**Data Source and Maintenance**

1. Ellie downloads sonde data, places into folders. She cleans the raw data and separates them by year/quarter/site.
2. Lake\_profiles\_graphing project cleans all data and combines into one large .csv file
3. This .csv file is copied to the lakes\_profiles\_app data folder

**Shiny apps hosting**

**Shinyapps.io (best short-term and testing)**

* <https://arkansas-deq-water.shinyapps.io/ADEQ_profile_app/>
* Will require setting up an account to manage the apps
* Free and paid tiers
* Pricing - <https://www.shinyapps.io/>
* Free plan
  + 5 Applications
  + 25 active hours/month
    - Once exceeded, application will not be available (Free and Starter plans)
    - If exceeded on Basic plans or above, overage charge will be applied
* Easy to deploy, we can update app and data anytime without going through IT
  + No custom URLs on free plan
  + Can be embedded with an HTML iframe into any DEQ webpage, or have its own webpage

**Shiny Server Open Source (best choice long-term if possible)**

* [Posit Connect vs Shiny Server Open Source Comparison](https://posit.co/wp-content/uploads/2023/07/PST-ShinyServerPositConnect-ComparisonChart-July2023-v4.pdf)
* Free
* Unlimited usage (?)
* Host multiple Shiny apps on a single server, each with a unique URL or port
* Requires a Linux server, then must install Shiny Server R
  + [Pre-compiled binaries for Ubuntu 16.04+, Red Hat/CentOS 6+, and SUSE Linux Enterprise Server 12+](https://posit.co/download/shiny-server/)
  + [May be possible to build on non-supported distributions](https://github.com/rstudio/shiny-server/wiki/Building-Shiny-Server-from-Source)
* Requires effort and maintenance from IT
* App updates need to be sent to IT
* Data Updates:
  + manually send IT new database file, they replace the old one and it updates
  + Or use cloud storage service and modify app to pull from cloud:
    - Will require testing
    - Google sheets (public link)
    - Dropbox file
    - GitHub (file in a repo)
  + Connect app to the SQL Server
    - Create a SQL Server table for lakes data, IT gives us read/write permissions
    - We can push new data anytime, app automatically uses newest data

**Internal Only: Shiny Server Open Source on the Intranet**

* IT can point the app to our E: drive directory, thus we can handle updates entirely ourselves

**Server-less method (ShinyLive)**

Initial loading may take longer than other methods (~30 seconds)

1. Deploy app to GitHub Pages

* 100GB traffic/month – with current app, ~820 uses/month
* Can manually update database, push changes to GitHub, app will automatically refresh (no IT)
* Or use cloud storage service and modify app to pull from cloud:
  + - Will require testing
    - Google sheets (public link)
    - Dropbox file
    - GitHub (file in a repo)
* <https://hbctraining.github.io/Training-modules/RShiny/lessons/shinylive.html>
* <https://github.com/RamiKrispin/shinylive-r>
* <https://tanner-senti.github.io/shinylive/>
* Use GitHub desktop to push web app to GitHub or it won’t work!

1. Configure app in R, send files to IT and have them deploy to webpage

* Unlimited traffic?
* Updating data would require rebuilding app, sending to IT to update
* Or use cloud storage service and modify app to pull from cloud:
  + Will require testing
  + Google sheets (public link)
  + Dropbox file
  + GitHub (file in a repo)

**Other Cloud Services (AWS, Azure, etc)**

[Guide for deploying Shiny App on AWS (with Shiny Server)](https://www.charlesbordet.com/en/guide-shiny-aws/)