System Documentation

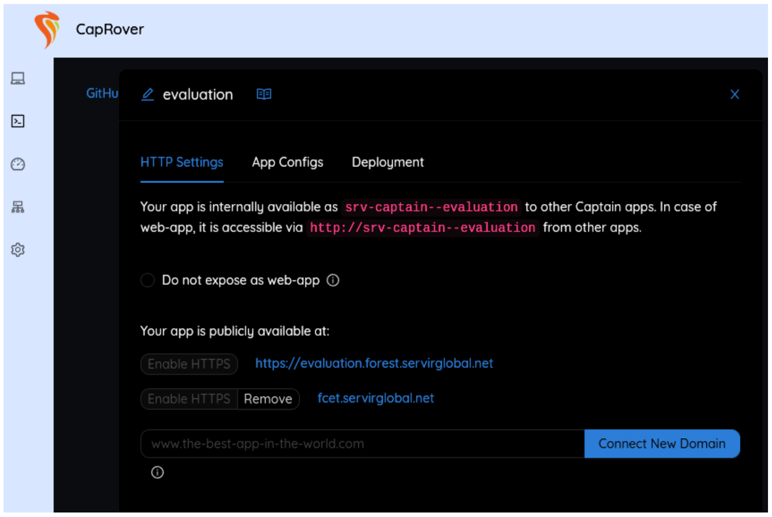
This document contains a high level description of how the system was implemented, and brief indications to basic application management. The document will continue as follows, the CapRover app deployment server, a system description for each tool, and the external components required to function.

CapRover app deployment server

Applications are deployed using the CapRover software hosted in the NASA Servir server with IP 216.218.226.135. To access the administrator dashboard you have to enter the following url: https://captain.forest.servirglobal.net and enter the password: capAdminIDB. To have more information on CapRover please refer to its documentation.

CapRover is an application deployment management system that allows users to easily set up a Continuous Integration and Development (CI/CD) workflow. This helps reduce the time and work needed to maintain, fix, and update applications.

One of the features of CapRover is that it allows users to easily assign a url, add new urls to applications and also to enable secure connection (HTPPS).



Your app is internally available as `srv-captain--evaluation` to other Captain apps. In case of web-app, it is accessible via `http://srv-captain--evaluation` from other apps.

Do not expose as web-app

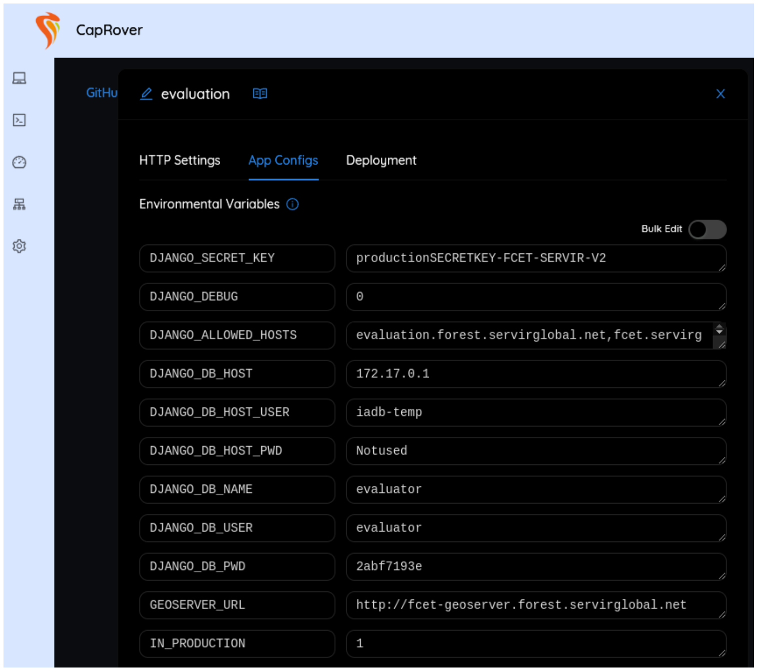
Your app is publicly available at:

Enable HTTPS https://evaluation.forest.servirglobal.net

Enable HTTPS Remove fcet.servirglobal.net

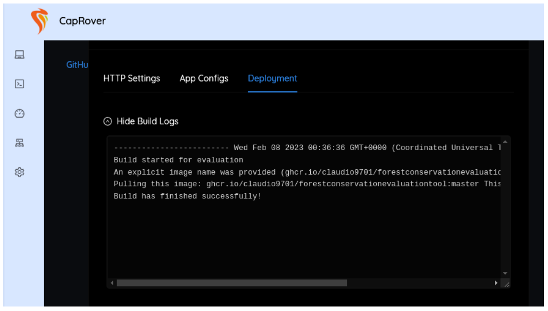
www.the-best-app-in-the-world.com Connect New Domain

Also, to avoid exposing sensitive credentials such as passwords, API Keys, among others, CapRover allows you to set up Environment variables for each application. These values will be set only on the isolated application docker container.

