



CHAMELEON EVCFLO PROJECT

Web Dev Team Handover

Leigh Rowell and Ezekiel Griffin
lrowell@deakin.edu.au | ekgriffin@deakin.edu.au



Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFLO)

Table of Contents

Tech Stack and IDE:.....	2
Front end:.....	2
Back end:.....	2
IDE:	2
Updating the Front End:	4
Packaging and Deploying Docker container to Google Cloud Platform:	7
Install the GCP CLI on your local machine:	7
Deploy to Cloud Run from source:.....	7
Use the following procedure to update the artifact to the latest deployment:.....	8
Connecting to the Git Repo:	12
Pushing updates to the Git repository:.....	14
Pulling Updates from the Repository:.....	16
Web Server API:	18
Links:	20
Contacts:	21



Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFLO)

Tech Stack and IDE:

Front end:

The front end for this project is a very simple combination of HTML, CSS and JS. Currently there is no framework being employed to build the front end upon.

Future plans for this project are to build a front-end web client using the ReactJS framework, inline with the company branding and layout as set by the web development team this trimester (T2 2022).

A good resource for upskilling is w3 schools, here: <https://www.w3schools.com/html/>

Back end:

The back-end web server for this project is written in Python and built using the Flask framework. This was chosen based upon the majority of the data analysis being done using the Python language it reduced the learning curve for new team members having to simultaneously learn multiple languages. It also enables the potential use of many pre-existing Python libraries for future predictive modelling being built into the web service if required.

Flask Documentation here: <https://flask.palletsprojects.com/en/2.2.x/>

IDE:

The recommended IDE to use is VS Code, due to the ability to use it for all components of the project by simply adding in the plugins required for the language/framework being used. It is also cross platform so can be used on Windows, Linux or Mac. Download here: <https://code.visualstudio.com/download>



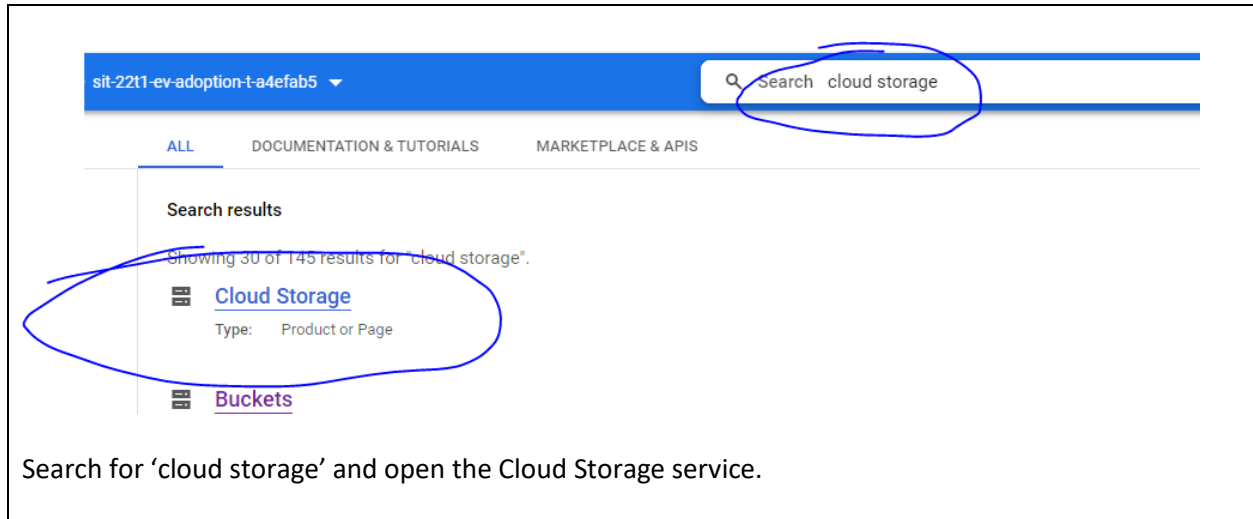
Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFLO)

Updating the Front End:

Log into your browser using Deakin credentials in order to access the GCP.

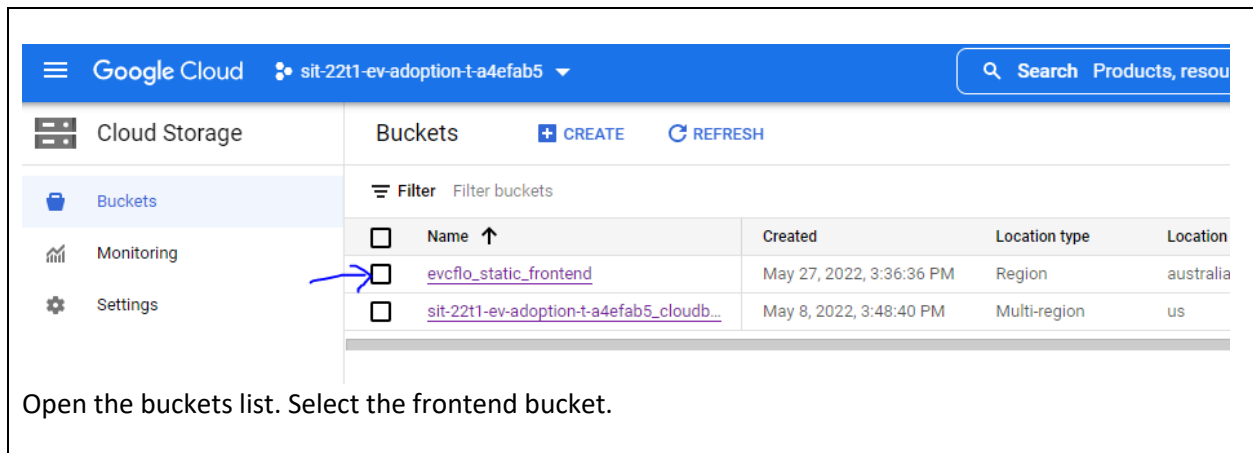
Open Google Cloud Console (<https://console.cloud.google.com/>)

