This tutorial shows how you can deploy and run GALLOC using Docker. Before you start, you will need to install Docker on your computer. If you haven't done it yet, visit the Docker website (https://www.docker.com/) to download and install it.

Once Docker is installed on your computer, you have two options to run GALLOC: (1) using a Docker image of GALLOC that we shared on Docker Hub; and (2) using a Docker image of GALLOC created by yourself. Option (1) is simpler, but the Google Maps geocoding service used in GALLOC is not enabled (because it requires your own API key linked to a Google account) and the Nominatim map service is based the default URL (https://nominatim.openstreetmap.org/) which might not meet your specific needs. If you would like to enable the Google Maps service and/or using your own Nominatim service, you may want to use Option (2) which is not that complex. We recommend Option (1) for a simple exploration of GALLOC, and we recommend Option (2) for deployment for real data annotation work. Details for these two options are provided below.

(1) Using a Docker image that we shared on Docker Hub

We have created a Docker image for GALLOC and shared it on Docker Hub with the name of *sunkai8304/galloc:latest*. To use this image, do the following:

Step 1: Pull the GALLOC Docker image using the following command in your Terminal:

docker pull sunkai8304/galloc:latest

```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

>docker pull sunkai8304/galloc:latest
latest: Pulling from sunkai8304/galloc
Digest: sha256:a0596bfd009943a5990dfbe59c243a4ba4c5d06f7ab45a526af6aad9000bf6ee
Status: Image is up to date for sunkai8304/galloc:latest
docker.io/sunkai8304/galloc:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout
quickview sunkai8304/galloc:latest
```

Step 2: Verify the image is pulled successfully using the following command, and if successful, you will see the image you have pulled.

docker images

```
>docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
sunkai8304/galloc   latest    a0596bfd0099   13 minutes ago   751MB
```

Step 3: Run the Docker container using the following command:

docker run -d -p 8080:8080 --name your-container-name sunkai8304/galloc:latest

- For "your-container-name" in the command above: give a container name, such as "galloc-container"

```
>docker run -d -p 8080:8080 --name galloc_container sunkai8304/galloc:latest
07fce01bf93d7f8e648698c4300507dd87e06f6dc9e4e6a81c17fca98c0b1fcc
```

The long code in this image is the Container ID. You may have a different ID for your container.

Step 4: GALLOC is now ready. You can access it by opening a browser and navigating to: http://localhost:8080/GALLOC

(2) Using a Docker image created by yourself

As mentioned earlier, the Docker image that we uploaded to Docker Hub does not have Google Maps service enabled and it uses the default Nominatim. To enable the Google Maps service and/or use your own Nominatim, you can do the following steps:

Step 1: Download the source code of GALLOC from its GitHub repository (https://github.com/geoai-lab/GALLOC).

Step 2: Load the downloaded source code to an IDE (e.g., Eclipse). In the IDE, navigate to the folder "src/main/webapp", you should see a file called "config.json". Open this file and add your Google Maps API key and/or your own Nominatim URL.

Step 3: Use your IDE to export the edited source code into a ".war" file. Put the ".war" file in a directory on your computer that you will work with.

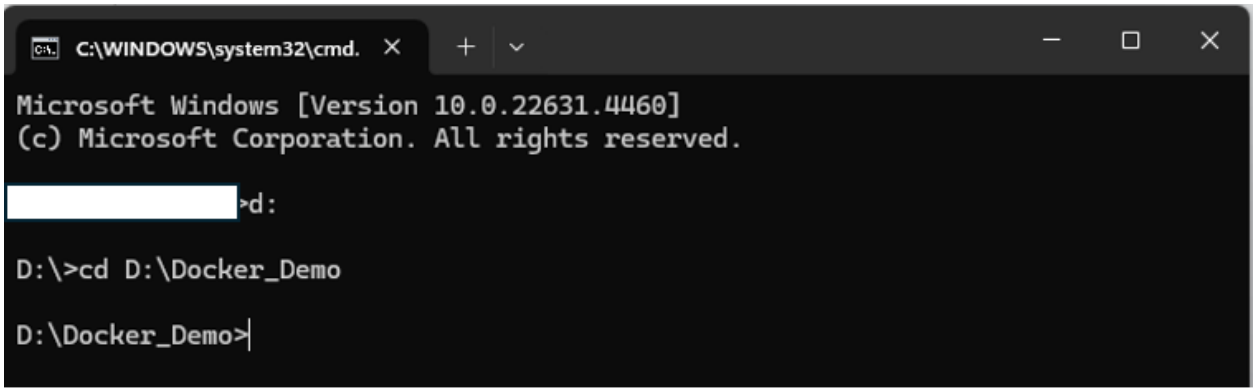
Step 4: Create a new file named "Dockerfile" and edit the Dockerfile based on the following example:

```
# Use an official Tomcat image as the base
FROM tomcat:9.0

# Copy the WAR file into the Tomcat webapps directory
COPY GALLOC.war /usr/local/tomcat/webapps/

# Expose the default Tomcat port
EXPOSE 8080
```

Step 5: Use the Terminal app on your computer, and navigate to the directory containing your Dockerfile.



Step 6: Build the Docker image by typing the following command in Terminal:

docker build -t your-image-name .

- For "your-image-name" in the command above: give a name to the image, such as "galloc"

```
D:\Docker_Demo>docker build -t galloc .
[+] Building 2.0s (8/8) FINISHED                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                      0.0s
 => => transferring dockerfile: 262B                                      0.0s
 => [internal] load metadata for docker.io/library/tomcat:9.0             0.6s
 => [auth] library/tomcat:pull token for registry-1.docker.io             0.0s
 => [internal] load .dockerignore                                         0.0s
 => => transferring context: 2B                                           0.0s
 => [internal] load build context                                         0.6s
 => => transferring context: 19.73MB                                      0.6s
 => CACHED [1/2] FROM docker.io/library/tomcat:9.0@sha256:1746595b30bd     0.0s
 => => resolve docker.io/library/tomcat:9.0@sha256:1746595b30bd1f4f9c5     0.0s
 => [2/2] COPY GALLOC.war /usr/local/tomcat/webapps/                      0.1s
 => exporting to image                                                    0.6s
 => => exporting layers                                                   0.5s
 => => exporting manifest sha256:3bb11e4f48111f238e6e73fcb86ede9fc901c     0.0s
 => => exporting config sha256:a3c6dcb891cb05c3b1169574ef7d6725e6be296     0.0s
 => => exporting attestation manifest sha256:200d7e17d0c954aa57727722f     0.0s
 => => exporting manifest list sha256:a0596bfd009943a5990dfbe59c243a4b     0.0s
 => => naming to docker.io/library/galloc:latest                          0.0s
 => => unpacking to docker.io/library/galloc:latest                       0.1s

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout q
uickview
```

Step 7: Run a container from the image that you built using the command below:

docker run -d -p 8080:8080 --name your-container-name your-image-name

- For "your-container-name" in the command above: give a container name, such as "galloc-container"

```
D:\Docker_Demo>docker run -d -p 8080:8080 --name galloc-container galloc
f1b17db707ab90eb21d8d4afb76a2698c6dd0a52af4a0160d0f3644ccd024f83
```

The long code in this image is the Container ID. You may have a different ID for your container.

Step 8: GALLOC is now ready. You can access the application by opening a browser and navigating to: http://localhost:8080/your-app (please replace "your-app" with the name of your .war file; for example, if your war file is named "GALLOC.war", then the URL is http://localhost:8080/GALLOC)