WINYAMA
DIGITAL SOLUTIONS

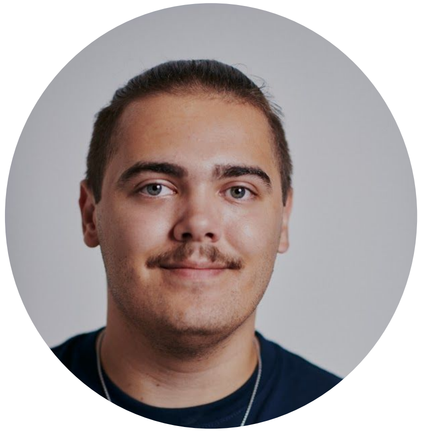
Geogeeks Speaker Night

Serverless Imagery Processing with OpenDroneMap and AWS

About Us

Yukio Chaplin

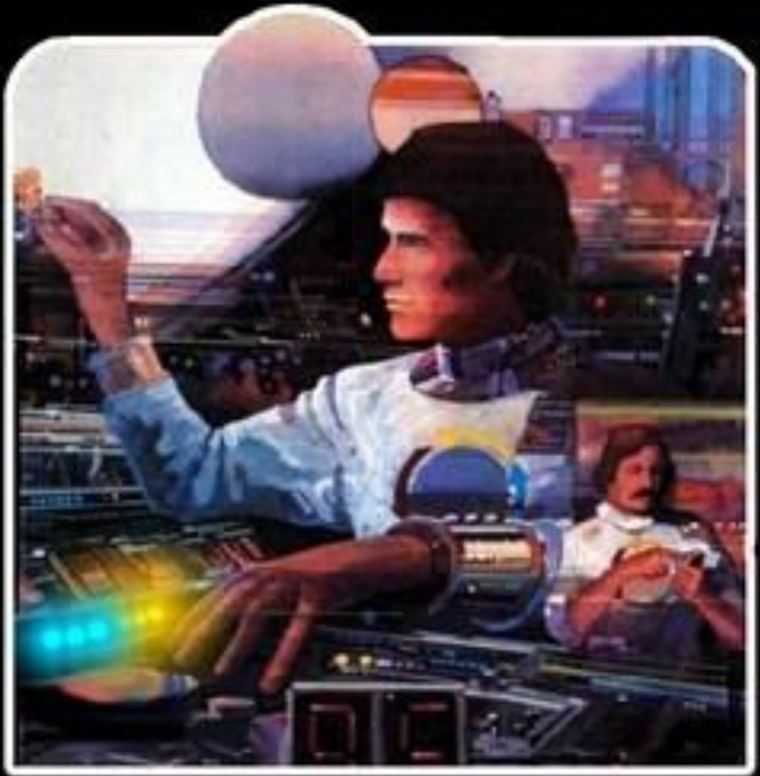
- **Background**: Conservation and Wildlife biology
- **Career**: Previous work as an Environmental Undergraduate and now as a GIS Intern.
- **Future**: Background in Environmental Science combined with GIS is a complementary skill that only looks to strengthen my future at Winyama.



River Bali

- **Background**: Aviation / Drone operations
- **Career**: Winyama is my first step in starting my career, previous work in fast food and retail
- **Future**: Intend on moving into a permanent position with Winyama after Internship as a GIS analyst/Drone's Specialist

THE TWO STATES OF EVERY PROGRAMMER



I AM A GOD.

I HAVE NO IDEA WHAT I'M DOING.

Solution Introduction

OpenDroneMap (ODM) is an open source toolkit for processing imagery. It can take images (such as photos captured by a drone) as inputs and produce a variety of georeferenced assets as outputs, such as maps and 3D models.

Traditionally, ODM is a command-line toolkit, which you would download and install and then interact with through a series of commands.

The OpenDroneMap Organisation has also released some other products such as WebODM, which provides a web GUI, visualisation, storage and data analysis functionality.

The full suite of OpenDroneMap products can be found on their website:
https://www.opendronemap.org/

ODM and Imagery Processing Challenges

ODM software can be installed and run locally, however it has high compute requirements for large processing jobs that aren't accessible for most users.

Maintaining and running a dedicated server for ODM (or other products such as WebODM) can also introduce significant technical overhead and high infrastructure costs that aren't feasible for smaller organisations.

Software as a service (SaaS) solutions are available (DroneDeploy, Pix4D etc…) - however their subscription pricing models don't meet the once-off/project based imagery processing requirements of some organisations.

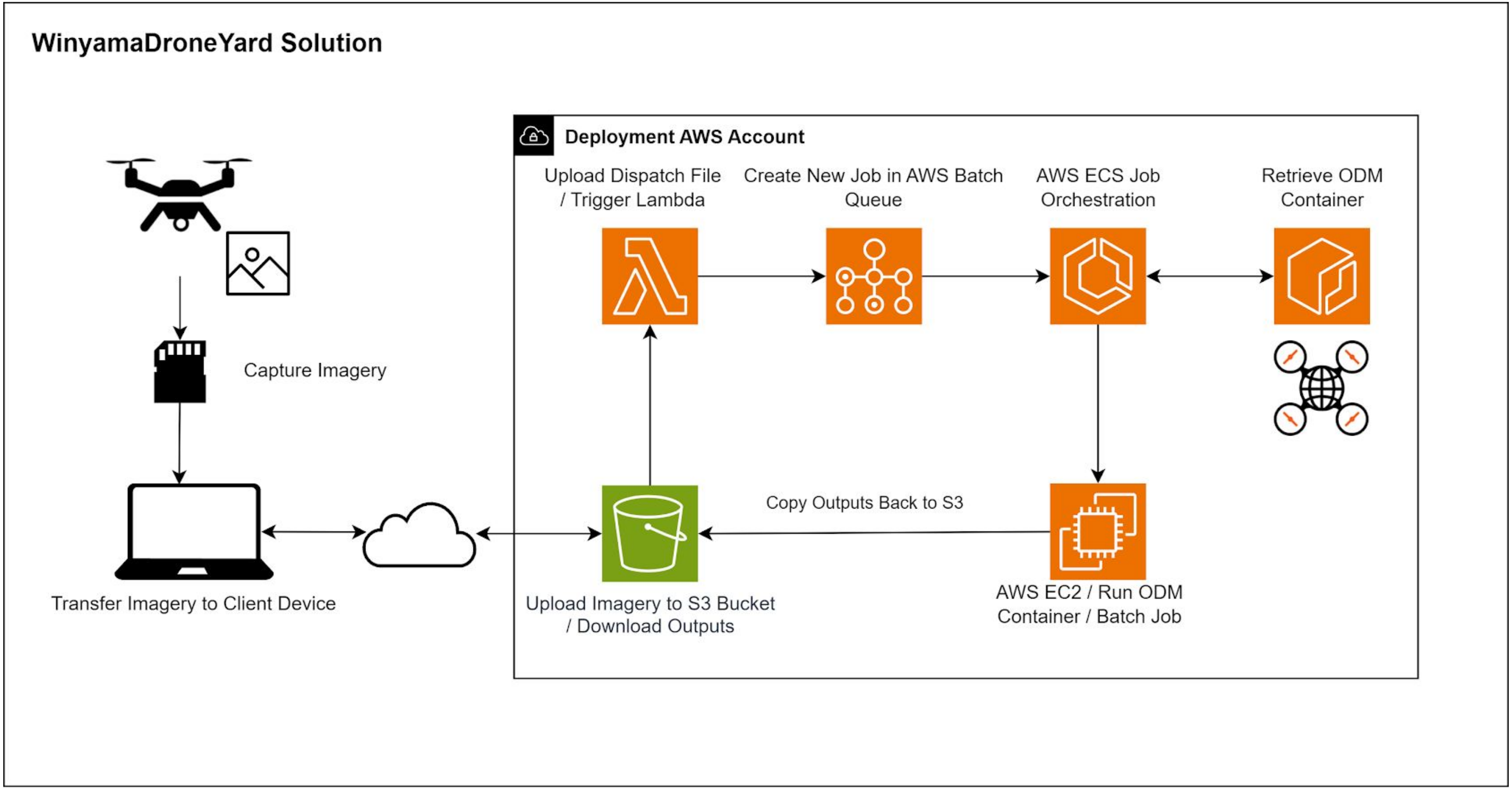
Why WinyamaDroneYard?

WinyamaDroneYard is **serverless**, meaning that although servers are certainly involved - they are only created when required, and there is no maintenance or direct interaction required with these servers.