Make sure you are in the project 'sit-22t1-ev-adoption-t-a4efab5'



The screenshot shows the Google Cloud Console interface. At the top, the project name 'sit-22t1-ev-adoption-t-a4efab5' is displayed. A search bar contains the text 'cloud storage'. Below the search bar, the 'Search results' section shows 'Showing 30 of 145 results for "cloud storage"'. The first result is 'Cloud Storage', which is circled in blue. Below it, 'Buckets' is also listed. The 'Type' for 'Cloud Storage' is 'Product or Page'.

Search for 'cloud storage' and open the Cloud Storage service.



The screenshot shows the Google Cloud Console interface with the 'Buckets' list selected. The left sidebar shows 'Cloud Storage' and 'Buckets' under the 'Filter' section. The main content area displays a table of buckets. The first bucket, 'evcflo_static_frontend', is selected with a blue arrow. The second bucket, 'sit-22t1-ev-adoption-t-a4efab5_cloudb...', is also listed.

Name	Created	Location type	Location
evcflo_static_frontend	May 27, 2022, 3:36:36 PM	Region	australia
sit-22t1-ev-adoption-t-a4efab5_cloudb...	May 8, 2022, 3:48:40 PM	Multi-region	us

Open the buckets list. Select the frontend bucket.

Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFO)

OBJECTS CONFIGURATION PERMISSIONS PROTECTION LIFECYCLE

Buckets > evcflo_static_frontend

UPLOAD FILES UPLOAD FOLDER CREATE FOLDER TRANSFER DATA MANAGE HOLDS DOWNLOAD

Filter by name prefix only Filter Filter objects and folders

<input type="checkbox"/>	Name	Size	Type	Created	Storage class	Last modified
<input type="checkbox"/>	add_station.html	4.6 KB	text/html	May 28, 20...	Standard	May 28, 20...
<input type="checkbox"/>	index.html	1.3 KB	text/html	May 28, 20...	Standard	May 28, 20...
<input type="checkbox"/>	main.js	1.7 KB	text/javascript	May 27, 20...	Standard	May 27, 20...
<input type="checkbox"/>	notebooks.html	5 KB	text/html	May 29, 20...	Standard	May 29, 20...
<input type="checkbox"/>	notebooks/	—	Folder	—	—	—
<input type="checkbox"/>	style.css	294 B	text/css	May 27, 20...	Standard	May 27, 20...

Select UPLOAD FILES to open the file selector.

Open

« Chameleon » EV Project » Git Repo » frontend

Search frontend

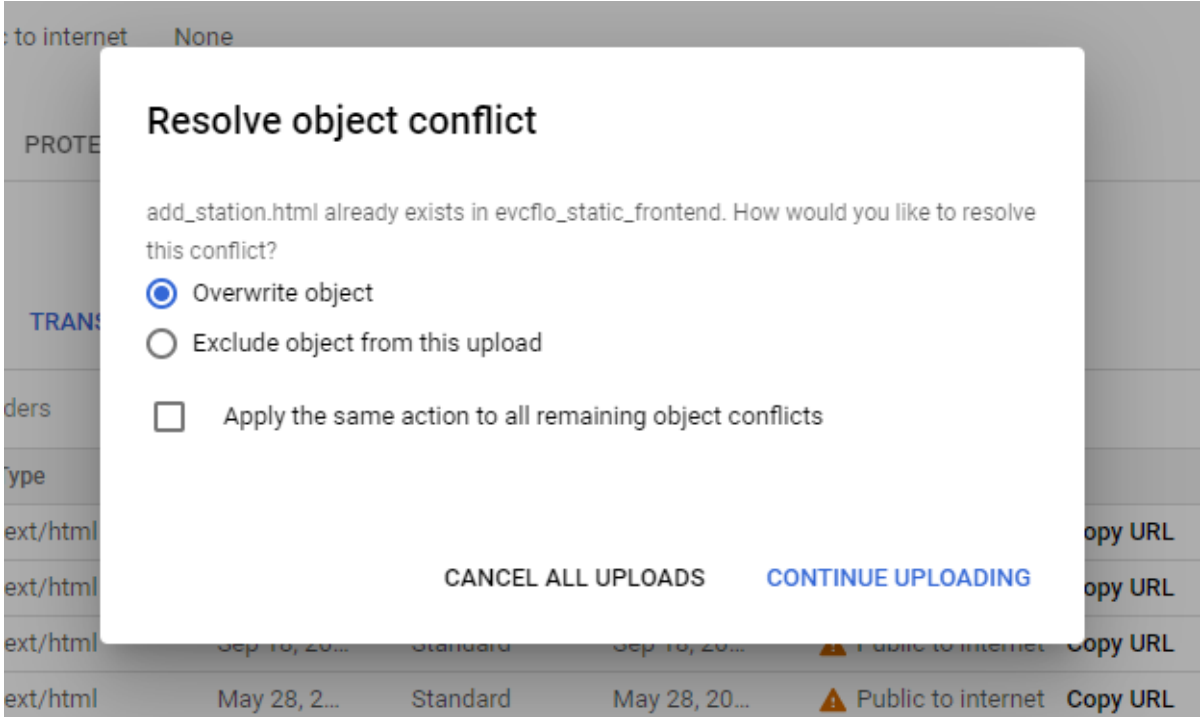
Organize New folder

Name	Date modified	Type	Size
notebooks	3/08/2022 5:10 PM	File folder	
add_bounding.html	7/09/2022 7:57 PM	Chrome HTML Do...	3 KB
add_station.html	7/09/2022 7:57 PM	Chrome HTML Do...	6 KB
add_suggested_station.html	18/09/2022 8:00 PM	Chrome HTML Do...	3 KB
index.html	18/09/2022 8:00 PM	Chrome HTML Do...	3 KB
main.js	18/09/2022 7:21 PM	JavaScript File	7 KB
notebooks.html	7/09/2022 7:57 PM	Chrome HTML Do...	6 KB
style.css	3/08/2022 5:10 PM	Cascading Style S...	1 KB

