Broadsea 3.x: “BROADening the ohdSEA”

Ajit Londhe, Lee Evans, Sanjay Udoshi

Affiliations

Ajit Londhe (1,4); Lee Evans (2,4); Sanjay Udoshi (3,4)

1. **Boehringer Ingelheim, Ridgefield, CT USA**
2. **LTS Computing, West Chester, PA, USA**
3. **Acumenus Data Sciences, USA**
4. **Observational Health Data Sciences and Informatics, New York, NY USA**

OHDSI Broadsea Evolution

“Broadsea is the easiest way to install (& upgrade) the OHDSI tools”

10k+ Downloads

docker hub

v1.0 Atlas/WebAPI & RStudio Docker images on Mac/Linux/Windows

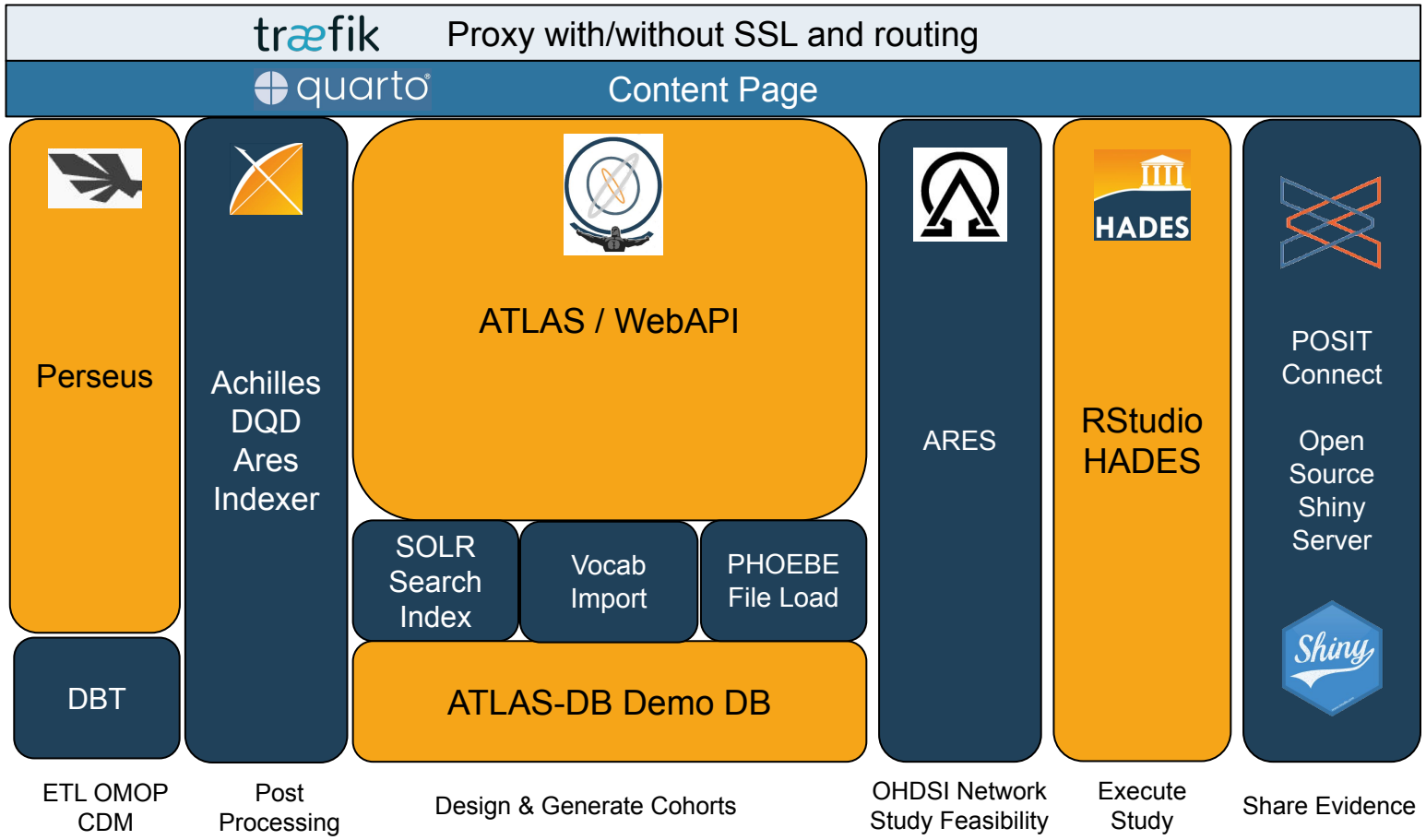
v2.0 Pre-populated demo postgres database image & Traefik reverse proxy

v3.0 Docker profiles for a-la-carte services, more Traefik networking, environment variable driven deployment, new OHDSI apps, build from Git

```
docker-compose --profile default up -d
```

