Project Tasks

1. Investigate Vagrant
    a. Simple CentOS 7 setup
    b. Simple Ubuntu setup
2. Investigate Docker
    a. Composite setup
3. Present options.
    a. Discuss stage-based approach: development, testing, deployment.
4. Check out RubyMine.
5. Generate and propose ideas (#1, #2, #3).
    a. Respond to feedback.
6. Create Vagrant project. Do a manual stack install to test Vagrant.
7. Create Vagrant project that provisions automatically.
    a. Choose provisioning tool. (Shell)
8. Create Vagrant project that provisions and installs the Ruby stack automatically.
    a. RVM setup
9. Investigate RubyMine Vagrant connection
10. Create sample RubyMine Vagrant project to document how RM works with Vagrant.
11. Experiment with RubyMine multi-project setup options
    a. RM Multimodule
    b. Git Submodule => doesn't work with RM
12. Document and share # 7
13. Create Quicksearch Vagrant project
    a. Create Vagrant file
    b. Create provision file (shell)
    c. Git SSH key import into Vagrant
    d. Set up search-frontend
    e. Set up search-backend
    f. Set up quicksearch-morris
14. Request and act on project technical documentation.
    a. Request architecture diagram to see how the components interact.
15. Resolve all component dependencies:
    a. MySql
    b. PostgreSql
    c. Oracle client library
        i. Distribution/licensing issue => Resolved with 'binaries' repo
        ii. GitHub repo for Oracle client library (tmp)

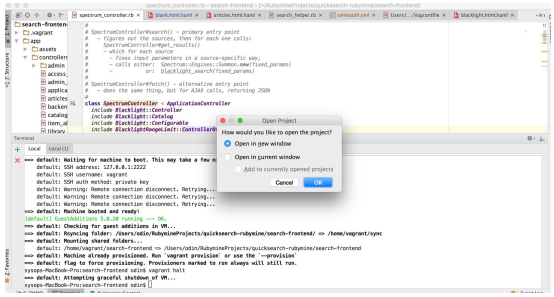
    d. Solr
        i. Jetty Solr, Tomcat Solr, Solr version
        ii. Import issue => Resolved with actual index
16. Resolve all configuration dependencies:
    a. ominauth.yml
    b. solr/schema.xml, solrconfig.xml
17. Resolve all issues:
    a. RM-Debugger
    b. RM-Remote SDK
    c. Mounts
    d. Networking
18. Load data (e.g., into Solr).
19. Backend-search test/configuration
20. Document # 10 and ask for feedback
21. Incorporate any feedback from user testing
22. Investigate networking options, firewall issues
    a. Voyager (clark) connection test
    b. Morris connection test/configuration (if necessary)
    c. Access from one guest to another via private networks
    d. Bridged versus private versus Ethernet
    e. Firewall issues
23. Figure out box updates (i.e., who, when, . . .)
24. Refactor (TODO):
    a. Script -- directory, redundant load
    b. Sync between different files
25. Decide on separate VMs for MySQL, etc. (todo)
26. For each individual box:
    a. Enable ip settings in the Vagrantfile; set up MySQL/Solr remote
27. Communication
    a. Updates
    b. Present
28. AWS

Steps for QuickSearch RubyMine (RM) Vagrant:

- Memorize:
  - `vagrant up`
  - `vagrant reload --provision`
  - `vagrant ssh`
  - `vagrant snapshot push`
  - `vagrant global-status`
  - `vagrant status`
  - `vagrant halt`
  - `vagrant destroy id`
- Install VirtualBox, Vagrant, RubyMine.
  - Run "`vagrant -v`" on Terminal to verify vagrant is working.
  - Install Vagrant vbguest plugin: `vagrant plugin install vagrant-vbguest`
- Make sure you have set up GitHub keys in repo. The steps are:

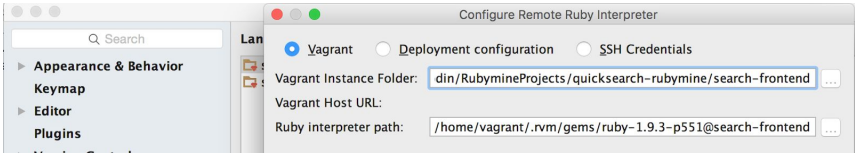
```
ssh-keygen -t rsa -b 4096 -C you@email.com
# press enter to save key to your home folder
eval "$(ssh-agent -s)"
# it should print "Agent pid . . ."
ssh-add ~/.ssh/id_rsa
# it should print "Identity added . . ."
pbcopy < ~/.ssh/id_rsa.pub
# now paste this key into your GitHub SSH Settings page
ssh -T git@github.com
# enter "yes" to recognize host
```

- Start RubyMine. Git clone in RubyMine (third option in the launcher) all the QS projects:
  - https://github.com/osmandin/search-frontend
  - https://github.com/osmandin/search-backend/
    - Make sure that you import the second project in "Current Window"