File name: "add_bounding.html" "add_station.html" "add_suggested_station.html" "index.html" "main.js" "n" All Files (*.*)

Open Cancel

Navigate to the source folder and select the new front end files to update the website.



Resolve object conflict

add_station.html already exists in evcflo_static_frontend. How would you like to resolve this conflict?

☒ Overwrite object

☐ Exclude object from this upload

☐ Apply the same action to all remaining object conflicts

CANCEL ALL UPLOADS CONTINUE UPLOADING

Select 'Overwrite object' option to replace any old files with the new, and continue.

Once the files have finished uploading, the changes will be live on the public website. Note that you may have to erase browser cache in order for the page to display correctly depending on the cache settings you are using in your browser.

Packaging and Deploying Docker container to Google Cloud Platform:

Refer the following GCP guide:

<https://cloud.google.com/run/docs/quickstarts/build-and-deploy/deploy-python-service>

This will walk you through creating the Dockerfile, requirements.txt, and .dockerignore files which are required in order to create the docker image.

Substitute in the root server folder (containing server.py) instead of the sample helloworld.py source code.

Note that these files are already existing in the project source code, but may require review or updating depending on your deployment.

The project to select is: "sit-22t1-ev-adoption-t-a4efab5"

Install the GCP CLI on your local machine:

Refer the following guides for instructions on how to do this:

<https://cloud.google.com/sdk/docs/install>

Make sure the GCP CLI is initialized so that you are able to use it, refer this guide:

<https://cloud.google.com/sdk/docs/initializing>

Deploy to Cloud Run from source:

Refer to the section 'Deploy to Cloud Run from source' in the quickstart guide for the following.

From the command prompt navigate to the root server folder containing server.py and run the following command:

'gcloud run deploy'

Press enter to select the current source code location.

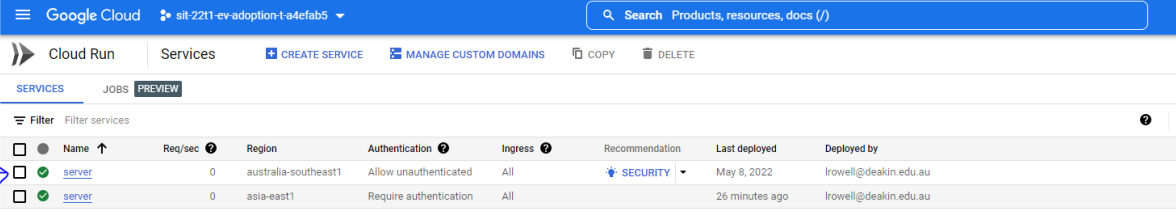
Enter 'server' for the service name

Select the region: 'australia-southeast1'

Continue through the process until the service is complete, you will receive a service URL. This will be disregarded as we do not need to use it. Instead we will update the existing server to use the docker artifact which was just created on the GCP platform.

Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFO)

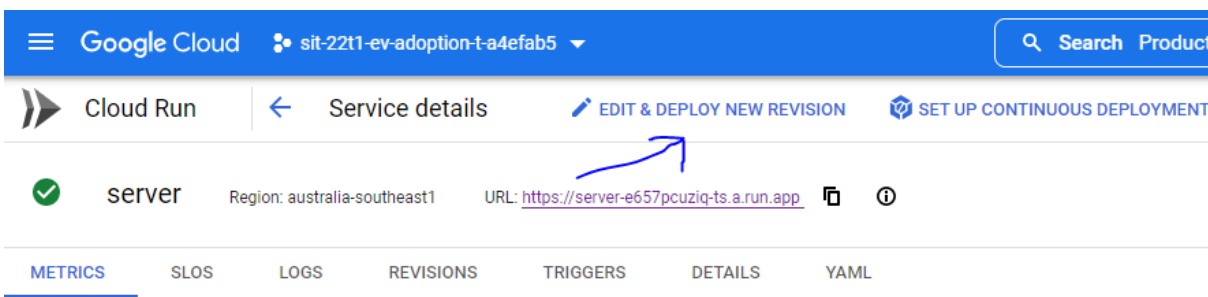
Use the following procedure to update the artifact to the latest deployment:



The screenshot shows the Google Cloud Run 'Services' page for project 'sit-22t1-ev-adoption-t-a4efab5'. A table lists two services, both named 'server'. The first service is in the 'australia-southeast1' region with 'Allow unauthenticated' authentication. The second service is in the 'asia-east1' region with 'Require authentication'.

