

Weather Radar Prediction

- Jiras
- DevOps
- Architecture
 - High Level Architecture
- Analysis
 - Data Model
 - Image Format
 - Intermittent Effects
- API
- Use Cases
 - UC 1: Continuous Radar Download
 - UC 2: Radar Processing
 - UC 3: Radar Monitoring
 - UC 4: Radar Serving
 - UC 5: Historical Radar Download
- Design
 - Design Issues
 - Periodic Capture Framework
 - Static Image Analysis
 - Pixel Track Prediction
 - Continuously run a simulation of future storm pixel position and adjust the cone based on actual future tracks.
- Devops
 - Implementation and deployment
 - Virtual Machine based - EC2 VMware
 - Container based - Helm on Kubernetes on Docker
 - Serverless Based - Lambda
 - Testing
 - Security
 - Monitoring
- Releases
- Documentation
- Development Log
- Links

Jiras

[RADAR-2](#) - Getting issue details...

STATUS

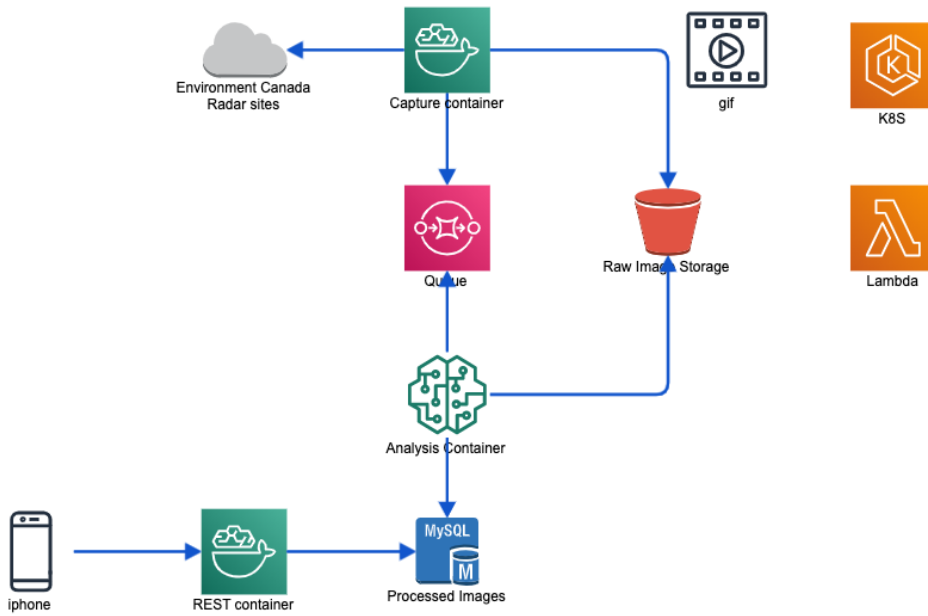
DevOps

Repo: <https://github.com/obrienlabs/radar>

Architecture

There are several data sites that serve up doppler radar images every 6-10 minutes. In addition we have correlating satellite, lightning and weather data. We also have historical radar imaging for missed downloads.

High Level Architecture



Analysis

Data Model

There are several radar sites across canada that provide individual and combined images at 10 and 6 minute intervals.

Image Format

480 x 560 with legend - active area is 480 x 480

XFT_2018_09_21_21_20_f
filtered



example XFT_2008_11_08_14_00.GIF

Intermittent Effects

xam is a good example

Out of order image ordering (historical)

Transient blocking from wind farms

Transient blocking from RF interference

Permanent false positive elements

single frame noise

Image jitter at 1 pixel

Noise past circular range rings

Historical radar missing data due to grid/names overlay

API

Use Cases

UC 1: Continuous Radar Download

UC 2: Radar Processing

UC 3: Radar Monitoring

UC 4: Radar Serving

UC 5: Historical Radar Download

Design

Design Issues

Periodic Capture Framework

Static Image Analysis

Pixel Track Prediction

Continuously run a simulation of future storm pixel position and adjust the cone based on actual future tracks.

Devops

Implementation and deployment

Virtual Machine based - EC2 VMware

Container based - Helm on Kubernetes on Docker

Serverless Based - Lambda

Testing

Security

Monitoring

Releases

Documentation

Development Log

Links