OpenCDSS Memorandum

To:	Carolyn Kemp (Colorado Water Conservation Board), Brian Macpherson (CWCB), Mary Halstead (Colorado Division of Water Resources)
From:	Steve Malers (Open Water Foundation)
Subject:	Recommendations for OpenCDSS Website Hosting
Date:	April 3, 2018

Contents

Introduction

This memorandum summarizes recommendations for hosting OpenCDSS (and potentially other) website content, in particular:

- Software documentation
- Software installers
- Datasets

The OpenCDSS project and other projects are generating significant electronic content, including software documentation, data files, and software installers. The content (files) may be updated multiple times during projects and are also the results of ongoing work on open source projects. There is a need to implement a solution that will allow content to be accessed from the CDSS website while also allowing content maintainers/contributors to efficiently work with the content and make sure the content is deployed to the CDSS website.

Background

The CDSS website (http://cdss.state.co.us/Pages/CDSSHome.aspx) and other State websites have in the past been created using SharePoint, with content maintained by CWCB staff. The CWCB will soon migrate the CDSS website to Statewide Internet Portal Authority (SIPA) Drupal 7 platform (see https://www.colorado.gov/pacific/sipa/coloradogovservices). There is a need to provide access to

OpenCDSS content while also planning for such access in the new design. Currently, contractors cannot directly contribute content to the State's SharePoint websites and it is unclear how easy it will be to contribute to the new SIPA websites. Therefore, the following sections discuss options to use static websites to host content for specific topic areas.

Documentation Content

The Open Water Foundation (OWF) has experimented with a number of website tools and content management systems. OWF has utilized traditional website building tools (WordPress) for its own website; however, such an approach is inconvenient and inefficient for content that evolves and needs to be updated in bulk. Editing source files often need to be edited outside of the website editor. For example, OWF has utilized the MkDocs software (http://www.mkdocs.org/) to create interactive navigable documentation for CDSS software. MkDocs uses Markdown (https://en.wikipedia.org/wiki/Markdown) to format source documents into a static HTML website. Markdown has the following advantages:

1. Source files are text, so they can be easily edited with a text editor.
2. Because source files are text, they can be easily version-controlled, and standard "diff" (difference) tools can be used to view differences between documentation versions.
3. Markdown can also be used for other documentation purposes, such as README.md files with information about model datasets or other electronic files.

OWF has created quite a few MkDocs documents to provide documentation for various software tools and technologies (see http://learn.openwaterfoundation.org/). OWF has also created a number of MkDocs documents for CDSS software, at various levels of completion. It is expected that the documentation for CDSS software as implemented for the OpenCDSS project will continue to evolve (rather than being a project deliverable that is frozen at the end of a project contract. A goal is to update the documentation to be consistent with the current features of software. See the following table for examples of documentation. Note that the StateMod repository is currently private and can only be viewed if GitHub permissions have been granted (contact Steve Malers with GitHub account name if access is needed).

Program and Documentation	Source Code Repository for Documentation (MkDocs/Markdown Files)	Online Documentation (Static Website)
StateMod Developer Documentation	https://github.com/OpenWaterFoundation/cdss-app-statemod-fortran/tree/master/doc-dev-mkdocs-project	http://learn.openwaterfoundation.org/cdss-learn-statemod-dev/
TSTool User Documentation	https://github.com/OpenWaterFoundation/cdss-app-tstool-doc-user	http://learn.openwaterfoundation.org/cdss-app-tstool-doc-user/

The following sections illustrate how standard cloud web services can be used to host documentation content on a static website. This allows content to be hosted on a major cloud platform and accessed from a URL of choice.

Note that the approaches described below are not equivalent to solutions such as Dropbox or Google Drive, which are unable to host websites because they provide access to the source files with the "skin" of the hosting service. Options equivalent to those discussed below may be available and most are similar. One area that may need additional evaluation is the use of https rather than http. For example Esri Story Maps appear to prefer https sites.

Hosting Content on Google Cloud Platform

It may be appropriate to host website content on the Google Cloud Platform (GCP) Cloud Storage, for example when contractors such as OWF or others use GCP for cloud storage for project deliverables. The State of Colorado water agencies including CWCB and DWR also use GCP for cloud IT services.