Name	Req/sec	Region	Authentication	Ingress	Recommendation	Last deployed	Deployed by
server	0	australia-southeast1	Allow unauthenticated	All	SECURITY	May 8, 2022	Irowell@deakin.edu.au
server	0	asia-east1	Require authentication	All		26 minutes ago	Irowell@deakin.edu.au

From the GCP Cloud Run dashboard page
<https://console.cloud.google.com/run?authuser=0&orgonly=true&project=sit-22t1-ev-adoption-t-a4efab5&supportedpurview=organizationId>
 Locate the existing service which we are using to host the public web server. This is named 'server' and is identifiable by the 'Allow unauthenticated' property under Authentication. Click the 'server' link to open the service.



The screenshot shows the 'Service details' page for the 'server' service in the 'australia-southeast1' region. The URL is 'https://server-e657pcuziq-ts.a.run.app'. A blue arrow points to the 'EDIT & DEPLOY NEW REVISION' link.

Click the link to 'EDIT & DEPLOY NEW REVISION'.

Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFL0)

Google Cloud

• sit-22t1-ev-adoption-t-a4efab5

Cloud Run

Deploy revision to server (australia-southeast1)

Every change to the service configuration creates an immutable revision. A revision consists of a specific container image, along with other environment settings.

CONTAINER

CONNECTIONS

SECURITY

General

Container image URL

australia-southeast1-docker.pkg.dev/sit-22t1-ev-adoption-t-a4efab5/cloud

SELECT

E.g. us-docker.pkg.dev/cloudrun/container/hello
Should listen for HTTP requests on \$PORT and not rely on local state. [How to build a container?](#)

Container port

8080

Requests will be sent to the container on this port. We recommend listening on \$PORT

Click the Container Image URL link to select a new container image.

Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFL0)

Select container image

CONTAINER REGISTRY

ARTIFACT REGISTRY

Project: sit-22t1-ev-adoption-t-a4efab5 [CHANGE](#)

▼

asia-east1-docker.pkg.dev/sit-22t1-ev-adoption-t-a4efab5/cloud-run-source

▼ server

474b625335 latest

▶ australia-southeast2-docker.pkg.dev/sit-22t1-ev-adoption-t-a4efab5/cloud-r

▶ australia-southeast1-docker.pkg.dev/sit-22t1-ev-adoption-t-a4efab5/cloud-r

SELECT

CANCEL

Under the ARTIFACT REGISTRY tab, you should be able to locate the latest image (usually the top one). Expand this out and locate the image, once you click this the SELECT button will be clickable. Click Select.

Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFL0)

+ ADD VARIABLE

Secrets ?

REFERENCE A SECRET

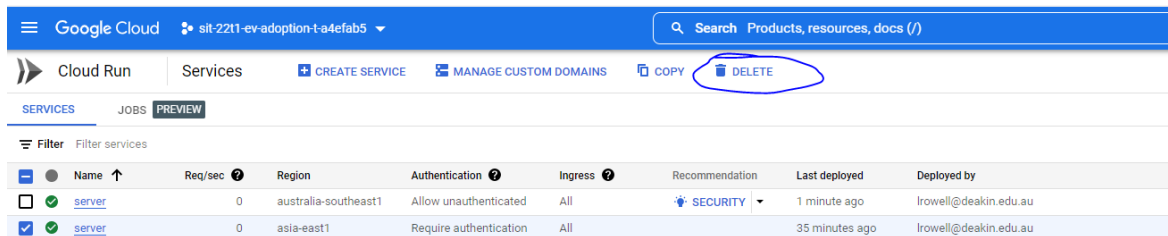
☒ **Serve this revision immediately**
100% of the traffic will be migrated to this revision, overriding all existing traffic splits, if any.

DEPLOY

CANCEL

Scroll to the bottom of the page and click DEPLOY.

If this is completed correctly the new version of the server will be deployed and reachable from the same existing URL that we have been using (<https://server-e657pcuziq-ts.a.run.app/>).



The screenshot shows the Google Cloud Cloud Run dashboard for project 'sit-2211-ev-adoption-ta4efab5'. The 'SERVICES' tab is active, showing a table of deployed services. Two services named 'server' are listed. The first service is in the 'australia-southeast1' region, has 'Allow unauthenticated' authentication, and was deployed 1 minute ago. The second service is in the 'asia-east1' region, has 'Require authentication' authentication, and was deployed 35 minutes ago. A blue arrow points to the checkbox for the second service. In the top right, the 'DELETE' button is circled in blue.

	Name	Req/sec	Region	Authentication	Ingress	Recommendation	Last deployed	Deployed by
<input type="checkbox"/>	server	0	australia-southeast1	Allow unauthenticated	All	SECURITY	1 minute ago	Irowell@deakin.edu.au
<input checked="" type="checkbox"/>	server	0	asia-east1	Require authentication	All		35 minutes ago	Irowell@deakin.edu.au

Clean up by returning to the Cloud Run dashboard, selecting the redundant deployed service (the latest deployment created using the 'gcloud run deploy' command). Select this and click DELETE to remove it from the project.

