PROTOCOL TBro Amazon Lightsail Template 1.0

Introduction

This protocol will setup up TBro in Docker in Amazon AWS Lightsail. **Items in red** must be replaced with your own information or files.

Materials

- Transcriptome transcripts and peptides
  - T1-transcriptome.tr
  - T1-transcriptome.aa
- TBro peptide tables
  - T1-tbro-table.tbl
- InterProScan annotation
  - T1-annotations.tsv
- Zipped BLAST databases and md5sums
  - T1-blastdb-TR.zip
  - T1-blastdb-AA.zip
  - list-zipped-blastdb-md5sums

Procedure

REVIEW TBRO INSTALLATION AND USE

1. **TBro documentation**

   https://tbro-tutorial.readthedocs.io/en/latest/index.html#

CREATE AMAZON AWS LIGHTSAIL INSTANCE

2. **Create an Amazon AWS account (requires phone for security check)**

   https://aws.amazon.com/

3. **Create an Amazon AWS Lightsail instance (under Services > OS only > Ubuntu)**

   https://lightsail.aws.amazon.com/ls/webapp/home/resources

4. **Create a static IP**

   https://lightsail.aws.amazon.com/ls/webapp/home/resources

5. **Assign the static IP to your Lightsail instance**

   Follow directions on Amazon AWS Lightsail

6. **SSH to Amazon AWS Lightsail instance**

   Click the button

INSTALL DOCKER

7. **Uninstall any old CE versions of Docker**

   sudo apt-get purge docker-ce

   sudo rm -rf /var/lib/docker

   sudo apt-get remove docker docker-engine

8. **Select Ubuntu docker installation instructions**

   https://docs.docker.com/engine/installation/

   https://docs.docker.com/engine/installation/linux/ubuntu/

9. **Install packages for apt**

   sudo apt-get install apt-transport-https ca-certificates curl software-properties-common

10. **Add Docker GPG key**

    curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

11. **Check that key fingeprint is 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88**

    sudo apt-key fingerprint 0EBFCD88

12. **Set up stable repository**

    sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

13. **Update apt**

    sudo apt-get update

14. **Install Docker CE Unbuntu AMD64**

    sudo apt-get install docker-ce

15. **Add your user to user group**

    sudo usermod -aG docker $USER

16. **Connect to your instance using ssh from Amazon Lightsail interface**

    close the Lightsail terminal and then re-connect by SSH from Lightsail

17. **Run hello world from your Docker user group user**

    docker run hello-world

18. **Configure system to start Docker when it boots up**

    sudo systemctl enable docker

INSTALL TBRO

19. **Pull Tbro Docker images**

    docker pull greatfireball/generic_postgresql_db

    docker pull tbroteam/generic_chado_db_reload

    docker pull tbroteam/tbro_worker_ftp

    docker pull tbroteam/tbro_worker

    docker pull tbroteam/tbro_apache

20. **Start CHADO database and install schema**

    docker run -d -e DB_NAME=chado -e DB_USER=tbro -e DB_PW=tbro --name "Chado_DB_4_TBro_official" greatfireball/generic_postgresql_db

    sleep 60

    docker run --rm -i -t --link Chado_DB_4_TBro_official:CHADO --name "Chado_DB_4_TBro_load_official" tbroteam/generic_chado_db_reload

21. **Start database container for BLAST worker**

    docker run -d -e DB_NAME=worker -e DB_USER=worker -e DB_PW=worker --name "Worker_DB_4_TBro_official" greatfireball/generic_postgresql_db

22. **Start FTP server to host BLAST database**

    docker run -d --name "Worker_FTP_4_TBro_official" -e FTP_USER="tbro" -e FTP_PW="ftp" tbroteam/tbro_worker_ftp

23. **Start worker to execute BLAST server**

    docker run -d --link Worker_DB_4_TBro_official:WORKER --link Worker_FTP_4_TBro_official:WORKERFTP --name "TBro_Worker_official" tbroteam/tbro_worker

    docker exec -i -t TBro_Worker_official /home/tbro/worker_build_installation.sh

24. **Build TBro Docker container**

    docker run -d --link Chado_DB_4_TBro_official:CHADO --link Worker_FTP_4_TBro_official:WORKERFTP --link Worker_DB_4_TBro_official:WORKER --name "TBro_official" -p 80:80 tbroteam/tbro_apache

    docker exec -i -t TBro_official /home/tbro/build_installation.sh

25. **Start TBro Docker container**

    docker exec -it TBro_official /bin/bash

PASSWORD PROTECTION

26. **Install apache2 utilities**

cd /etc/apache2

sudo apt-get update

sudo apt-get install apache2-utils

27. **Set up user and password**

    sudo htpasswd -c /etc/apache2/.htpasswd USER

    PASSWORD

    PASSWORD

28. **Modify Apache(?) configuration file**

    sudo apt-get install nano

    sudo nano /etc/apache2/sites-enabled/000-default.conf

```
### FILE BEGIN
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
         It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

        #### BEGIN: ADD THIS TEXT TO FILE

        <Directory "/var/www/html">
```

```
        AuthType Basic
        AuthName "Restricted Content"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Directory>

    #### END: ADD THIS TEXT TO FILE

</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
### FILE END
```