By using AWS and a serverless approach, WinyamaDroneYard allows the user to pay only for what they actually use - specifically the time where imagery processing is required and actively running, as well as very low service consumption costs for storage and job orchestration.

The solution can be deployed to AWS using the simplest of computers, and by anyone with just baseline command-line knowledge.

When asked to draw a flowchart of my code


WinyamaDroneYard Solution
Deployment AWS Account
Upload Dispatch File / Trigger Lambda
Create New Job in AWS Batch Queue
AWS ECS Job Orchestration
Retrieve ODM Container
Capture Imagery
Copy Outputs Back to S3
Transfer Imagery to Client Device
Upload Imagery to S3 Bucket / Download Outputs
AWS EC2 / Run ODM Container / Batch Job

Before you Start

Prerequisites for deploying the WinyamaDroneYard solution:

1. **AWS Account** - you will need an AWS account that you have access to deploy the solution resources into.

2. **Administrator Privileges** - you will need elevated access to your local computer/development environment in order to install the necessary software required to deploy the solution.

3. **Internet Access** - the deployment steps will require connectivity to the public internet.

***** A Windows operating environment is recommended, but not strictly required.

Development Environment Prerequisites (Software)

WINYAMA DIGITAL SOLUTIONS

1. Download and install **Git**

2. Download and install **Node/Node Package Manager (NPM)**

3. Download and install **Docker Desktop**

4. Download and install the **AWS CLI (Command Line Interface)**

5. Download and install the **AWS CDK Toolkit**

6. Configure your AWS CLI credentials

    6.1. Personal AWS accounts requirements - Create an IAM user/retrieve CLI credentials from the IAM service in AWS console.

    ⭐**Our recommendation is to create 2 different IAM users, one with AWS Administrator access to deploy the solution, and the other with just S3 - specifically for interacting with your deployed S3 bucket(s).**

Deployment Steps

1. Configure AWS credentials (**aws configure**) *** Administrator IAM account**

2. Clone WinyamaDroneYard repository from GitHub (**git clone**)

3. Navigate to root of cloned repository (**cd WinyamaDroneYard**)

4. Install Node packages (**npm install**)

5. Bootstrap your AWS account for CDK (**cdk bootstrap**)

6. Ensure Docker is running (open Docker Desktop program)

7. Deploy resources to AWS (**cdk deploy --require-approval never**)

Deployment steps can also be found in the README file of the repository.


Waiting for Deployment to Finish Successfully

AWS Configure (IAM User)

- For non organisational use, use an AWS IAM USER account
- Run the AWS Configure command and input the access key ID + secret key ID
- Set the region - **ap-southeast-2**

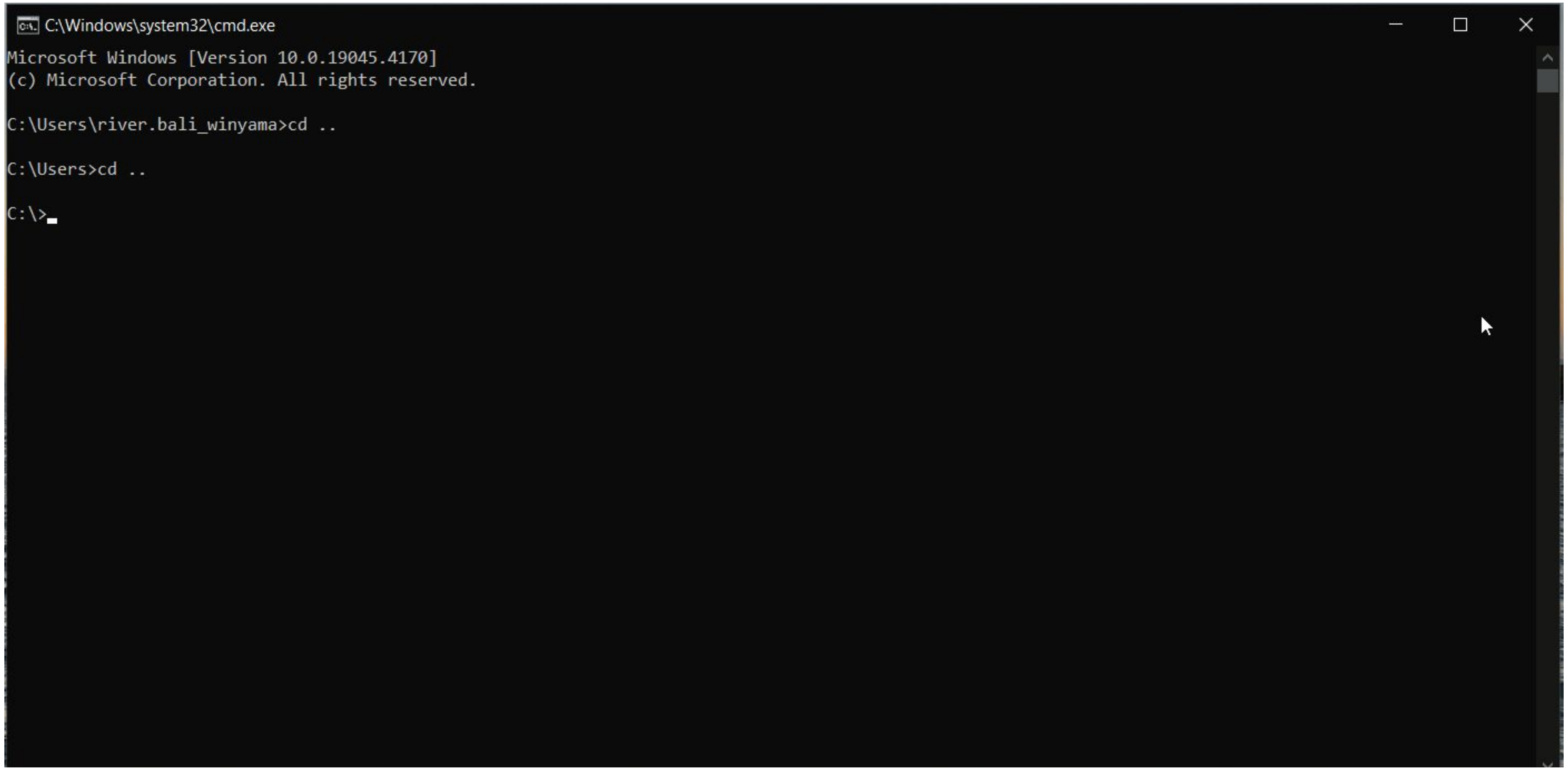
C:\Windows\system32\cmd.exe

```
C:\Users\yukio.chaplin_winyam>aws configure
AWS Access Key ID [****************WS6U]: AKIAYS2NWQAWACORD6FV
AWS Secret Access Key [****************kw/i]: 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
Default region name [ap-southeast-2]:
Default output format [None]:

C:\Users\yukio.chaplin_winyam>aws configure list
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                  default              env    ['AWS_PROFILE', 'AWS_DEFAULT_PROFILE']
access_key     ****************D6FV shared-credentials-file
secret_key     ****************FozU shared-credentials-file
    region           ap-southeast-2      config-file    ~/.aws/config
```