OWF recently completed a project to install CDSS SNODAS Tools on GCP. In this case, the Google Cloud Platform bucket "snodas.state.co.us" was created to hold content. A script in the following repository folder is used to deploy the website files to the GCP bucket:

https://github.com/OpenWaterFoundation/cdss-webapp-snodas-tools/tree/master/build-util

The deployed website is then accessible at the following URL:

http://snodas.cdss.state.co.us/app/index.html

The snodas.cdss.state.co.us internet domain is a sub-domain under the cdss.state.co.us domain and is configured using CNAME alias for the State of Colorado domain name service (DNS). GCP documentation describes how to set up the static website:

https://cloud.google.com/storage/docs/hosting-static-website

Hosting Content on Amazon S3

It is possible to host website content on Amazon S3, for example when contractors such as OWF or others use Amazon S3 for cloud storage for project deliverables.

The content for the documentation websites shown in the table above are maintained in GitHub repositories. In this case, the Amazon S3 bucket "learn.openwaterfoundation.org" was created to hold content. Scripts in the repository are used to deploy the documentation to public cloud servers. For example, see the scripts in the following repository folder, which installs the StateMod developer documentation to OWF's Amazon S3 cloud storage site.

https://github.com/OpenWaterFoundation/cdss-app-statemod-fortran/tree/master/doc-dev-mkdocs-project/build-util

The site can then be viewed using the following URL:

http://learn.openwaterfoundation.org/cdss-learn-statemod-dev/

The learn.openwaterfoundation.org internet domain is a sub-domain under the openwaterfoundation.org domain and is configured using CNAME alias for the Open Water Foundation domain name service (DNS). Amazon Web Services supports this approach, as documented below:

https://docs.aws.amazon.com/AmazonS3/latest/dev/website-hosting-custom-domain-walkthrough.html

Data Content

The approach described above can also be used to host data content. For example, model datasets, GIS files, or other data files can be hosted on a static website. The advantage of this approach is that the content can be developed on a desktop computer and then loaded with a script. OWF has used this approach on the following static website:

http://data.openwaterfoundation.org/

A landing page can be created, which provides links to the files, or add those links to another website, such as the new SIPA website that will be implemented.

Software Installers

The approach described for data above can also be used to host software installers. For example, different versions of software installers can be made available.   OWF has used this approach to host the download files available from a Google sites website:

https://sites.google.com/site/cdssstaging/tstool/download

Recommendations

The OpenCDSS project and other projects are generating significant electronic content, including software documentation, data files, and software installers. These files may be updated multiple times. Content may be delivered to the State using traditional approaches (zip file, Google Drive, Dropbox, DVD, etc.). Or, delivery may occur using a GitHub repository, in which case an installation script may be used to copy files from source to the installed location. The challenge is how to allow contractors to develop content over time and deploy to standard CDSS website location for public access.

It is recommended that the State consider using cloud storage buckets and static websites to serve content, as described in this memorandum. This will allow standard processes to be put in place to deliver and deploy content, as well as providing public access to content. It is assumed that such content is not suitable for the State's imaged document platform (LaserFiche), although the CWCB can determine to use that platform where it makes sense. If a static website approach with cloud storage is taken, the following are options to handle deployment of the content:

1. **Allow third parties such as contractors, basin roundtables, etc. to configure cloud storage**. In this case, State websites can link to those sites directly, or can define a DNS CNAME entry to associate a CDSS URL with the third party URL. The latter typically requires proof of ownership to allow the alias, which may present a problem. The third parties would be able update the content as products

are updated. A disadvantage of this approach is that there is a cost for cloud storage. However, Amazon S3 and Google Cloud Storage costs tend to be minimal (OWF's bill is often less than $5 per month).

2. **Configure cloud storage within the State's Google Cloud Platform**, similar to SNODAS storage bucket. In this case, there are two options to load the content:
    a. **State provides credentials to the third party to have access to the storage bucket.** This is similar to OWF's access to the SNODAS storage bucket. The third party would deploy the content.
    b. **State staff uploads content to the static website**. In this case, the third party needs to provide the file to the State by some intermediate method, such as an FTP site, GitHub repository, USB drive, DVD, Google Drive, Dropbox, email, etc. The burden of deploying content would fall on State staff.

It is also recommended that the above options should also use GitHub (or comparable cloud version control platform) to version-control content. Any scripts that upload content can then be version-controlled. Sensitive information such as passwords for cloud platforms should not be included in public repositories.

Finally, the State should coordinate website content with third parties to ensure that content on the CDSS website is consistent with contractor work. For example, documentation for CDSS software maintained as separate static websites will need to fit into the CDSS site map. The State's CDSS team should have a leading role in defining this structure.

Implementing one of the above options will allow content deployment to be automated and controlled through standard processes, such as the scripts that OWF has implemented in repositories to deploy content. This approach is particularly suitable for technical content that needs to be updated as work is completed.