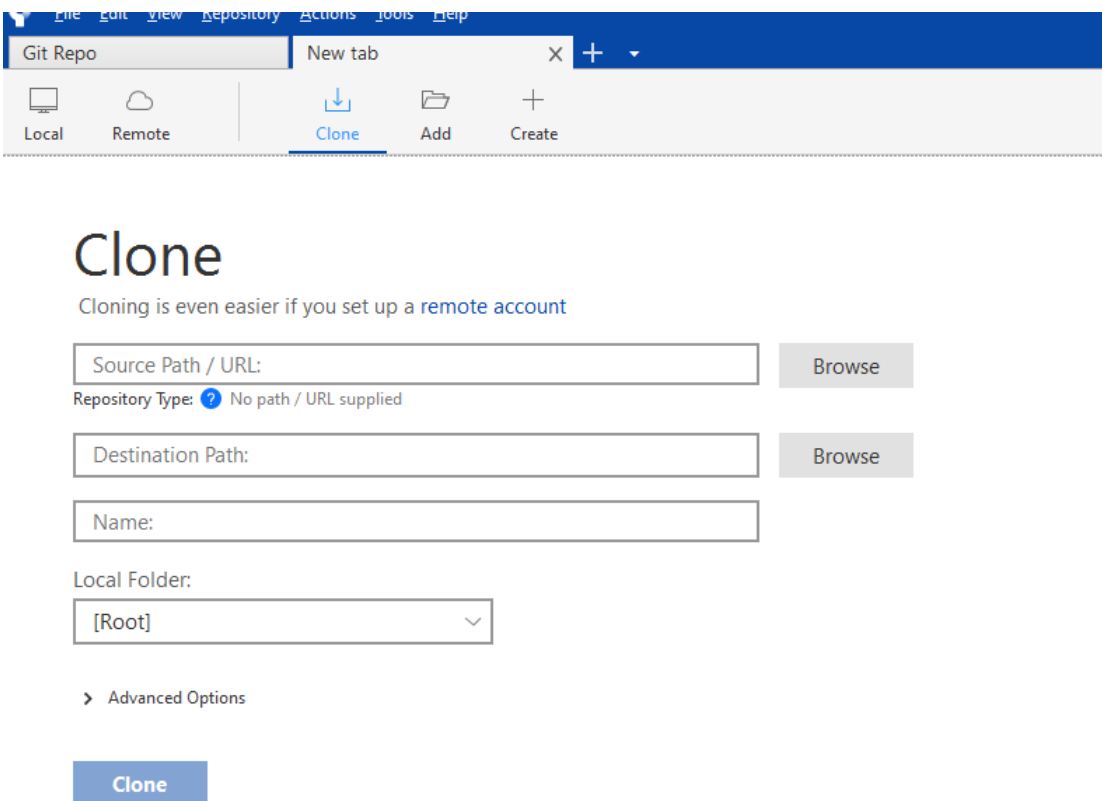
Connecting to the Git Repo:

To simplify the process of connecting to the Git repo and pushing your changes, it is recommended to use a Git tool such as SourceTree (<https://www.sourcetreeapp.com/>)

There are many Git tools available, however this guide will take you through the process of connecting and interacting with the Git repo using Sourcetree.

Download the installer from <https://www.sourcetreeapp.com/> and install on your machine.

Open the application, click on the 'file' dropdown menu and select new/clone.

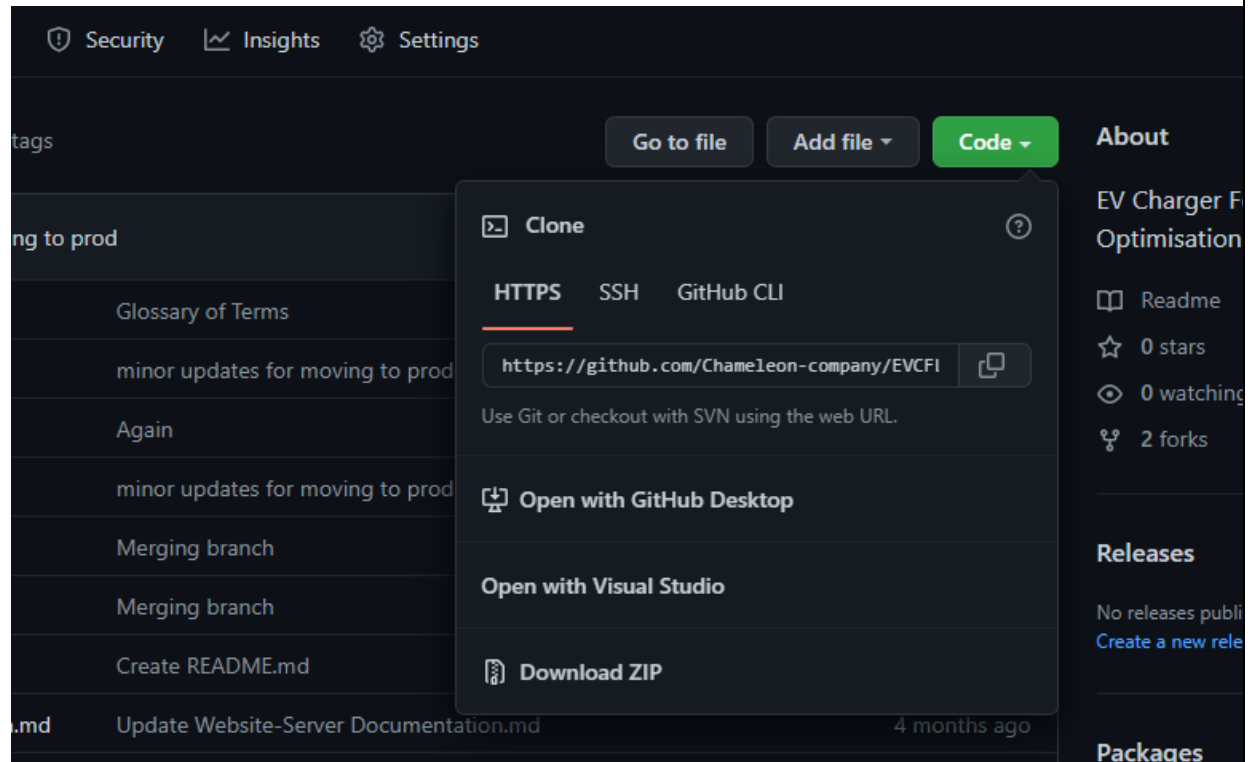


The screenshot shows the SourceTree application window with the 'Clone' dialog box open. The dialog has a title bar with 'Git Repo' and a 'New tab' button. Below the title bar are icons for 'Local', 'Remote', 'Clone', 'Add', and 'Create'. The 'Clone' button is highlighted. The main area of the dialog is titled 'Clone' and contains the following fields and buttons:

- 'Source Path / URL:' text box with a 'Browse' button to its right.
- 'Repository Type:' label with a question mark icon and the text 'No path / URL supplied'.
- 'Destination Path:' text box with a 'Browse' button to its right.
- 'Name:' text box.
- 'Local Folder:' label with a dropdown menu showing '[Root]'.
- 'Advanced Options' link with a right-pointing arrow.
- 'Clone' button at the bottom.

Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFLO)

Navigate to the github repo (<https://github.com/Chameleon-company/EVCFLO>) and obtain the link to the EVCFLO git repo under the 'code' dropdown. Copy the HTTPS path.



The screenshot shows the GitHub repository page for 'Chameleon-company/EVCFLO'. The 'Code' dropdown menu is open, displaying the following options:

- Clone** (with a question mark icon)
- HTTPS** (selected, with a red underline)
- SSH**
- GitHub CLI**

The HTTPS URL is displayed in a text box: `https://github.com/Chameleon-company/EVCFI`. Below the URL, it says: 'Use Git or checkout with SVN using the web URL.' Below the URL box, there are three more options:

- Open with GitHub Desktop**
- Open with Visual Studio**
- Download ZIP**

The background of the repository page shows the file list with items like 'Glossary of Terms', 'minor updates for moving to prod', 'Again', 'minor updates for moving to prod', 'Merging branch', 'Merging branch', 'Create README.md', and 'Update Website-Server Documentation.md'.

Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFLO)

Paste the link into the first text box, it should be verified automatically with the note underneath saying it is a git repo.

You will need to make a designated working folder on your machine to store the source code locally. This is where Sourcetree will track the changes you make and use this to push back to the git repo. Browse to this folder and select it in the second text box.

Name your project and select 'Clone' – this will begin copying all the files from the git repo to your machine.

Clone

Cloning is even easier if you set up a [remote account](#)

Repository Type:  This is a Git repository

Local Folder:

> Advanced Options

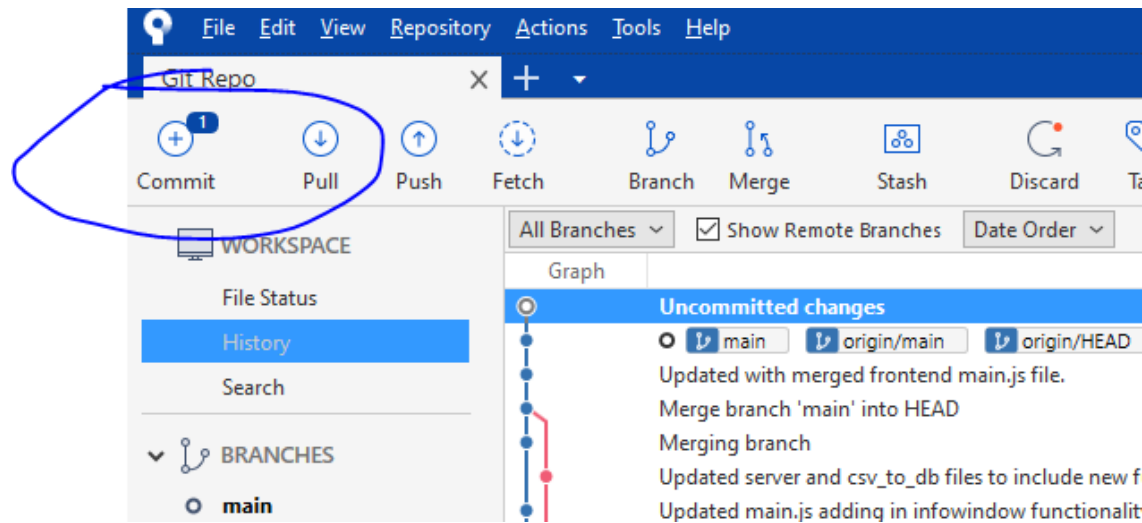
Pushing updates to the Git repository:

Updated files will be represented in Sourcetree as unstaged files. To push these changes first select the files you want to push (generally you would select 'Stage All')

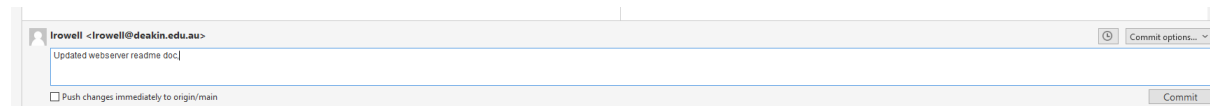


Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFO)

