

NMR web scheduler migration

~13 (16?!) year old system suffering from benign neglect. Public web server with few if any updates to OS or web server, on Chemistry's "trusted" network. Original re-write plans fell through.

Update

Added documentation, after this project, to clarify support is from Ivan at NMR, not Chemistry IT or CIT

- [NMR Scheduler, manage by NMR](#)

10/11/16: Ivan and CIT successfully migrate service to Amazon Web Services. With support by Chemistry IT staff.

- Oliver conceived of solution.
- Oliver invested effort to convince Ivan this was a worthy investment, and connected Ivan to free service at CIT.
- Oliver invested effort to ensure conversation did not drop as Ivan explored migration with CIT staff.

Goal

Completion by at least the end of August (complete this summer). What does success look like?

- NMR Scheduler still works. (See below for access requirements.)
- Current old (at-risk) hardware no longer being used.
 - Prone to power-outage problems.
- Current OS and Apache (insecure) software no longer being used.
- System is no longer on Chemistry's trusted network.
- Chemistry IT continues not to have any responsibilities regarding the functioning of the server or its services.

NMR Scheduler access requirements

- Public: Web access to the NMR Scheduler web application.
- Ivan's access:
 - ssh: read and edit files
 - sftp: read and copy off files

Process roles for migration

- Ivan to make the migration happen.
- Chemistry IT (mostly Oliver) to provide consultation services and:
 - Chemistry IT to provide this documentation so project moves forward efficiently and with fewer surprises.
 - Chemistry IT to induce pressure to ensure the server moves off of Chemistry's trusted network in a timely manner.
 - This will include organizing regular check-in meetings. And managerial escalation if pace or engagement not sufficient to move server off of Chemistry's trusted network.

To do's

Date	Person	Task	Notes
5/20/16	Shawn (CIT)	Root around server to determine if good candidate for encapsulating in Docker. If it is, copy out so a test instance can be tested.	If and when a successful Docker container is made, it can be hosted at AWS (intention) or elsewhere, such as at Bluehost. The lowest cost system is \$4.76 (1CPU, .5GB RAM, "T2-Nano") which likely is all that is needed. Goes up another \$5 for a full 1GB ("T2-Micro"), and onwards similar to this (procs and RAM) for up to ~\$40/month. You select the price and then only upgrade if you need to in order to meet performance needs, so no surprise billing. Seem unlikely it will need more than the lowest, right? CIT is willing to allow us to use their existing account for the AWS accounting rather than set one up just for this instance. If other instances are need, we should then set up an AWS account. • CIT would then pass-through bill, via KFS, our specific AWS charge.
5/20/16	Ivan	On hold? Review web-hosting services (Future task: Select and try moving a copy to one of these services.)	• https://www.bluehost.com/ ◦ https://www.bluehost.com/shared (too high on the stack? Can shoe-horn existing into their structure?) ◦ https://www.bluehost.com/vps (too low on the stack? Apache included?) ◦ https://www.bluehost.com/professional-services/sitetransfer For \$149.99, this one-time cost seems really cheap, if it actually works! Especially considering their statement here: ▪ https://my.bluehost.com/cgi/help/433

5/20/16	Oliver	Create calendar proposals for check-in meetings every other week, through August, to ensure on-track.	Must skip July since Oliver away that month.

Completed tasks

Idea: Simply copy completed above items to here, specifying approx. date completed.

Completion date (approx)	Original date	Person	Task	Notes
5/20/16	5/20/16	Oliver	Get CIT root access to the NMR Scheduler	Passwords sent via Cornell Dropbox, 'natch.

Meeting notes (reverse chron)

5/20/16 Fri

Oliver and Ivan met. Reviewed Oliver's queries to CIT (IWS and CIT's AWS/ Docker folks).

Ivan can't afford web developer work so re-write not appropriate.

Learning Docker and associated abstraction layers also does not seem appropriate.

Instead, look for server hosting to host current Perl-based application (with server-side includes) such that someone else is maintaining current:

- OS
- Apache

We hope a migration "just works" with minimal input from Ivan. Oliver's concern is that this won't be a slam-dunk migration due to age of code (deprecated functions, etc) and configuration difference. Bluehost has experience with users doing this, and have written to this point here:

- <https://my.bluehost.com/cgi/help/433>