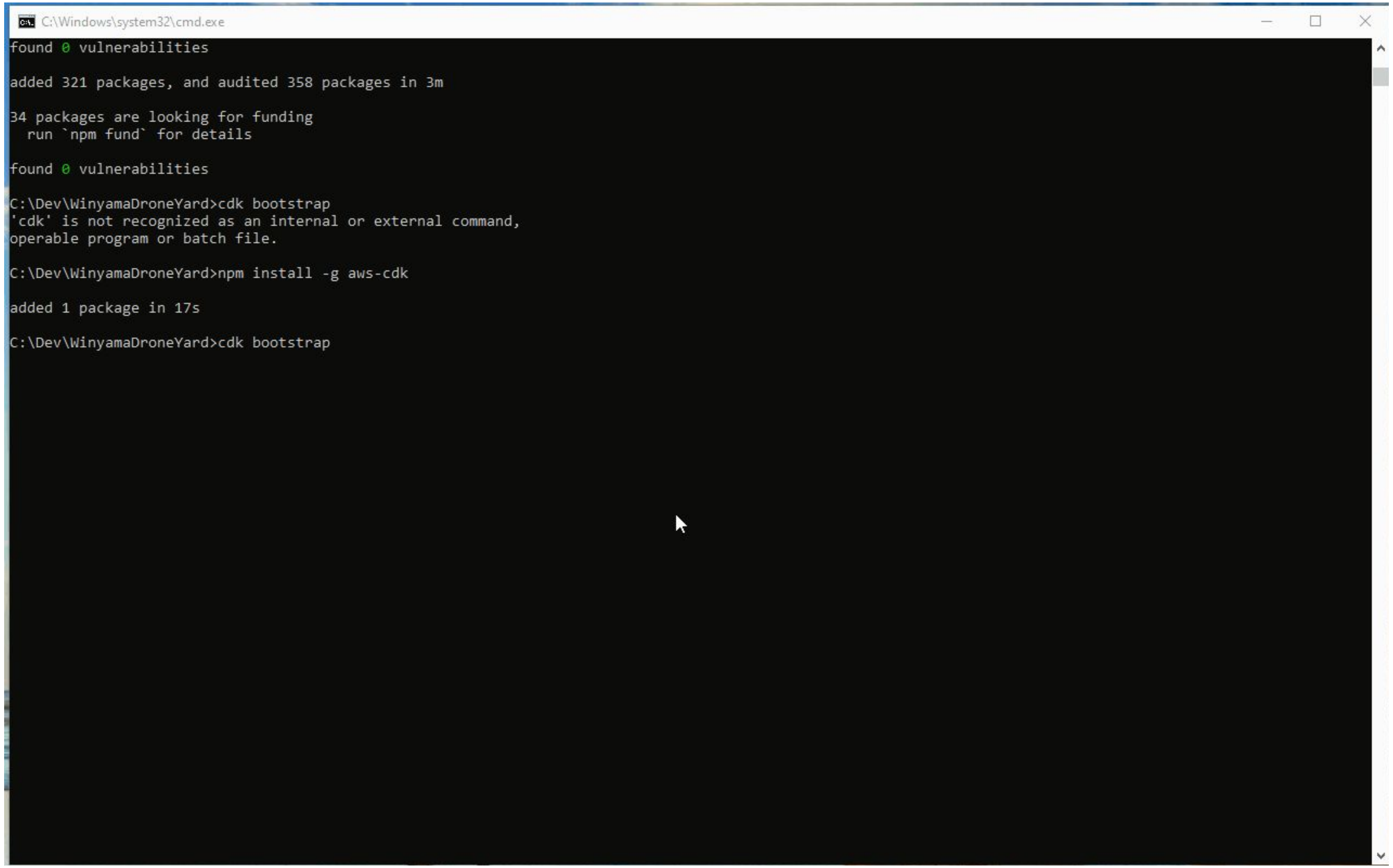
<u>Deployment Steps 2 (GIT CLONE)</u>

- Navigate in your command-line client to where you want to clone the repository Dev.
- Input the command (**<u>git clone</u>**) along with the url for the Winyama Drone Yard repository.



CDK Bootstrap (IAM User)

- Open the WinyamaDroneYard folder (cd winyamadroneyard) and run **CDK Bootstrap**



CDK Deploy (IAM User)

- Again within WinyamaDroneYard run **CDK Deploy**

C:\Windows\system32\cmd.exe

```
#2 DONE 4.9s

#4 [internal] load .dockerignore
#4 transferring context: 2B 0.0s done
#4 DONE 0.1s

#5 [1/3] FROM docker.io/opendronemap/odm:latest@sha256:ea02224aad55e6e6474879915ba2e12f341d31d28b74a9ea6ff172cad8898c0f
#5 resolve docker.io/opendronemap/odm:latest@sha256:ea02224aad55e6e6474879915ba2e12f341d31d28b74a9ea6ff172cad8898c0f
#5 ...

#6 [internal] load build context
#6 transferring context: 1.50kB 0.2s done
#6 DONE 0.3s

#5 [1/3] FROM docker.io/opendronemap/odm:latest@sha256:ea02224aad55e6e6474879915ba2e12f341d31d28b74a9ea6ff172cad8898c0f
#5 resolve docker.io/opendronemap/odm:latest@sha256:ea02224aad55e6e6474879915ba2e12f341d31d28b74a9ea6ff172cad8898c0f 0.3s done
#5 sha256:edc6db0df4120ebc83a69db4a378f5f15bea7e6f7193f547fa4a771a7dd6b63a 3.29kB / 3.29kB done
#5 sha256:ea02224aad55e6e6474879915ba2e12f341d31d28b74a9ea6ff172cad8898c0f 1.61kB / 1.61kB done
#5 sha256:006907b089d115552a4d70ad64af64138a1af5e77e3bb15e60eb3171fa2dd0a5 1.44kB / 1.44kB done
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 0B / 31.70MB 0.2s
#5 sha256:765faef40afef964ebf7a392908c064efd533f949159a8688a53bffcb963f756 0B / 94B 0.3s
#5 sha256:7c03d5650373b717b709a54c1d85c884ee7a1504050d73c179060a76ecb2b6b4 0B / 153.00MB 0.5s
#5 sha256:765faef40afef964ebf7a392908c064efd533f949159a8688a53bffcb963f756 94B / 94B 0.9s done
#5 sha256:894059b69676333e4094767f281f263ead267b56ebee124d3493fe9441f3e589 0B / 229.75MB 1.0s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 2.10MB / 31.70MB 1.2s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 4.19MB / 31.70MB 3.4s
#5 sha256:7c03d5650373b717b709a54c1d85c884ee7a1504050d73c179060a76ecb2b6b4 3.15MB / 153.00MB 5.6s
#5 sha256:894059b69676333e4094767f281f263ead267b56ebee124d3493fe9441f3e589 2.10MB / 229.75MB 6.5s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 6.29MB / 31.70MB 7.7s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 8.39MB / 31.70MB 9.6s
#5 sha256:7c03d5650373b717b709a54c1d85c884ee7a1504050d73c179060a76ecb2b6b4 7.34MB / 153.00MB 10.9s
#5 sha256:894059b69676333e4094767f281f263ead267b56ebee124d3493fe9441f3e589 7.34MB / 229.75MB 11.7s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 10.49MB / 31.70MB 12.2s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 12.58MB / 31.70MB 13.6s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 14.68MB / 31.70MB 15.8s
#5 sha256:7c03d5650373b717b709a54c1d85c884ee7a1504050d73c179060a76ecb2b6b4 12.58MB / 153.00MB 16.1s
#5 sha256:894059b69676333e4094767f281f263ead267b56ebee124d3493fe9441f3e589 12.58MB / 229.75MB 16.8s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 16.78MB / 31.70MB 18.1s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 18.87MB / 31.70MB 20.7s
#5 sha256:7c03d5650373b717b709a54c1d85c884ee7a1504050d73c179060a76ecb2b6b4 17.83MB / 153.00MB 21.2s
#5 sha256:894059b69676333e4094767f281f263ead267b56ebee124d3493fe9441f3e589 17.83MB / 229.75MB 22.0s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 20.97MB / 31.70MB 24.9s
#5 sha256:7c03d5650373b717b709a54c1d85c884ee7a1504050d73c179060a76ecb2b6b4 18.87MB / 153.00MB 26.3s
#5 sha256:894059b69676333e4094767f281f263ead267b56ebee124d3493fe9441f3e589 19.92MB / 229.75MB 27.7s
#5 sha256:6f172cdbcbefcebd1e06708f01ce8850613686821fc603db67efc9832d59ae35 23.07MB / 31.70MB 29.5s
#5 sha256:7c03d5650373b717b709a54c1d85c884ee7a1504050d73c179060a76ecb2b6b4 22.02MB / 153.00MB 31.6s
```