29. **Test syntax and restart TBro**

    sudo apache2ctl configtest

    exit

    docker restart TBro_official

IMPORT TRANSCRIPTOMES

30. **Enter TBro Docker container**

    docker exec -it TBro_official /bin/bash

31. **Add species information to CHADO in TBro database**

    tbro-db organism insert --genus Octopus --species chierchiae --common_name pygmy octopus --abbreviation O.chierchiae

32. **Check organism table for organism id**

    tbro-db organism list

33. **OPTIONAL To update/correct errors in organism table**

    tbro-db organism update --genus Octopus --species chierchiae --common_name tiny octopus --abbreviation O.chierchiae

34. **Build directory system in TBro Docker container**

    mkdir -p octopus

    cd octopus

    mkdir -p transcripts

    mkdir -p peptides

    mkdir -p annotations

    mkdir -p blastdbs

35. **Copy transcript files into TBro Docker container**

    cd transcripts

scp id@address:/path/to/transcripts/T1-transcriptome.tr .

36. **Create list of all identifiers for each transcriptome**

    grep ">" T1-transcriptome.tr | perl -pe 's/>(\S+).*/$1/' > T1-identifiers.ids

37. **Import transcript sequence ids into TBro database**

    tbro-import sequence_ids --organism_id 13 --release octopus-T1 --file_type only_isoforms T1-identifiers.ids

38. **Import transcript sequences into TBro database**

    tbro-import sequences_fasta --organism_id 13 --release octopus-T1 T1-transcriptome.tr

39. **Copy TBro peptide fasta files into Docker TBro Docker container**

    cd ../peptides

    scp id@address:/path/to/peptides/T1-transcriptome.aa .

40. **Copy TBro peptide tbl files into Docker TBro Docker container**

    scp id@address:/path/to/peptides/T1-tbro-table.tbl .

41. **Import peptide table into Tbro database**

    tbro-import peptide_ids --organism_id 13 --release octopus-T1 T1-tbro-table.tbl

42. **Import peptide sequences into Tbro database**

    tbro-import sequences_fasta --organism_id 13 --release T1-tbro-table.tbl T1-transcriptome.aa

IMPORT ANNOTATIONS

43. **Copy InterProScan tsv files into TBro Docker container**

    cd ../annotations

    scp id@address:/path/to/annotations/T1-annotations.tsv .

44. **Import InterProScan annotations into Tbro database**

    tbro-import annotation_interpro --organism_id 13 --release octopus-T1 -i T1-annotations.tsv

BUILD BLAST DATABASE SERVER

45. **Pull zipped blast databases and md5sums files into TBro Docker container**

    cd ../blastdbs

    scp id@address:/path/to/transcript/blastdbs/T1-blastdb-TR.zip .

    scp id@address:/path/to/peptides/blastdbs/T1-blastdb-AA.zip .

    scp id@address:/path/to/blastdb/md5sums/list-zipped-blastdb-md5sums .

46. **Copy zipped blast databases into Docker FTP container**

```
curl --data-binary --ftp-pasv --user $WORKERFTP_ENV_FTP_USER:$WORKERFTP_ENV_FTP_PW -T T1-blastdb-TR.zip

curl --data-binary --ftp-pasv --user $WORKERFTP_ENV_FTP_USER:$WORKERFTP_ENV_FTP_PW -T T1-blastdb-AA.zip ftp://"$WORKERFTP_PORT_21_TCP_ADDR"/
```

47. **Update the queue_config.sql**

```
cd /home/tbro

cp  queue_config.example.sql queue_config.sql

nano queue_config.sql

### BEGIN within file

-- database files available. name is the name it will be referenced by, md5 is the zip file's sum, download_uri specifies where the file can be retreived
INSERT INTO database_files (name, md5, download_uri) VALUES

### BEGIN text to add 1
### database_files( name ) is the name of the zipped blastdb file without '.zip' - and is referred to as program_database_relationships( database_name ) below

('T1-blastdb-TR', 'becd11699b54377106482fdc7f54906d', 'ftp://172.17.0.4/T1-blastdb-TR.zip'),
('T1-blastdb-AA', '52acbc6e906029a04ea3ac850bb75674', 'ftp://172.17.0.4/T1-blastdb-AA.zip');

### END text to add 1

-- contains information which program is available for which program.
-- additionally, 'availability_filter' can be used to e.g. restrict use for a organism-release combination INSERT INTO program_database_relationships (programname, database_name, availability_filter) VALUES

### BEGIN text to add 2
 ### availability_filter is ( organism_id )_( release ) from earlier steps

('blastn', 'T1-blastdb-TR', '13_octopus-T1'),
('blastp', 'T1-blastdb-AA', '13_octopus-T1'),
('blastx', 'T1-blastdb-AA', '13_octopus-T1'),
('tblastn', 'T1-blastdb-TR', '13_octopus-T1'),
('tblastx', 'T1-blastdb-TR', '13_octopus-T1');

### END text to add 2

### END within file
```

48. **Import configuration file**

```
PGPASSWORD=$WORKER_ENV_DB_PW psql -U $WORKER_ENV_DB_USER -h $WORKER_PORT_5432_TCP_ADDR -p $WORKER_PORT_5432_TCP_PORT <queue_config.sql
```

49. **Explore your TBro database**

Enter your Amazon AWS Lightsail static IP address (see Step 4) into a web browser

Enter your user and password information (see Step 26)

Explore