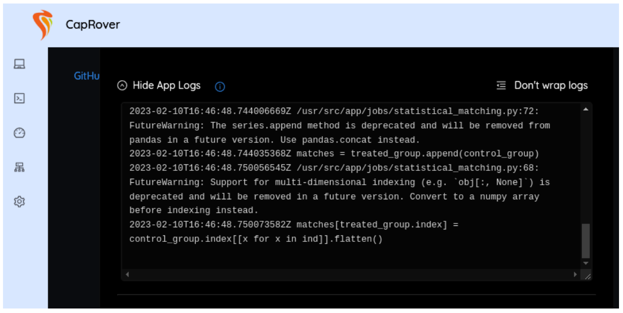


DJANGO_SECRET_KEY	productionSECRETKEY-FCET-SERVIR-V2
DJANGO_DEBUG	0
DJANGO_ALLOWED_HOSTS	evaluation.forest.servirglobal.net,fcet.servirg
DJANGO_DB_HOST	172.17.0.1
DJANGO_DB_HOST_USER	iadb-temp
DJANGO_DB_HOST_PWD	Notused
DJANGO_DB_NAME	evaluator
DJANGO_DB_USER	evaluator
DJANGO_DB_PWD	2abf7193e
GEOSERVER_URL	http://fcet-geoserver.forest.servirglobal.net
IN_PRODUCTION	1

Similarly, CapRover makes it easy to see the applications logs. These are separated in development and applications logs. Development logs come from the process of building and running the application docker container. Application logs come from the application itself while it is running. Troubleshooting is best done by looking at deployment and application logs in the CapRover admin interface.

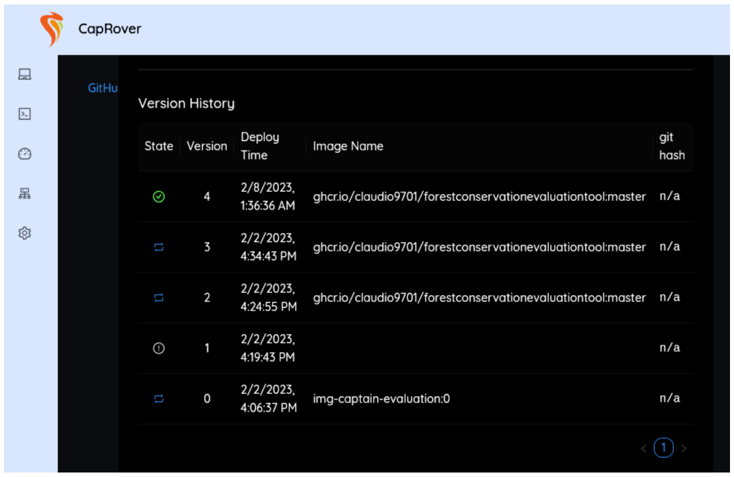


```
------------------------ Wed Feb 08 2023 00:36:36 GMT+0000 (Coordinated Universal T
Build started for evaluation
An explicit image name was provided (ghcr.io/claudio9701/forestconservationevaluatic
Pulling this image: ghcr.io/claudio9701/forestconservationevaluationtool:master This
Build has finished successfully!
```



```
2023-02-10T16:46:48.744006669Z /usr/src/app/jobs/statistical_matching.py:72:
FutureWarning: The series.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
2023-02-10T16:46:48.744035368Z matches = treated_group.append(control_group)
2023-02-10T16:46:48.750056545Z /usr/src/app/jobs/statistical_matching.py:68:
FutureWarning: Support for multi-dimensional indexing (e.g. `obj[:, None]`) is
deprecated and will be removed in a future version. Convert to a numpy array
before indexing instead.
2023-02-10T16:46:48.750073582Z matches[treated_group.index] =
control_group.index[[x for x in ind]].flatten()
```

Finally if there is any bug with the latest version of an application. CapRover allows you to revert to a previous functioning version with just a click in the version history table.