Running Jobs

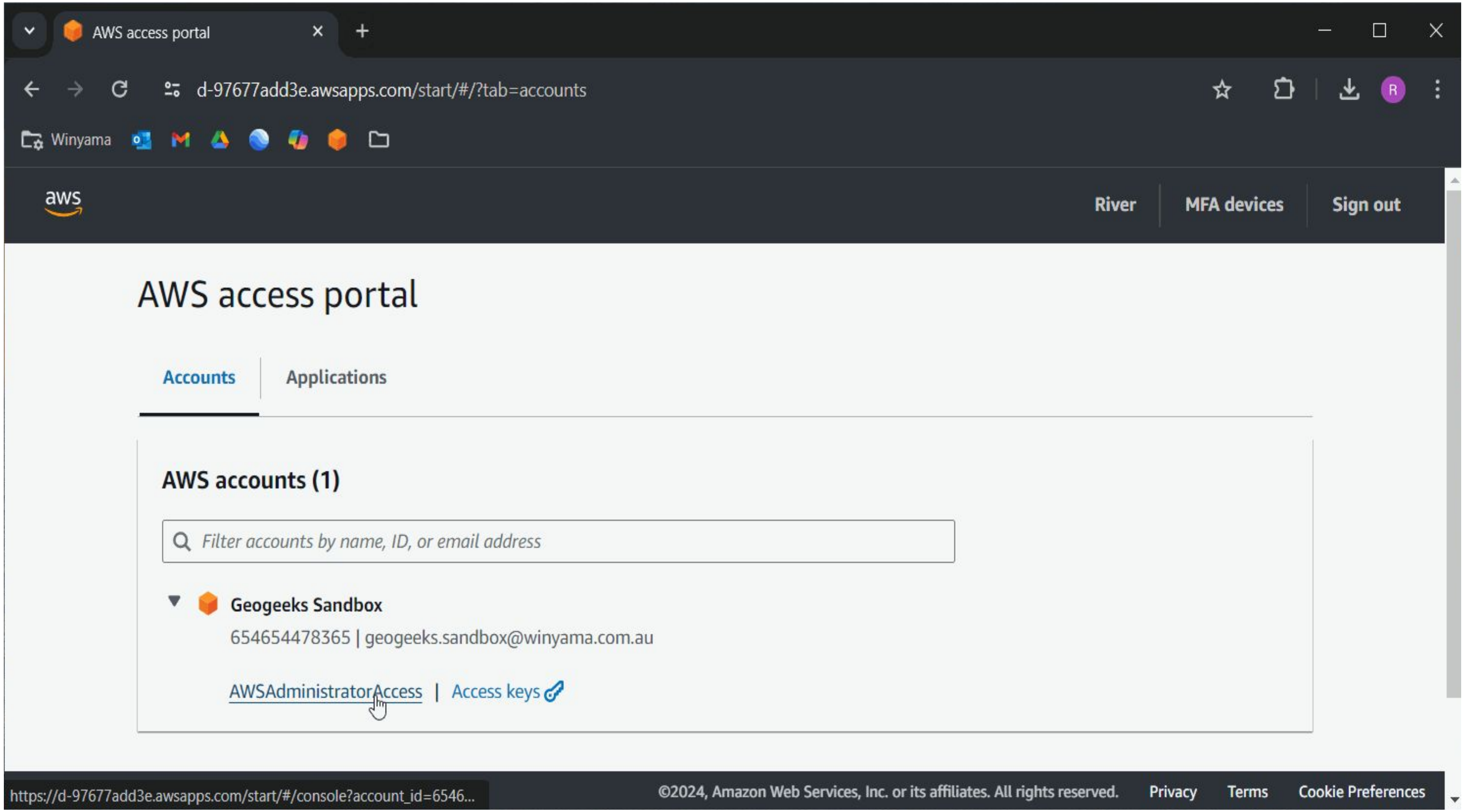
Settings.yaml

- Outlines job requirement settings for WinyamaDroneYard.
- Uploaded to the S3 bucket containing your jobs and can either be outside the folder as a default settings file or be added directly to the job folder to specify different settings for a specific job.
- Can be edited through any text editor, we suggest Visual Studio Code.

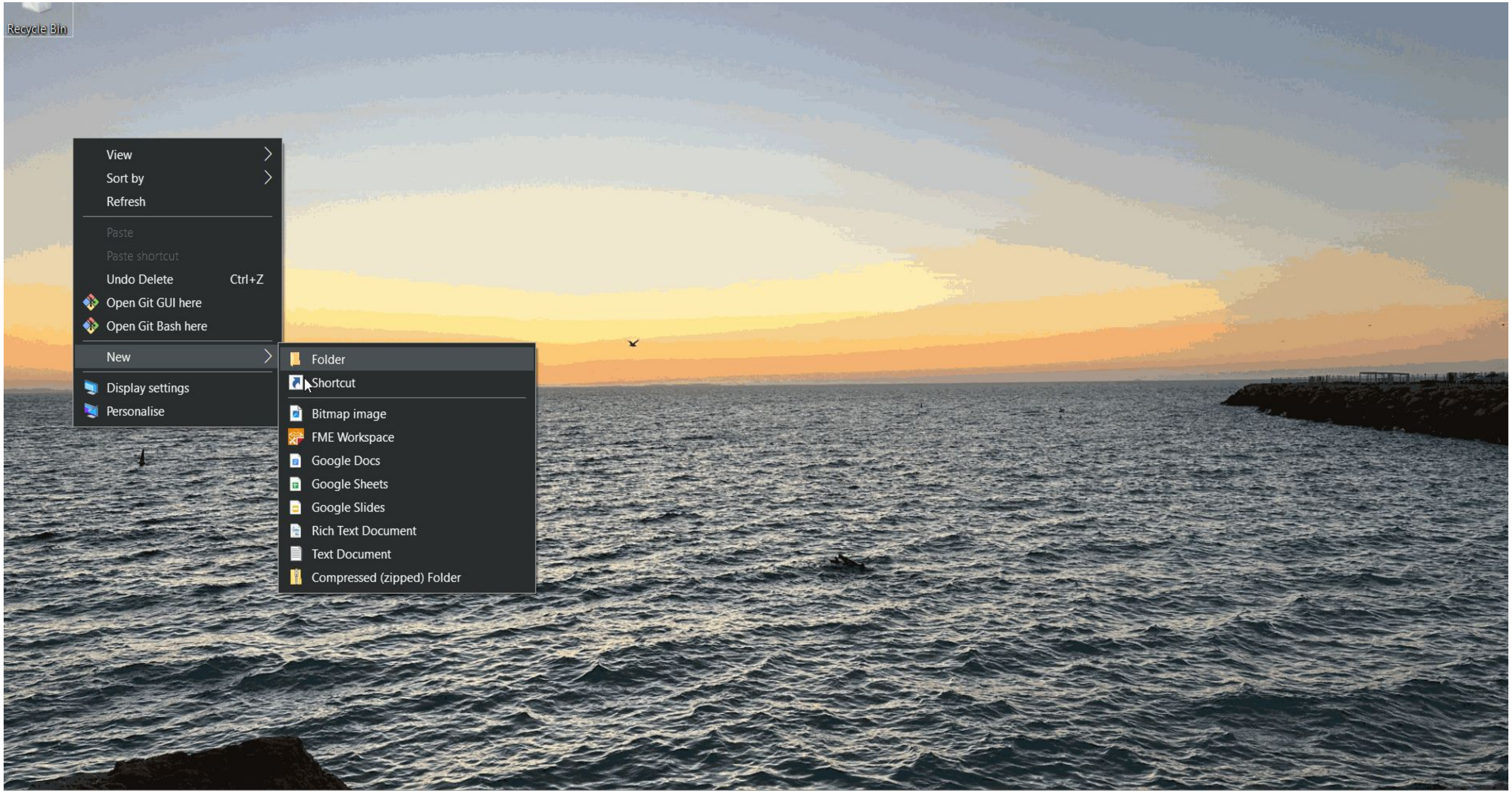

```yaml
3d_tiles: False
auto_boundary: True
build_overviews: False
camera_lens: "auto"
cog: True
crop: 3
debug: False
dem_decimation: 1
dem_euclidean_map: False
dem_gapfill_steps: 3
dem_resolution: 5
depthmap_resolution: 640
dsm: False
dtm: False
end_with: odm_postprocess
fast_orthophoto: False
feature_quality: ultra
feature_type: sift
force_gps: False
gps_accuracy: 10
ignore_gsd: False
matcher_neighbors: 0
matcher_type: flann
max_concurrency: 32
merge: all
mesh_octree_depth: 13
mesh_size: 500000
min_num_features: 10000
no_gpu: False
optimize_disk_space: False
orthophoto_compression: DEFLATE
orthophoto_cutline: True
orthophoto_kmz: True
orthophoto_no_tiled: False
orthophoto_png: True
orthophoto_resolution: 4
pc_classify: False
pc_copc: False
pc_csv: False
pc_ept: True
pc_filter: 2.5
pc_geometric: False
pc_las: False
pc_quality: ultra
pc_rectify: False
pc_sample: 0
pc_tile: False
primary_band: auto
radiometric_calibration: none
rolling_shutter: False
sfm_algorithm:  triangulation
skip_3dmodel: False
skip_band_alignment: False
skip_orthophoto: True
```