https://github.com/OHDSI/Broadsea

OHDSI Broadsea 3.x



ATLAS Security Providers
Basic, LDAP, AD
OpenID, SAML

Build / Launch Options
Build from GitHub
Docker profiles

Databases
All OHDSI supported DBs

Configuration

One templated .env file

Docker Secrets

- default
  - atlas (“/atlas”)
  - WebAPI (“/WebAPI”)
  - AtlasDB (a Postgres instance for Atlas/WebAPI)
  - HADES (“/hades”)
  - A splash page for Broadsea (“/”)
- atlas-from-image
  - Pulls the standard Atlas image from Docker Hub
- atlas-from-git
  - Builds Atlas from a Git repo
  - Useful for testing new versions of Atlas that aren’t in Docker Hub
- webapi-from-image:
  - Pulls the standard WebAPI image from Docker Hub
- webapi-from-git
  - Builds WebAPI from a Git repo
  - Useful for testing new versions of WebAPI that aren’t in Docker Hub
- atlasdb
  - Pulls the standard Atlas DB image, a Postgres instance for Atlas/WebAPI
  - Useful if you do not have an existing Postgres instance for Atlas/WebAPI
- solr-vocab-no-import
  - Pulls the standard SOLR image from Docker Hub
  - Initializes a core for the OMOP Vocabulary specified in the .env file
  - No data is imported into the core, left to you to run through the SOLR Admin GUI at “/solr”
- solr-vocab-with-import
  - Pulls the standard SOLR image from Docker Hub
  - Initializes a core for the OMOP Vocabulary specified in the .env file
  - Runs the data import for that core
  - Once complete, the solr-run-import container will finish with an exit status; you can remove this container

- ares
  - Ares “/ares”
  - Builds Ares web app from Ares GitHub repo
  - Exposes a volume mount point for adding Ares files (see Ares GitHub IO page)
- content
  - A splash page for Broadsea (“/”)
- omop-vocab-pg-load
  - Using OMOP Vocab files downloaded from Athena, this can load them into a Postgres instance (can be Broadsea’s atlasdb or an external one)
  - Rebuilds the CPTs using the CPT jar file from Athena, with UMLS API Key (see .env file Section 9)
  - Creates the schema if necessary
  - Runs copy command for each vocabulary CSV file
  - Creates all necessary Postgres indices
  - Once complete, the omop-vocab-load container will finish with an exit status; you can remove this container
- phoebe-pg-load
  - For Atlas 2.12+, which offers Concept Recommendation options based on the Phoebe project
  - Loads Phoebe files into an existing OMOP Vocabulary hosted in a Postgres instance (can be Broadsea’s atlasdb or an external one)
  - Note: your Atlas instance must use this OMOP Vocabulary as its default vocabulary source in order to use this feature
  - Once complete, the phoebe-load container will finish with an exit status; you can remove this container

Broadsea 3.1 dot env file sections

Section 1: Broadsea Host

Section 2: Atlas GUI configuration

Section 3: WebAPI Database configuration

Section 4: Atlas security provider configuration

Section 5: WebAPI security configuration

Section 6: Building Atlas or WebAPI from Git

Section 7: SOLR Vocab (optional)

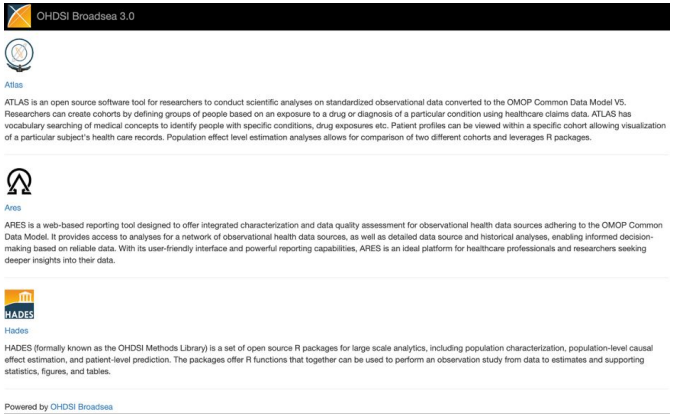
Section 8: HADES credentials to use in RStudio

Section 9: Postgres and UMLS credentials for loading OMOP Vocab files into Postgres schema