Version History

State	Version	Deploy Time	Image Name	git hash
✓	4	2/8/2023, 1:36:36 AM	ghcr.io/claudio9701/forestconservationevaluationtool:master	n/a
⇄	3	2/2/2023, 4:34:43 PM	ghcr.io/claudio9701/forestconservationevaluationtool:master	n/a
⇄	2	2/2/2023, 4:24:55 PM	ghcr.io/claudio9701/forestconservationevaluationtool:master	n/a
ⓘ	1	2/2/2023, 4:19:43 PM		n/a
⇄	0	2/2/2023, 4:06:37 PM	img-captain-evaluation:0	n/a

Both applications are deployed using Docker under the hood. Docker allows us to build, test and deploy applications in containers which are the modern lightweight version of virtual machines used to deploy applications. Furthermore, CapRover allows easy deployment of new applications on the server. Apart from the previously mentioned methods, there are several alternatives described in the CapRover documentation. To have further detail on how to deploy an applications from a Github repository in CapRover refer to these links:

- Simple application deployment
- Complex application deployment

The applications information will be detailed in the following sections.

Evaluation tool

This application is available in the following links:

- https://evaluation.forest.servirglobal.net
- https://fcet.servirglobal.net

FCET is a complex application, thus to deploy it we are using an automated CI/CD process:

1. Update and publish the application as a package (docker image) directly from the Github repository to the Github Container registry (https://ghcr.io)
2. Download and deploy the published package as a web application on our server.

This process will be triggered automatically when the github repository's master branch is updated or manually from the "Actions" tab from the repository.

External applications requirements

Tomcat Geoserver

The application configuration proxied any request to `[main_url]/geoserver` to a Tomcat application running in the docker host machine on 8080. This is done with a NGINX Reverse

Proxy application called “fcet-geoserver” which points to the internal docker host machine IP and port 8080: “http://172.17.0.1:8080”. Tomcat is also enabled to run on startup through `systemd`. You can restart tomcat using:

```
$ systemctl restart tomcat
```

Configuration file locations:

- The primary files for tomcat are stored at: `/home/scoconf/appservers/apache-tomcat-7x`
- Geoserver data and settings are housed here can be found at: `/home/scoconf/appservers/apache-tomcat-7x/webapps/geoserver`
- The `data` directory inside can be transferred to another Geoserver while maintaining the maps, styles, and layers that have been created on Geoserver for the FCET.
- We use a custom `systemd` script located at:

  ```
  $ /etc/systemd/system/tomcat.service
  ```

PostgreSQL Database

We run PostgreSQL 9.5 on the docker host machine (172.17.0.1) in the default port 5432. Our application uses PostgreSQL aggressively and the custom configurations are important.

We enable several extensions for use with the tool:

- postgis
- postgis_topology
- fuzzystrmatch
- postgis_tiger_geocoder

PostgreSQL is also enabled on `systemd` and can be restarted using:

```
$ systemctl restart postgresql
```

The primary user and database are as follows:

- user: evaluator
- password: 2abf7193e
- database: evaluator

Configuration file locations:

- The configuration file is at: `/etc/postgresql/9.5/main/postgresql.conf`
- A database backup for evaluator is located at: `/home/scoconf/config/postgresql/evaluator.sql.gz`

Targeting tool

This application is available in the following links:

- https://targeting.forest.servirglobal.net
- https://fctt.servirglobal.net

To update or deploy the FCTT application in CapRover we need to do the following steps:

1. Enter and login to de App Manager
2. Go to the “Apps” section in the left sidebar and enter to the “fctt” application
3. Go to the “Deployment tab” and click on the “Force Build button” on the “Method 3: Deploy from Github/Bitbucket/Gitlab” section
4. Repeat step 3 for the “fctt-php” application on the “Apps” section

We have separated the PHP application as backend and the NGINX application as the frontend. The configuration files that CapRover needs to read to deploy the applications are called “captain-definition-backend” and “captain-definition-frontend” respectively. The application specific configuration files are located in the “conf” directory under “php” and “nginx” directories respectively. Finally, the application source code is stored in the “src” directory. Read and write permissions to this folders is needed for functionality with persistent data:

- usersystem/users/
- usersystem/uploads/
- usersystem/unitofanalysis_shapefiles/

External applications

Tomcat Geoserver

As well as in the FCET there is a Tomcat Geoserver running for this application. This Tomcat application is running on another NASA server (IP 216.218.226.134). Thus, any request to `[main_url]/geoserver` is proxied to the server internal network IP: “http://10.5.5.25/geoserver”. This request is then proxied again to “http://127.0.0.1:4894/geoserver/” inside the NASA Server running the Tomcat Geoserver.

