demo.dspace.org Notes

As of August 2023, the demo.dspace.org and sandbox.dspace.org sites are managed/hosted by the DSpaceDirect hosting team within Lyrasis as a donated, in-kind contribution back to the DSpace program.

This page is a guide for how these sites are setup and who to contact for specific needs. If major issues occur, it is best to contact Tim Donohue or one of the other Lyrasis team members.

- General Setup / Info
  - Infrastructure
  - Docker Images used
    - demo.dspace.org
    - sandbox.dspace.org
  - Updating Docker images
  - Restarting / Redeploying sites
  - Updating site configurations
  - Reset site content

General Setup / Info

Infrastructure

The demo.dspace.org and sandbox.dspace.org sites are hosted in Amazon Web Services (AWS) using the Docker images created via our Docker build scripts (more info below).

Both sites are deployed using Terraform modules to AWS Elastic Container Service (ECS).

The code behind these Terraform modules is open-sourced and made available at https://github.com/dts-hosting/terraform-aws-dspace It can be used as a guide/example for other similar setups, but these modules are unsupported at this time. They are built and maintained by the hosting team within Lyrasis specifically for these demo sites.

Docker Images used

The demo.dspace.org and sandbox.dspace.org setups are identical except for the Docker images used by each.

demo.dspace.org

demo.dspace.org uses the Docker Images from the current major release (7.x).

- The demo frontend uses the "dspace-7_x-dist" tagged image for "dspace-angular" on DockerHub: https://hub.docker.com/r/dspace/dspace-angular/
  - NOTE: The "-dist" image is the image which runs the UI in production mode, using PM2.
  - Dockerfile which builds this image: https://github.com/DSpace/dspace-angular/blob/dspace-7_x/Dockerfile.dist
- The demo backend uses the "dspace-7_x" tagged image for "dspace" on DockerHub: https://hub.docker.com/r/dspace/dspace
  - Dockerfile script which builds this image: https://github.com/DSpace/DSpace/blob/dspace-7_x/Dockerfile

sandbox.dspace.org

sandbox.dspace.org uses the Docker Images from the development branch (main).

- The sandbox frontend uses the "latest-dist" tagged image for "dspace-angular" on DockerHub: https://hub.docker.com/r/dspace/dspace-angular/
  - NOTE: The "latest" image is the image which runs the UI in production mode, using PM2.
  - Dockerfile which builds this image: https://github.com/DSpace/dspace-angular/blob/main/Dockerfile.dist
- The sandbox backend uses the "latest" tagged image for "dspace" on DockerHub: https://hub.docker.com/r/dspace/dspace
  - Dockerfile script which builds this image: https://github.com/DSpace/DSpace/blob/main/Dockerfile

Updating Docker images

Docker images for DSpace frontends/backends update automatically via GitHub Actions. Anytime a code change is made to either the "main" or "dspace-7_x" branches, the GitHub Action will trigger. This results in the image being rebuilt & send to DockerHub.

- Frontend's Docker GitHub Action: https://github.com/DSpace/dspace-angular/blob/main/.github/workflows/docker.yml
- Backend's Docker GitHub Action: https://github.com/DSpace/DSpace/blob/main/.github/workflows/docker.yml

NOTE: each branch uses the version of the GitHub action which exists on its branch. So, any updates to the GitHub action must be pushed to BOTH branches.

Restarting / Redeploying sites
demo.dspace.org and sandbox.dspace.org will **automatically redeploy** anytime a change is made to their Docker image. (This is controlled via webhooks installed in our DSpace organization at DockerHub)

For example:

- If the "latest-dist" image changes for "dspace-angular", then the sandbox.dspace.org frontend will redeploy. (This image will automatically update anytime a new change is pushed to the "main" branch of the "dspace-angular" codebase)
- If the "dspace-7_x" image changes for "dspace", then the demo.dspace.org backend will redeploy. (This image will automatically update anytime a new change is pushed to the "dspace-7_x" branch of the "dspace" codebase.)

### Updating site configurations

At this time, all site configurations updates must be sent to Tim Donohue or the Lyrasis team. All site configurations are currently managed via environment variables which must be configured in the (private) codebase for the demo site.

The basic environment variable setup can be seen in the Terraform modules codebase here: https://github.com/dts-hosting/terraform-aws-dspace/tree/main#custom-environment-and-secrets-configuration

### Reset site content

At this time, site content (for either site) can only be reset by Tim Donohue or the Lyrasis team. A command-line script exists to allow site content to be immediately reset to the demo entities data set. However, at this time it cannot be run externally, so someone at Lyrasis has to login the the instance and run it manually.

Eventually, we hope to schedule this script to run once per week to ensure site content is reset on a regular schedule.