AboutTime Virtual Machine Setup

The current AboutTime project is being hosted using the ngrok service, nodejs, as well as a locally hosted mysql database on the SDL student VM 02. These instructions outline three parts: connecting to the VM, setting up the project on a fresh VM (for Ubuntu 16), and hosting a new instance of the project on the VM.

SDL VM Connection Steps

1. Go to your command-line/terminal
2. Type: *ssh msoeUsername@ad.msoe.edu@sdlstudentvm02.msoe.edu*
    a. msoeUsername would be replaced with your MSOE username, the part that predicates @msoe.edu in your email
    b. sdlstudentvm02.msoe.edu is the virtual machine we're connecting to, the second software development lab virtual machine. If the virtual machine changes, as does this part
3. Put in your password, this should be the password you use to log into your other MSOE accounts such as your email and my.msoe.edu

VM Initial Setup Steps

Note: If the project is already set-up, skip this section

1. Download GIT, typing *sudo apt update*, then *sudo apt-get install git*
    a. Provide your password if prompted
2. Set up the Bitbucket ssh, by typing *ssh-keygen*   hitting enter to confirm the location   entering a passphrase for the ssh key typing *cat ~/.ssh/id_rsa.pub* copying the displayed keys   and pasting them to your Bitbucket account
    a. Success here can be verified by typing *ssh -T git@bitbucket.org*
    b. For more info see: https://confluence.atlassian.com/bitbucket/set-up-ssh-for-git-728138079.html
3. Type: *git clone git@bitbucket.org:hasker/abouttime.git*
    a. In the event that this is no longer the current repository, you want to find the new link and use that here
4. Type *cd abouttime* to switch to the abouttime directory
5. Install node.js by typing *sudo apt-get install nodejs*
6. As we're working in Ubuntu16, we also need to get legacy nodejs as well, so type *sudo apt-get install nodejs-legacy*
    a. This is due to an issue with Phantomjs not installing without legacy
7. Then type *sudo apt install npm* which grants us the npm command
8. To install project dependencies type *sudo npm install*
9. To install a plugin we need for our database migration, type *sudo npm install -g db-migrate*
10. Type *sudo apt-get update* to make sure you are up-to-date
11. Type *sudo apt-get install ngrok-client* to install ngrok, which we use to host the project
12. Then type *sudo apt-get install mysql-server* to install mysql
    a. Set a password, this will be used later. Make sure to keep this safe somewhere, or make sure you won't forget it.
13. Type *sudo mysql -u root*
    a. You may be prompted for your mysql password
14. Type *CREATE DATABASE time_logger;*
15. *Type use mysql;*
16. Type *update user set authentication_string=password(''), plugin='mysql_native_password' where user='root';*
    a. This should resolve a privilege issue when trying to migrate later
17. Type *flush privileges;*
    a. Not doing this will ruin the database migration later
18. Type *exit*
19. We need to set up our .env file, so type *cp .sample-env .env*
20. Then type *cp public/atlassian-connect-sample.json public/atlassian-connect.json*
    a. This will be used when hosting the project later on, but won't be touched for now
21. Type *nano .env* to start editing the file
    a. JIRA_URL: This can be left default
    b. JIRA_USERNAME: Your JIRA username
    c. JIRA_PASSWORD: Your JIRA password
    d. DB_HOST: Set to "localhost"
    e. DB_NAME: Set to "time_logger"
    f. DB_USER: Set to the set mysql username (root by default)
    g. DB_PASS: Set to the set mysql password (empty string by default)
    h. BITBUCKET_USERNAME: Retrieve the information from Dr. Hasker, as it's stored in a private confluence page
    i. BITBUCKET_PASSWORD: Same as above.**Note:** The Bitbucket username and password are used for issue reporting but are not required for the rest of the project to function
22. Type *sudo db-migrate up* for the purpose of database migration
23. Type *mkdir logs* to make a logs directory
    a. This will be used to store our node/ngrok hosting logs later on

VM Project Hosting

Note: the distribution of ngrok in Linux (using apt-get) results in a version with a bad hostname. Download the latest version from ngrok.com (64-bit linux version) and execute that instead.

Note: Skip to part 3 if you already have a ngrok account, and have saved the auth token to the virtual machine server.

1. Create a free ngrok account at https://dashboard.ngrok.com/user/signup
    a. Store the credentials somewhere you can remember
2. Log into the ngrok account go to https://dashboard.ngrok.com/get-started select Connect Your Account copy the authtoken line then paste it into the virtual machine terminal
    a. If you get 'no such file or directory', remove the *./* part of the auth token

On the linux host, place the credentials in ~/.ngrok2/ngrok.yml

3. If you are not already in the proper directory, type *ls* to find the *abouttime* directory and use *cd* to traverse to it.
    a. You should be the proper directory when *ls* returns a location with a file named index.js

mkdir logs

4. Type *nohup ./ngrok* jXX *HTTP 8081 > logs/ngrok_currentdate.out 2>&1 &*   <- lower case http
    a. Replace currentdate with the current date, using the format MM-DD-YYYY
    b. If you get 'no such file or directory', remove the *./* part of the ./ngrok

This step on MSOE blade server

5. To find the ngrok URL, go to https://dashboard.ngrok.com/status and find the ngrok URL using https, NOT http. This is the URL being used to host the project, so copy it
    a. There should only be two URLs as an account can only host one server at a time, and the only difference is http vs https. If this is not the case, try to kill any outstanding ngrok processes and start over from step 4
        i. To kill outstanding ngrok processes, in the VM terminal type *ps -A* | *grep ngrok* to see ngrok processes, and *kill pid* where pid is the process id given from ps
6. Type *nano public/atlassian-connect.json* and edit the file's baseUrl, replacing it with the ngrok URL you found.
    a. Keep the URL enclosed in quotation marks.

This typically replaces t00ter.com

7. Type *nohup node index.js > logs/node_currentdate.out 2>&1 &*
8. In another window, Go to Jira settings, then select *apps* *manage apps* *uninstall* the last AboutTime app (if applicable) -> click *upload* app -> then paste in the url from ngrok, appending at the end: /atlassian-connect.json, and submit the URL
    a. An example of the final url would be: https://b0da8a6c.ngrok.io/atlassian-connect.json
9. Type *sudo mysql -u root*
    a. Provide your password, if necessary
    b. **Note**: this step becomes is necessary due to confluence status report publishing requiring a secret key, and the project not getting one by default.
10. Type *use time_logger;*
11. Type *UPDATE organization SET confluence_shared_secret='fLxT3HrBFtboh0e3xfEXRUe7QkkkaDKForA+3HqoOe98v8c9VSJou0o6vPRaIZGuZgT0I8WKc2Ihmh4vrcGtfA' ORDER BY organization_id DESC LIMIT 1;*
12. Type *flush privileges;*
13. Type *exit*