Inside the 216.218.226.134 server, Tomcat is also enabled to run on startup through the `systemd` command. You can restart tomcat using:

```$ systemctl restart tomcat```

The geoserver login webpage can be accessed at `[main_url]/geoserver`:

- user: forestro
- password: NASA1616

The geoserver administrator login is admin, password: NASA1616.

Configuration file locations:

- The primary files for tomcat are stored at: `/home/scoconf/appservers/apache-tomcat-7x`
- Geoserver data and settings are housed here can be found at: `/home/scoconf/appservers/apache-tomcat-7x/webapps/geoserver`
- The `data` directory inside can be transferred to another Geoserver while maintaining the maps, styles, and layers that have been created on Geoserver for the FCTT.
- We use a custom `systemd` script located at:

  ```$ /etc/systemd/system/tomcat.service```

PostgreSQL Databases

We run PostgreSQL 9.5 on another NASA server (IP 216.218.226.134) on the default port 5432. Connections to the database have to set server internal IP “10.5.5.25” as the host. Our application uses PostgreSQL aggressively and the custom configurations are important.

We enable several extensions for use with the tool:
- postgis
- postgis_topology
- fuzzystrmatch
- postgis_tiger_geocoder

PostgreSQL is also enabled on `systemd` and can be restarted using:

```$ systemctl restart postgresql```

The primary user and database are as follows:

- user: forestro_users
- password: 2abf7193e
- database: forestro_users_db

Configuration file locations:

- The configuration file is at: `/etc/postgresql/9.5/main/postgresql.conf`
- A database back for forestro_users_db is located at: `/home/scoconf/config/postgresql/fctt_db.tar.gz`

Additional configuration mentioned in the previous system documentation

Front-end: JavaScript Single Page App

The main application code lives in a file with the prefix `fctt_user_`. Suffixes indicate a version of the code, and whether or not it has been "uglified" using fc-targeting-tool@rff.org to make the code harder to replicate (this is indicated by "ug"). The current version of the code in readable format is named `fctt_user.js`, though I have only been keeping this file locally. When uploading to the web, I've used the uglified version (current version is `fctt_user_v13_ug.js`).

Back-end: PHP

The files `data_setup_v1x.php` automated some of the data ingestion process for loading data into the FCTT, though these are not used by the users and shouldn't be kept online. Within `uploads/`, the subfolder `autoimport_onboarddata/` contains files for use with `data_setup_vx.php` files.

Database: PostgreSQL

The PostgreSQL database used by the FCTT is called `forestro_users_db`. Spatial data is stored in tables within the public schema:

- On board data (Mexico predios, MREDD areas, CA 10km and 1km cells, SA 10km and 1km cells), is stored here as tables that begin with `obd_`. User defined datasets (output from the User Console) are stored here as `userdata_[username]_[tablename]`.

- The table `standingdesk` stores user `layerPINs`, which is a system that prevents users from viewing layers associated with other users' accounts. `layerPINs` are also stored in the user's individual XML file, so the PIN is loaded into the client's browser after they login to their account, which can then be passed to GeoServer to authenticate requests to their private user layers.

Other relevant documents

1. acugis_changelog.txt` (documents server changes that were made when we were hosting on AcuGIS, most of which would have been propagated over to you). Credentials used for AcuGIS hosting were forestro/NASA1616.
2. `FCTT Master Document.docx` contains detailed documentation from when Len Goff left RFF, at which point this documentation was created as a reference to future RAs. Since then, most changes to the FCTT have been cosmetic or involved adding further on-board data to the system (e.g. expanding the geographic scope to include South America), not affecting the structure of the architecture.
3. `combined_sqlcalls_v13` documents SQL calls run for the setup of onboard data in the database, as well as loaded into Geoserver as SQL Views.