Running Jobs

Navigating AWS S3

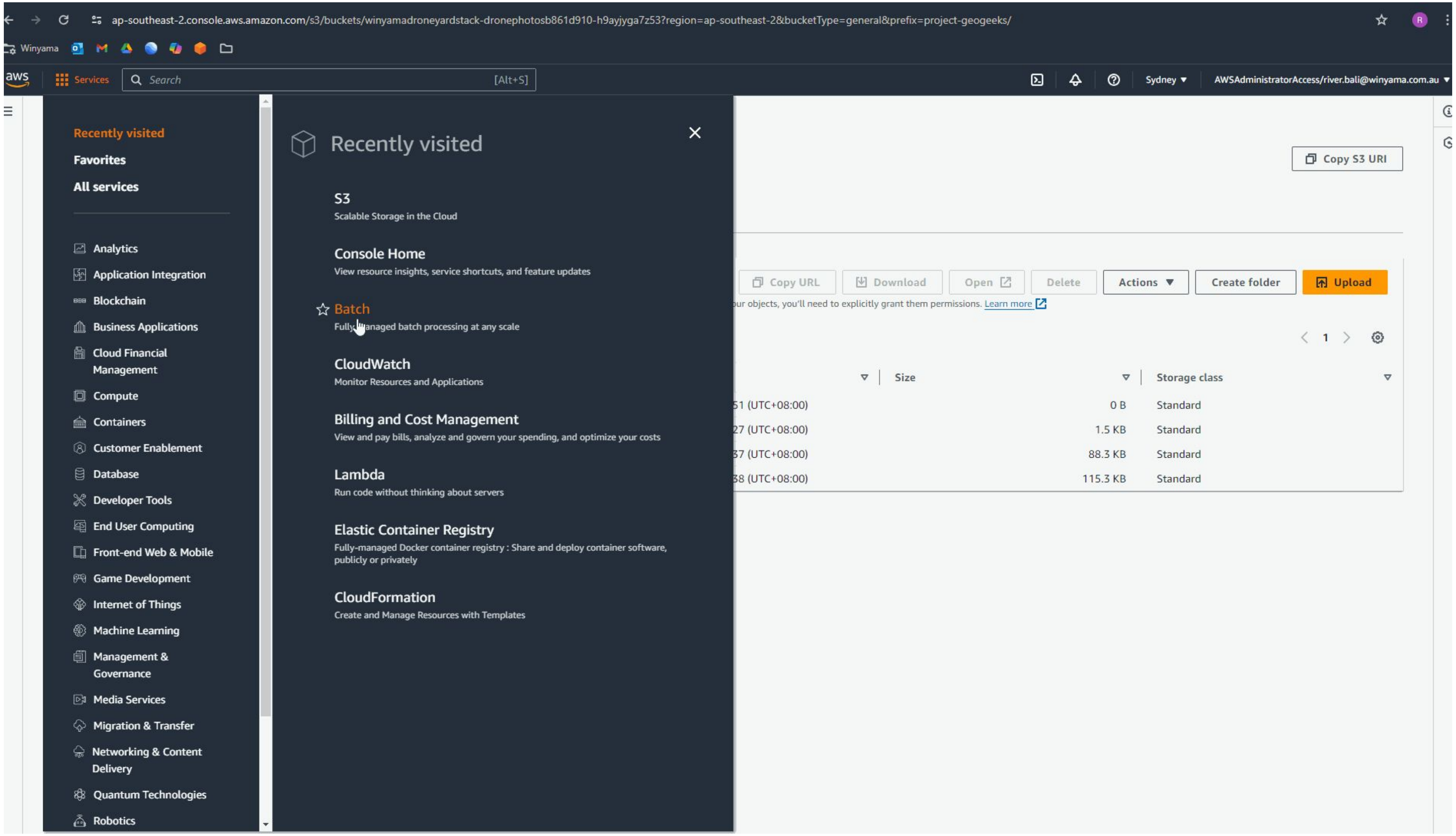


Running a Job with Dispatch File

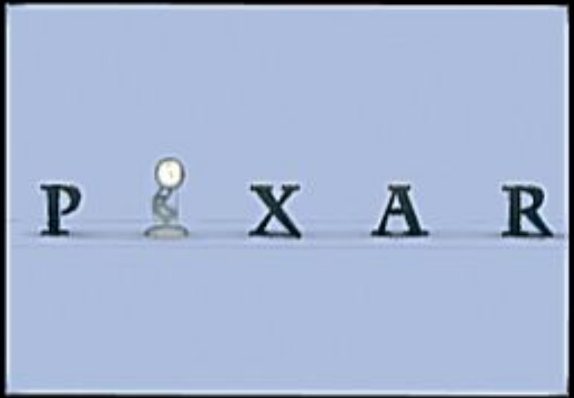


Running Jobs

Batch



What my friends think I do



What my mom thinks I do

What society thinks I do

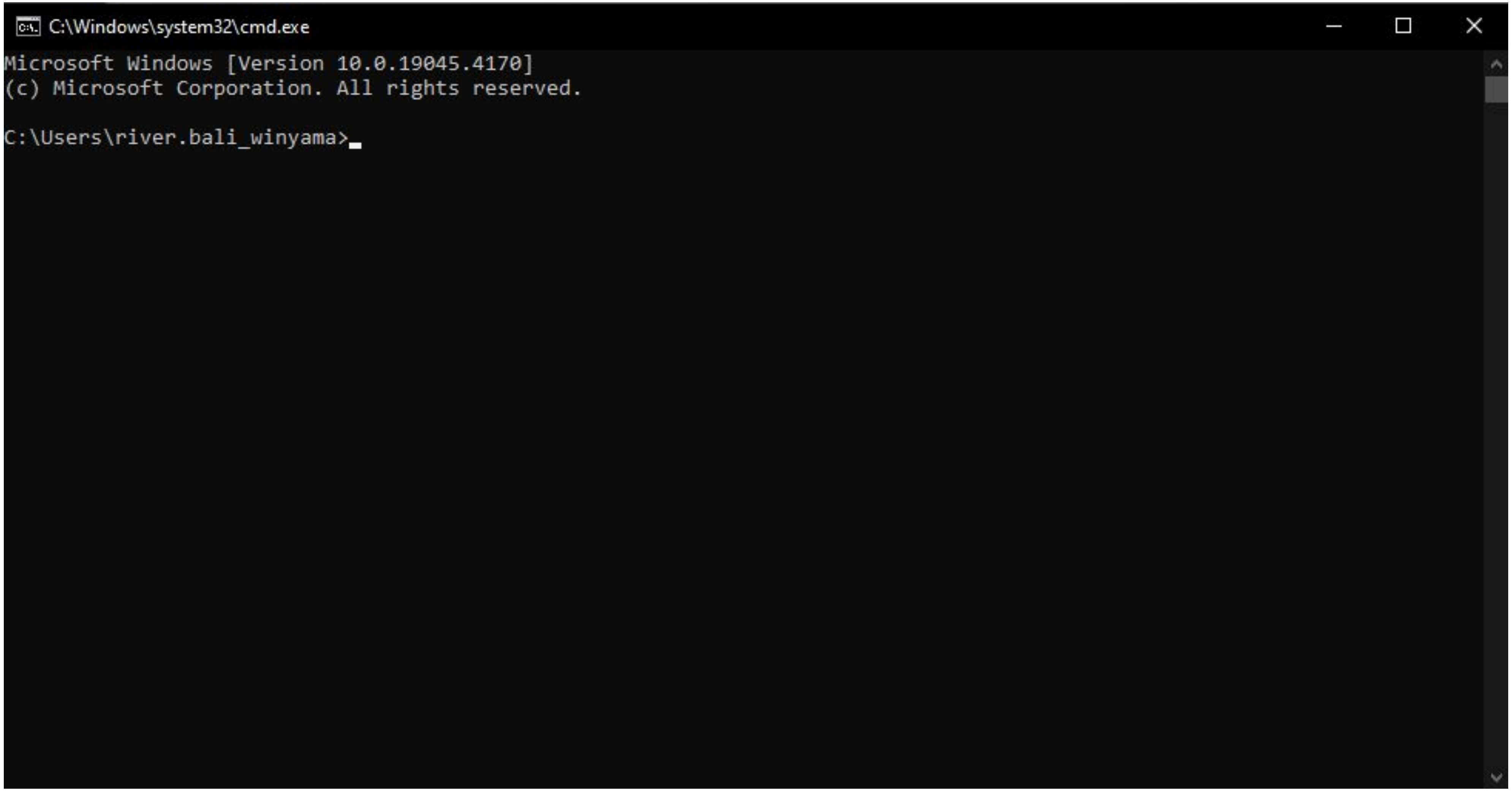
What the client think I do

What I think I do

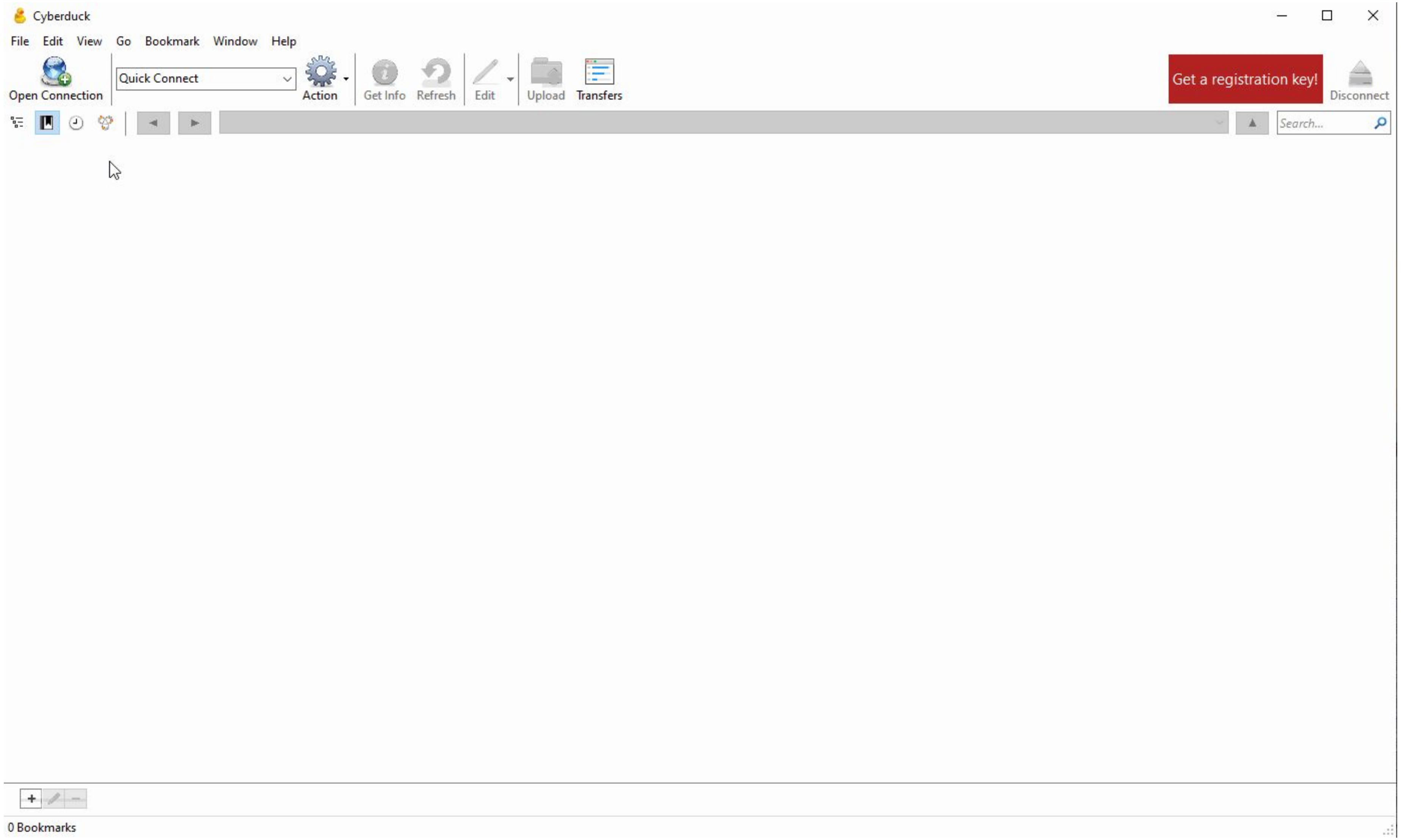
What I actually do

Extracting Outputs

(Cyberduck/recursive copy)

