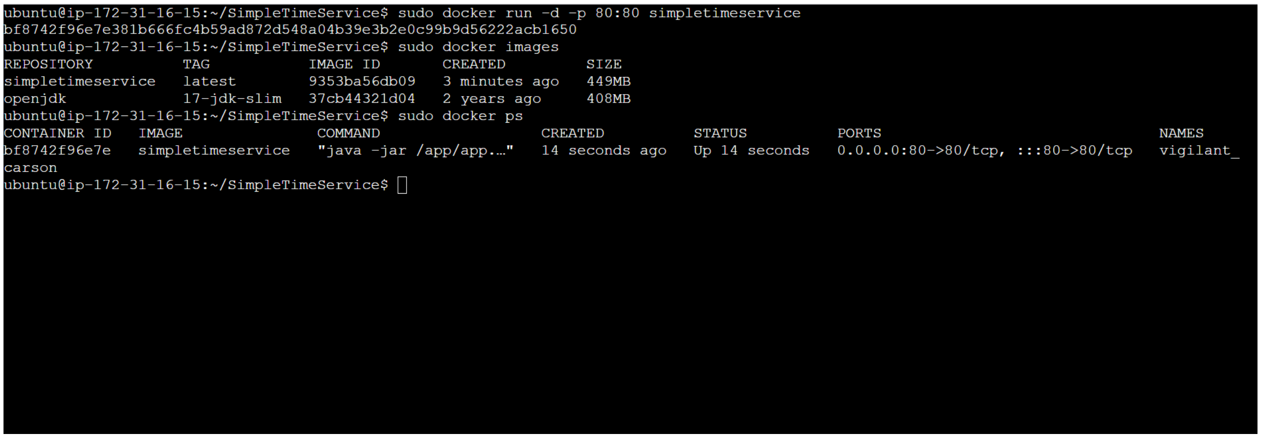
```
ubuntu@ip-172-31-16-15:~/SimpleTimeService$ sudo docker build -t simpletimeservice .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  39.43MB
Step 1/8 : FROM openjdk:17-jdk-slim
17-jdk-slim: Pulling from library/openjdk
1fe172e4850f: Pull complete
44d3aa8d0766: Pull complete
6ce99fdf16e8: Pull complete
Digest: sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774
Status: Downloaded newer image for openjdk:17-jdk-slim
 ---> 37cb44321d04
Step 2/8 : RUN groupadd -r appgroup && useradd -r -g appgroup appuser
 ---> Running in 80d091631683
 ---> Removed intermediate container 80d091631683
 ---> ee49ff96ab0f
Step 3/8 : WORKDIR /app
 ---> Running in 550a251311e3
 ---> Removed intermediate container 550a251311e3
 ---> db05b620288a
Step 4/8 : COPY jars/simpletimeservice-0.0.1-SNAPSHOT.jar app.jar
 ---> 7fde11111051
Step 5/8 : RUN chown -R appuser:appgroup /app && chmod +x /app/app.jar
 ---> Running in 1451131b195a
 ---> Removed intermediate container 1451131b195a
 ---> 2249560d9cc1
Step 6/8 : USER appuser
 ---> Running in 1a6111534c0e
 ---> Removed intermediate container 1a6111534c0e
 ---> 6ce3e231394e
Step 7/8 : EXPOSE 80
 ---> Running in eaa5cb2d11c8
 ---> Removed intermediate container eaa5cb2d11c8
 ---> 2ac0027fde9c
Step 8/8 : CMD ["java", "-jar", "/app/app.jar"]
 ---> Running in 81eeb6b7b4b8
 ---> Removed intermediate container 81eeb6b7b4b8
 ---> 9353ba56db09
Successfully built 9353ba56db09
Successfully tagged simpletimeservice:latest
```

This will build the image using the `Dockerfile` in the root of the repository and tag it as `simpletimeservice`.

2. Run the Docker Container

To run the **SimpleTimeService** container in detached mode, use the following command:

sudo docker run -d -p 80:80 simpletimeservice



```
ubuntu@ip-172-31-16-15:~/SimpleTimeService$ sudo docker run -d -p 80:80 simpletimeservice
bf8742f96e7e381b666fc4b59ad872d548a04b39e3b2e0c99b9d56222acb1650
ubuntu@ip-172-31-16-15:~/SimpleTimeService$ sudo docker images
REPOSITORY          TAG           IMAGE ID       CREATED         SIZE
simpletimeservice   latest        9353ba56db09   3 minutes ago   449MB
openjdk             17-jdk-slim   37cb44321d04   2 years ago     408MB
ubuntu@ip-172-31-16-15:~/SimpleTimeService$ sudo docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED          STATUS          PORTS                               NAMES
bf8742f96e7e   simpletimeservice   "java -jar /app/app.…"   14 seconds ago   Up 14 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp   vigilant_
carson
ubuntu@ip-172-31-16-15:~/SimpleTimeService$ 
```

This command does the following:

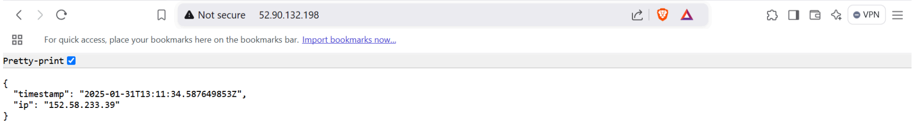
- `-d`: Runs the container in detached mode (in the background).
- `-p 80:80`: Maps port 80 on the host to port 80 on the container.
- `Simpletimeservice` is the docker image name

The container should now be running and accessible on port 80.

Accessing the Service

Once the container is up and running, you can access the **SimpleTimeService** by opening your browser and navigating to:

http://localhost:80


Not secure
52.90.132.198
VPN
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Pretty-print ☑

```
{
  "timestamp": "2025-01-31T13:11:34.587649853Z",
  "ip": "152.58.233.39"
}
```

Live demo: simpletimeservice.haneesh.cloud

You should see a simple response indicating that the service is running and serving the current time or similar functionality based on your container's configuration.

Additional Information

- **Docker Documentation**: https://docs.docker.com/
- **Docker Hub - haneeshdevops/simpletimeservice**: https://hub.docker.com/repository/docker/haneeshdevops/simpletimeservice

---

This documentation should guide you through the process of setting up and running the **SimpleTimeService** container. If you have any questions or run into issues, feel free to open an issue on the GitHub repository.