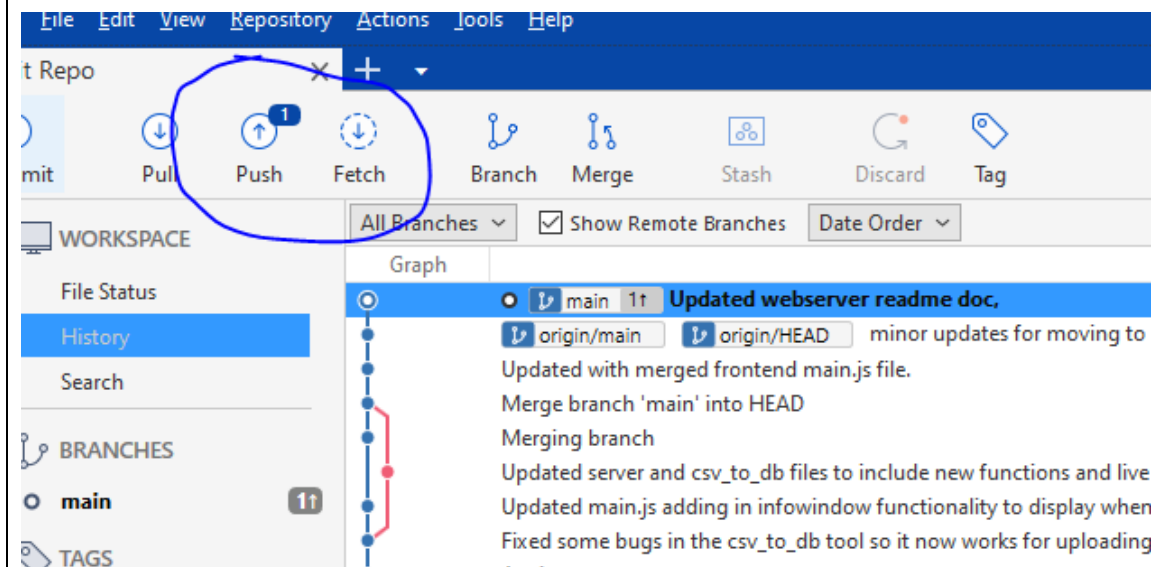
Select Commit.



Add a comment regarding the changes made. It is good practice to make a push every time a change to one part of the repo is made to avoid making multiple changes in a single push. This makes it easier to rollback if there is an error. Click 'Commit'.

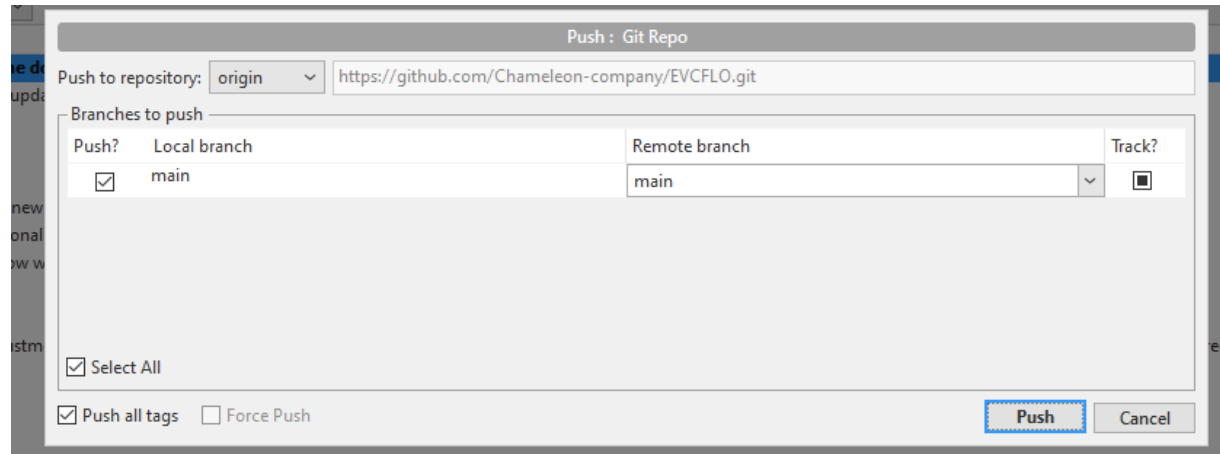


You will now be able to select 'Push' – click it to start the transfer.



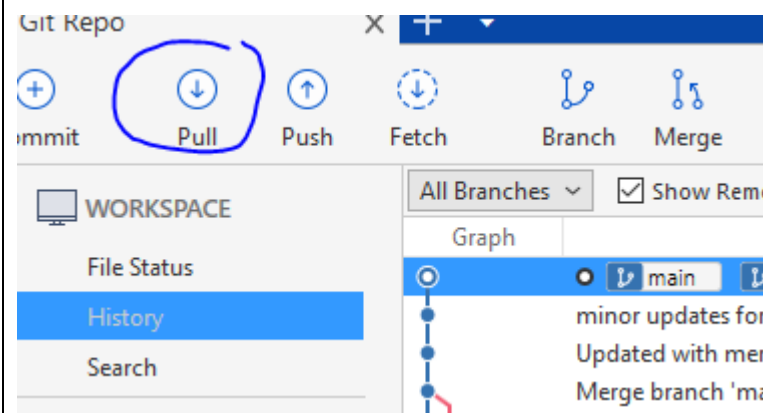
Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFO)

There is no need to create branches for the size of this project, so ensure you are pushing to the main branch. Click 'Push'. Once this completes the updates will be live on the Github repo.