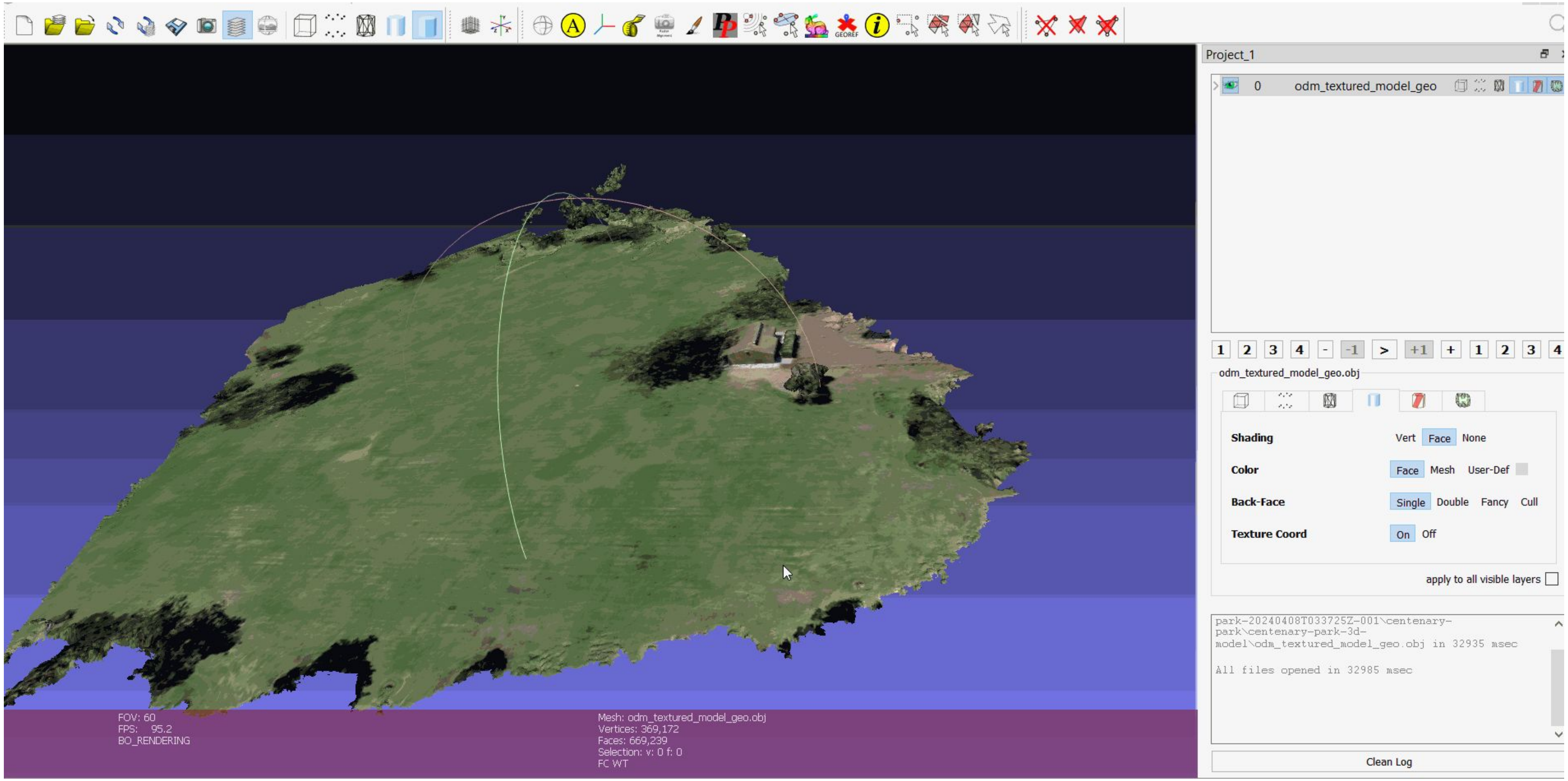


Extracting Outputs - Cyberduck



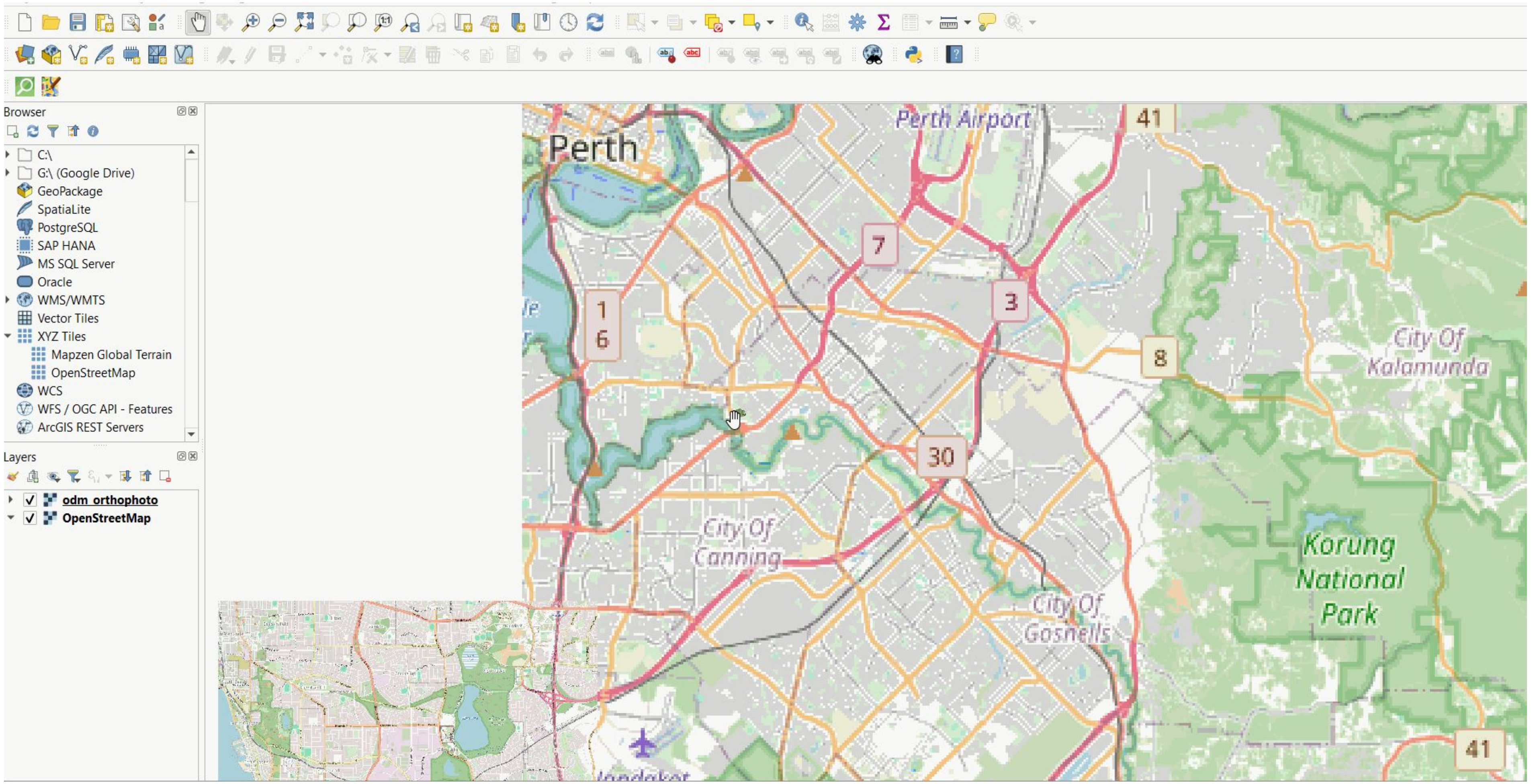
Viewing Outputs - Meshlab

OBJ in Meshlab



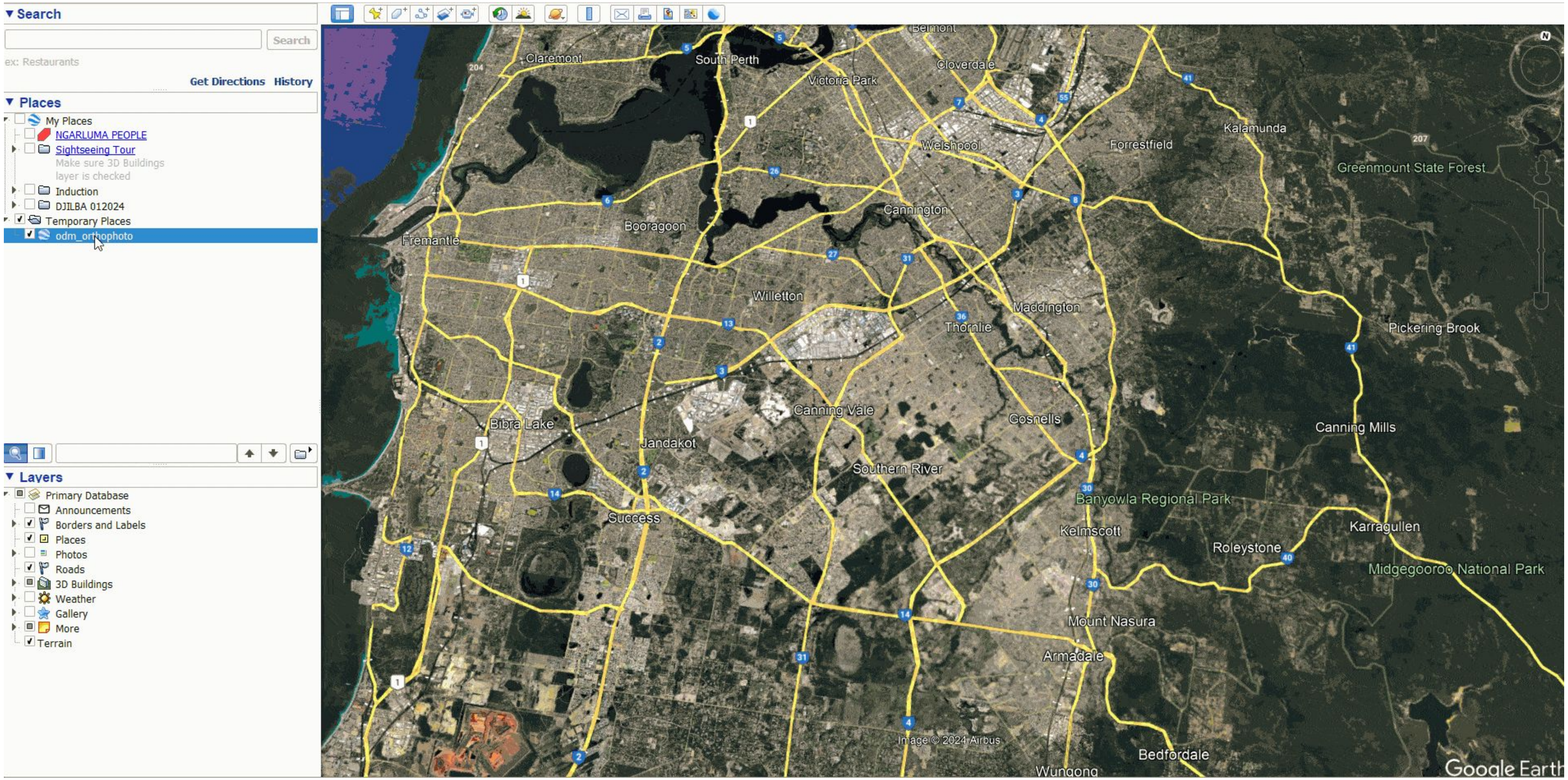
Viewing outputs through QGIS

- Viewed as a TIFF (file type)
- Output below - Centenary Park

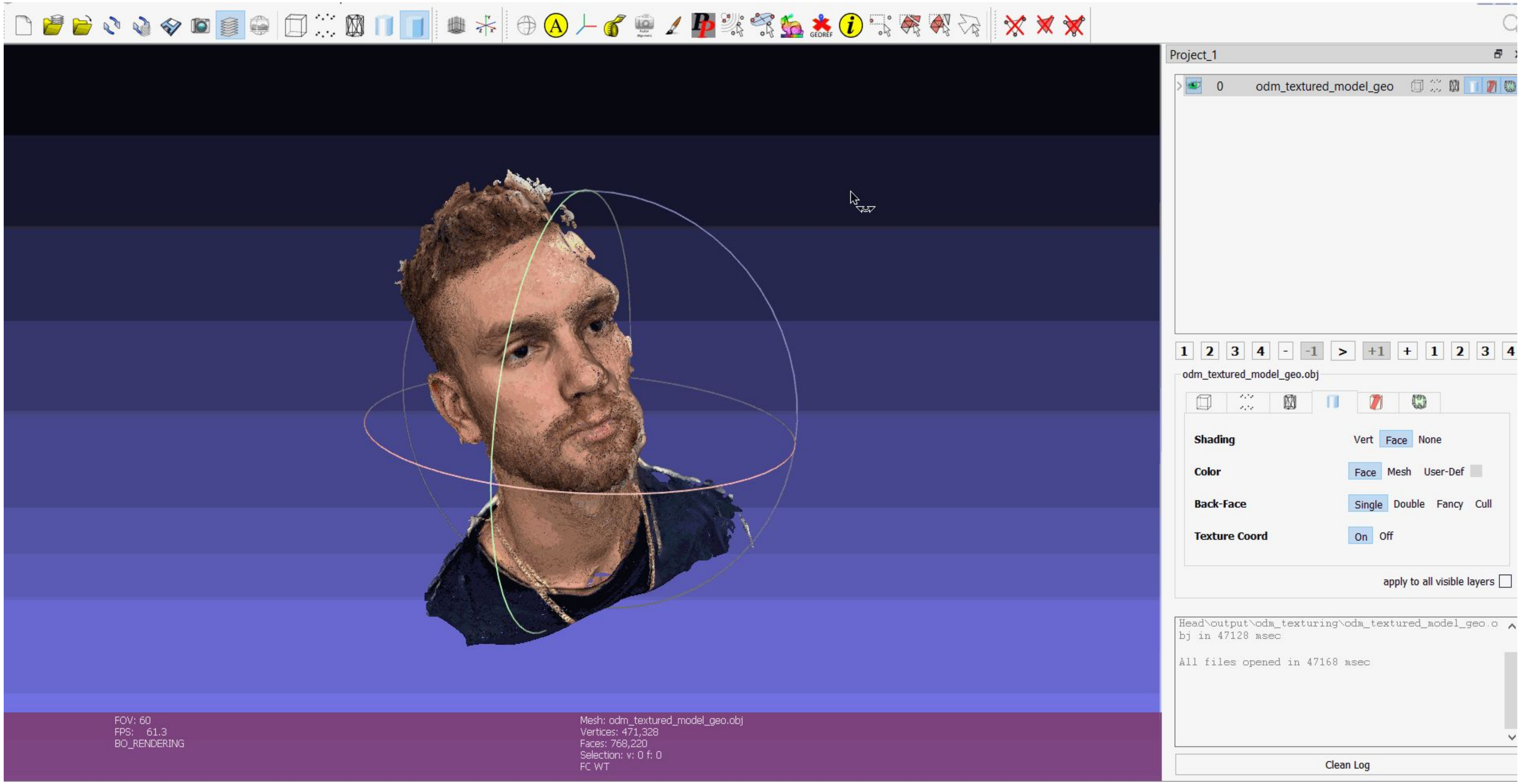


Viewing Outputs through Google Earth Pro

- Google Earth (KMZ) - Centenary Park



3D Model (Meshlab)



3D Model (Meshlab)

Didgeridoo

Hi-Res

Example

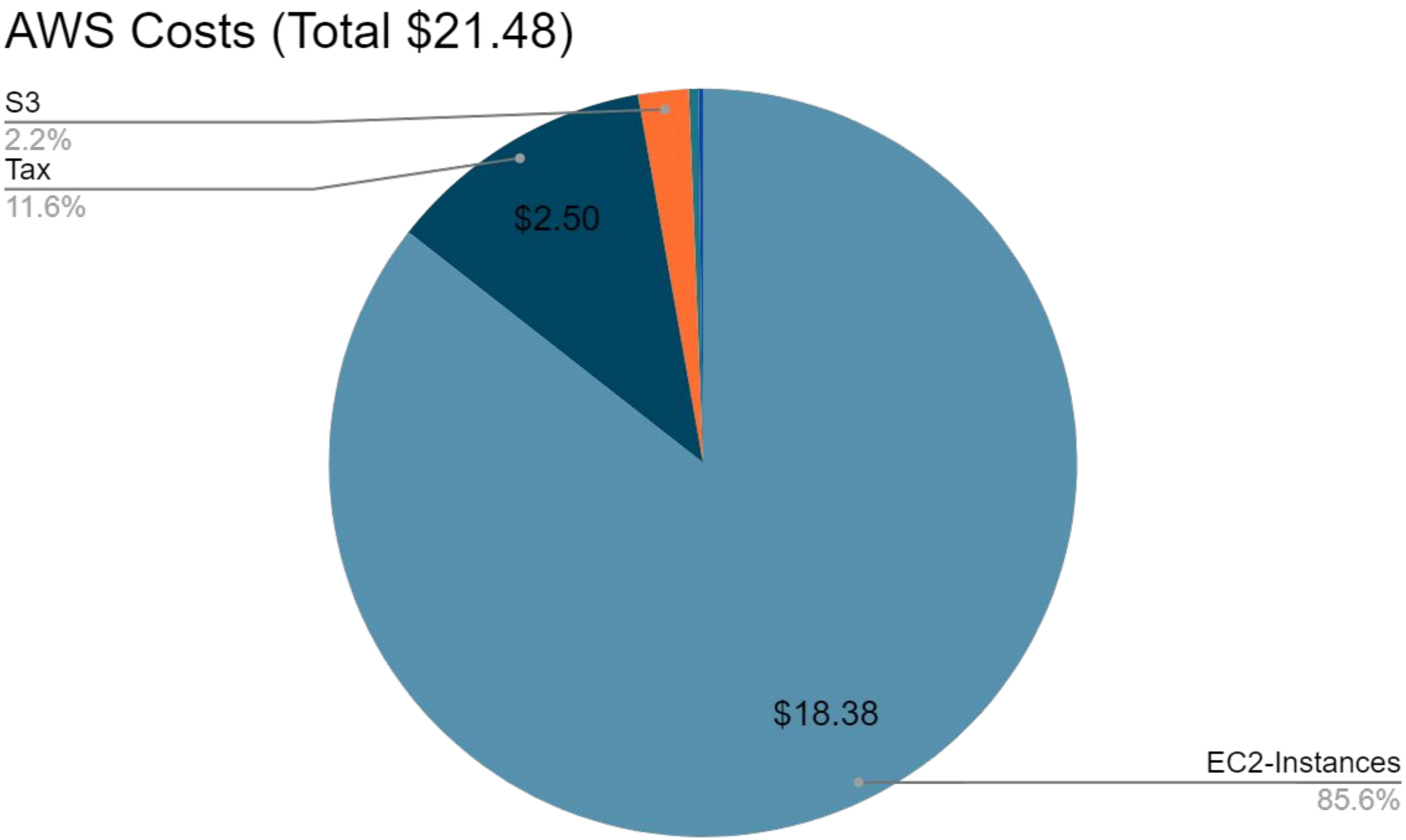
Cost Breakdown - March

Description		Usage Quantity	Amount in USD
Elastic Compute Cloud			
	Asia Pacific (Sydney)		
	Amazon Elastic Compute Cloud running Linux/UNIX		