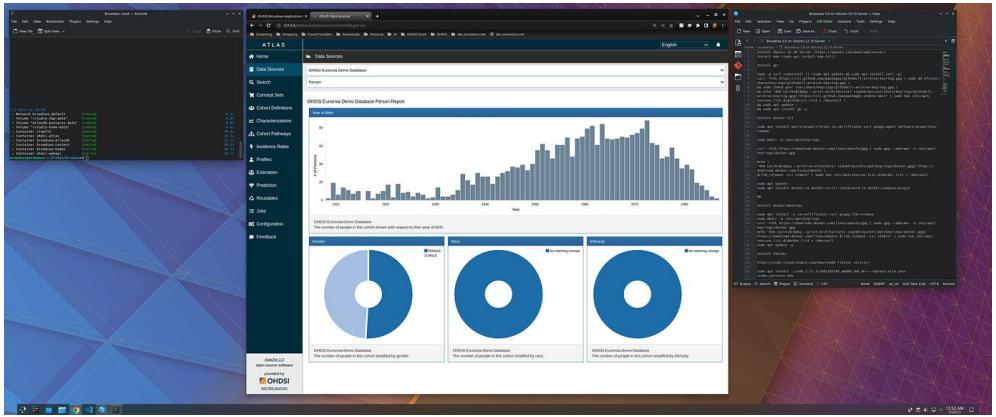
Section 10: Postgres credentials for loading Phoebe file for Atlas 2.12+ Concept Recommendations into Postgres hosted OMOP Vocabulary schema

Section 11: Ares Data Folder config

Section 12: Broadsea Content Page config



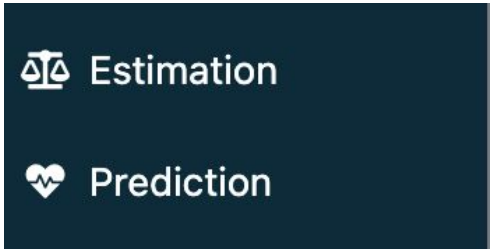
OHDSI Broadsea 3.1 Live Demo

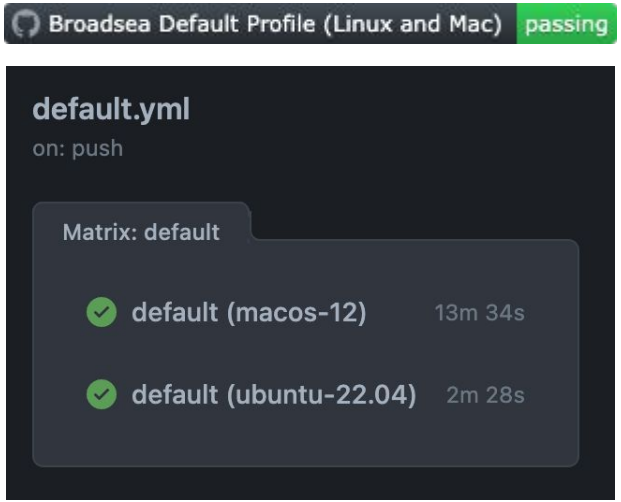


Broadsea future plans

podman

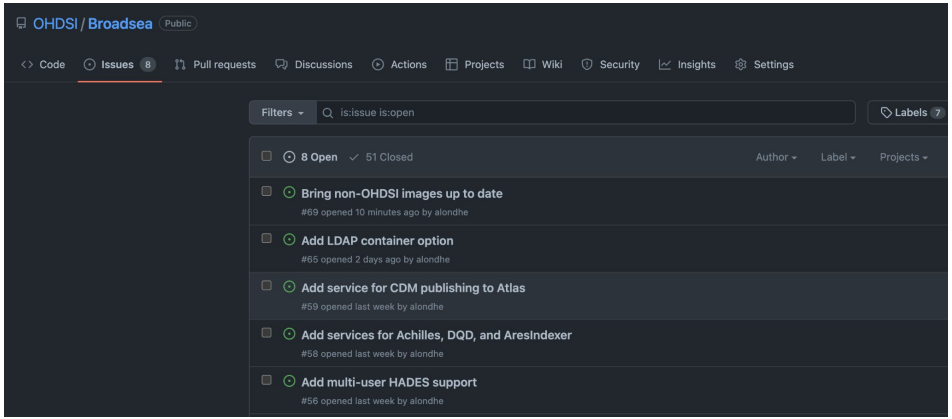
- Docker Secrets
- Add additional applications:
  - Example LDAP server
  - Arachne Execution Engine
  - OHDSI Perseus ETL
  - Community Edition Shiny Server
  - Posit Connect Commercial Server
- Add CDM lifecycle services
  - Achilles / DQD / AresIndexer
  - Atlas Source Management (Add/Archive)
- GitHub Actions for CI/CD testing





Broadsea vs WebAPI distinction

- Broadsea is a container of stuff already built
  - Either as standard releases or as GitHub repos
- For Databricks, any SQLRender / WebAPI specific code or JDBC assets are not handled explicitly by Broadsea
  - Code changes should be targeted for WebAPI development so as to become part of standard releases
  - However, if your WebAPI image has that code, Broadsea can run it
  - JDBC files can be mounted (upcoming Broadsea release) to add to the WebAPI if needed
- LDAPS and Databricks SSL needs can be handled via cacerts file
  - Build your cacerts file, then mount it



Add any questions or issues to the GitHub Repo