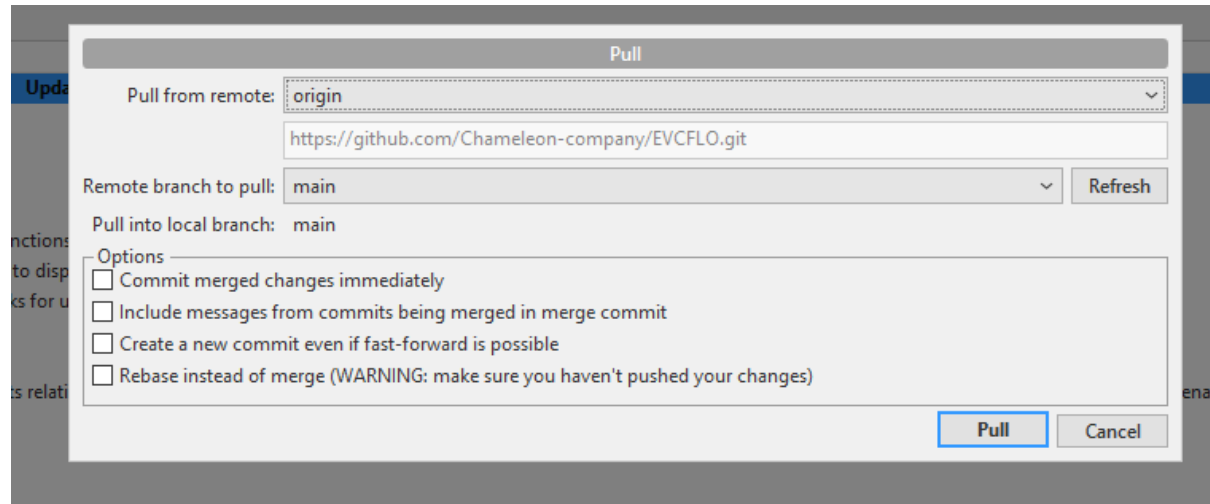
Pulling Updates from the Repository:

To sync the files on your local machine with the latest from the Github repo and pull in any changes that other contributors have made, click on the 'Pull' icon.



Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFLO)

Select the main branch again, and click 'Pull'.



Pull

Pull from remote: origin
https://github.com/Chameleon-company/EVCFLO.git

Remote branch to pull: main Refresh

Pull into local branch: main

Options

- ☐ Commit merged changes immediately
- ☐ Include messages from commits being merged in merge commit
- ☐ Create a new commit even if fast-forward is possible
- ☐ Rebase instead of merge (WARNING: make sure you haven't pushed your changes)

Pull Cancel

Web Server API:

The web server providing access to the charger station database is written in Python using the Flask framework, and hosted on the Google Cloud Platform in a docker container. Source code on GitHub does not include the required `/server/config/env.py` file for security reasons.

Public URL:

<https://server-e657pcuziq-ts.a.run.app>

Current Routes:

[/]

Method: GET

Returns: 'The server is running...' if the app is running.

[/api/get_chargers]

Method: GET

Returns: All of the chargerstation objects from the database as a json object.

[/api/get_bounds]

Method: GET

Returns: All of the bounding box objects from the database as a json object.

[/api/get_suggested]

Method: GET

Returns: All of the suggested points objects from the database as a json object.

[/api/search_chargers/ <lat>/<lng>/<rad>]

Method: GET

Returns: Chargerstation objects from the database within km from geo location with latitude and longitude as a json object.

[/api/add_station]

Method: POST

Returns: 201 if successful, 400 if failed.

Note: Requires body to contain at least valid values for 'name', 'latitude', and 'longitude' parameters.

The complete model for a charger station (parameter names and types) can be found in `/models/ChargerStation.py`



Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFLO)

[/api/add_bounding]

Method: POST

Returns: 201 if successful, 400 if failed.

Note: Requires body to contain at least valid values for 'north', 'south', 'east', 'west', 'source', and 'source_date' parameters. The complete model for a bounding box (parameter names and types) can be found in /models/BoundingBox.py

[/api/add_suggested]

Method: POST

Returns: 201 if successful, 400 if failed.

Note: Requires body to contain at least valid values for 'latitude', 'longitude', 'total_plugs', and 'suitability_score' parameters. The complete model for a charger station (parameter names and types) can be found in /models/SuggestedStation.py

[/api/db_populate]

Method: POST

Returns: 201 is successful, 400 if failed.

Note: This simply adds 5 demo charger station entries to the database and used during development. This should be removed from production.



Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFLO)

Links:

Trello Board:

<https://trello.com/b/10idYvBh/evcflo-t2-2022>

Chameleon EVCFLO GitHub:

<https://github.com/Chameleon-company/EVCFLO>

EVCFLO Website:

https://storage.googleapis.com/evcflo_static_frontend/index.html

EVCFLO Web server (API):

<https://server-e657pcuziq-ts.a.run.app/>

Chameleon Melbourne Open Data Playground GitHub (Reference Resource):

<https://github.com/Chameleon-company/MOP>

The Following Google Cloud Platform links require that you are logged into Google using your Deakin email address and credentials, and have been granted access from the Deakin admin team.

Google Cloud Platform (GCP):

<https://console.cloud.google.com/welcome?project=sit-22t1-ev-adoption-t-a4efab5&authuser=0&orgonly=true&supportedpurview=organizationId>

GCP Cloud Storage (Front end)

<https://console.cloud.google.com/storage/browser?authuser=0&orgonly=true&project=sit-22t1-ev-adoption-t-a4efab5&supportedpurview=organizationId&prefix=&forceOnObjectsSortingFiltering=false&pli=1>

GCP Cloud Run (Server)

<https://console.cloud.google.com/run?referrer=search&authuser=0&orgonly=true&project=sit-22t1-ev-adoption-t-a4efab5&supportedpurview=organizationId&pli=1>



Chameleon Electric Vehicle Charger Forecasting and Location Optimization (EVCFLO)

Contacts:

Company director:

Valeh Moghaddam (valeh.moghaddam@deakin.edu.au)

Google Cloud Platform Account Admin:

Nghia Dang (Ngh_adm@deakin.edu.au)

Blac_adm@deakin.edu.au

Justin.rough@deakin.edu.au

GitHub Admin:

Leigh Rowell (lrowell@deakin.edu.au)

Ezekiel Griffin (ekgriffin@deakin.edu.au)