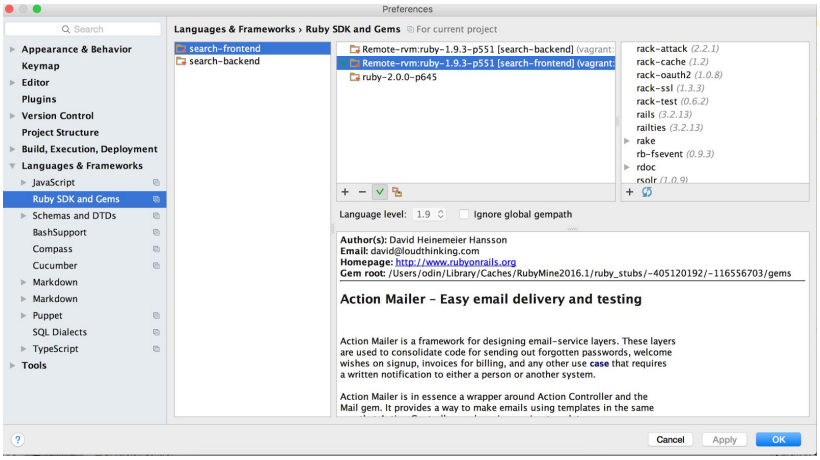


  - Open Terminal (in RM)
  - (Switch to branch "vagrant" through Terminal in RubyMine)

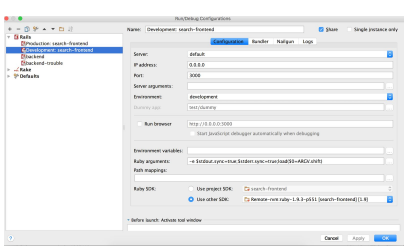
- Optional: Copy in “tmp” config folder (from your email) to both search-frontend and search-backend (e.g., through Finder)
- Edit Vagrantfile synced_folder setting for both repos to make sure that it points to the correct folder.
- Run “vagrant up”
    - Install vbguest plugin (in case you haven’t already): `vagrant plugin install vagrant-vbguest`
    - Run “`vagrant reload --provision`”
- Configure connections.
    - Verify box “centos/7” shows up in RM/ Preferences/ Tools/ Vagrant
    - RM/Languages & Frameworks/Ruby SDKs & Gems
    - Click on + icon to add remote SDK. Select Vagrant in the dialog box. Make sure the path is to search-frontend in the first box, and the last path points to gem locations (the same location as the path vagrant loading says “Happy Coding!”)

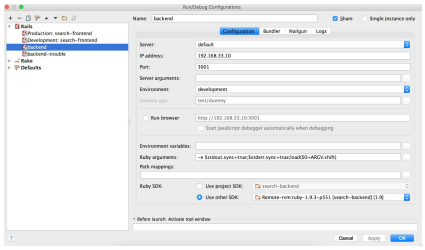


    - Verify gem/bundler availability for each (e.g., Rails 4 for search-backend; Rails 3 for search-backend):

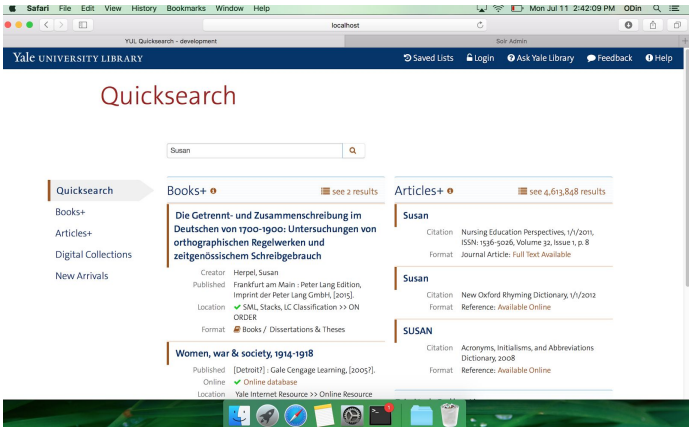


- “`vagrant ssh`” into the box. If the default gemset is not selected, select it.
- Verify database for both SF and SB:
    - `cd quicksearch-vagrant/db; sqllite3 dbname`
    - `rake db:migrate` (in case the database is missing)
- Start Solr on search-frontend (if it’s not running).
    - Verify by running `ps -A | grep java`.
- RM/Run to run both programs (or select each project and click on Run -- it should automatically pick up the project runtime info, but if it doesn’t click on Edit Configuration). The Edit Configuration should look like the following. Note the different IP and port for search-backend.





- Finally, browse to
    - `localhost:3000` for search-frontend
    - `localhost:8983` for solr



Possible Issues

- Sometimes RM might have problem connecting to the box. Try again.

Consolidated VM Instructions:

- Git clone:
    - https://github.com/yalelibrary/quicksearch-vagrant
- Follow the steps in the repo.

Pending:

1. Morris