	$1.208 per On Demand Linux r5.4xlarge Instance Hour	15.218 Hrs	**USD 18.38**
Simple Storage Service			
	Asia Pacific (Sydney)		
	Amazon Simple Storage Service APS2-Requests-Tier1		
	$0.0055 per 1,000 PUT, COPY, POST, or LIST requests	14,812 Requests	**USD 0.08**
	Amazon Simple Storage Service APS2-Requests-Tier2		
	$0.0044 per 10,000 GET and all other requests	57,586 Requests	**USD 0.03**
	Amazon Simple Storage Service APS2-TimedStorage-ByteHrs		
	$0.025 per GB - first 50 TB / month of storage used	14.357 GB-Mo	**USD 0.36**
Virtual Private Cloud			
	Asia Pacific (Sydney)		
	Amazon Virtual Private Cloud Public IPv4 Addresses		
	$0.005 per In-use public IPv4 address per hour	17.299 Hrs	**USD 0.09**

Cost Breakdown - March

16 Total Projects

- project-alpha/
- Project-Amos-B/
- project-amos-head/
- project-amos/
- project-banana/
- project-beta/
- project-didge/
- project-eclipse/
- project-gamma/
- project-grady-head/
- project-kevin/
- project-river-b/
- project-vegemite/
- project-yukio-b/
- project-yukio/



Job and Cost Estimates

Fast Orthophoto Job Estimates (Approximate Values Only)		
# of Photos	**Time (Mins)**	**Cost (USD)**
200	15	$0.30
500	30	$0.60
1000	60	$1.20

3D Model Job Estimates (Approximate Values Only)		
# of Photos	**Time (Mins)**	**Cost (USD)**
100	30	$0.60
250	90	$1.80
500	180	$3.60

Thank you for listening to our presentation

Questions?

WINYAMA
DIGITAL SOLUTIONS