TRAC Deployment

# Introduction.

TRAC Deployment is designed for two environments development and production. The development environment is where TRAC is built from scratch into container images and a server is stood up for verification testing. The production server is then upgraded using the container images built in development. TRAC Deployment also manages the configuration and storage of TRAC using docker volumes. All of this is done using a set of NodeJS scripts which generate and run docker compose commands.

# Description of the script set.

node docker/backup-volumes.js -?

Options:

--version Show version number [boolean]

--production volumes for production [boolean]

-?, --help Show help [boolean]

This script copies all of the docker volumes into a zip file.

node docker/clean-docker-development.js -?

Options:

--version Show version number [boolean]

--removeVolumes remove all of the volumes [boolean]

-?, --help Show help [boolean]

This script resets the development environment and shuts down and remove the containers. Optionally, also remove the development volumes.

node docker/create-test-certs -?

Options:

--version Show version number [boolean]

--certdir local cert directory for the containers

[string] [default: "/Users/mbletzin/Work/actalent/trac/tmp/certs"]

-?, --help Show help [boolean]

Uses the docker version of omgwtfssl to generate self-signed certificates to test TRAC docker containers.

node docker/create-volumes.js -?

Options:

--version Show version number [boolean]

--pw password for the postgresql database [string] [required]

--tpw password for the mssql database [string]

--apihost hostname for the trac-api server

[string] [default: "localhost"]

--smtphost hostname for the SMTP server for emails

[string] [default: "localhost"]

--smtpport port for the SMTP server for emails[number] [default: 1025]

--configdir local configuration directory for the containers

[string] [default: "/Users/mbletzin/Work/actalent/trac/tmp/config"]

--certdir local cert directory for the containers

[string] [default: "/Users/mbletzin/Work/actalent/trac/tmp/certs"]

--certfile Certificate filename [string] [default: "cert.pem"]

--certkeyfile Certificate Key filename [string] [default: "key.pem"]

-r, --reset remove existing volumes and network if true [boolean]

-d, --data clear all data but keep the audits [boolean]

--production volumes for production if true [boolean]

-?, --help Show help [boolean]

This script creates and populates all of the volumes used by TRAC. The long list of parameters is used to synchronize the configurations of all of the containers.

node docker/restore-volume.js -?

Options:

--version Show version number [boolean]

--name name of the volume [string]

-?, --help Show help [boolean]

This script restores a volume from the back zip file. The old volume needs to be removed before running this script.

node docker/start-development.js -?

Options:

--version Show version number [boolean]

--pw password for the database [string] [required]

--certdir local cert directory for the containers

[string] [default: "/Users/mbletzin/Work/actalent/trac/tmp/certs"]

-r, --reset remove existing volumes and network if true [boolean]

-?, --help Show help [boolean]

-c, --configdir local configuration directory for the containers [string]

This script builds the containers starts the development servers.

node docker/start-production.js -?

Options:

--version Show version number [boolean]

--useVersion version to start [string] [required]

--configdir local configuration directory for the containers

[string] [default: "/Users/mbletzin/Work/actalent/trac/tmp/config"]

--certdir local cert directory for the containers

[string] [default: "/Users/mbletzin/Work/actalent/trac/tmp/certs"]

-r, --reset remove existing volumes and network if true [boolean]

-?, --help Show help [boolean]

This script starts the production servers from existing docker images.

# Building and deploying to development.

This section shows how TRAC is installed on the existing server slvstlappiold01.blamericas.bausch.com. The installation is located in the /iol directory.

## Folders and Servers on trac-dev.baush.com

/iol/certs – Directory containing the server certificates for trac-dev.baush.com

/iol/deployment-configs – Directory where the configurations are deployed and maintained.

/iol/build/apps/db-reset/src/assets = Directory containing TRAC data used to populate the database.

smtp.baush.com port 25 – server and port for TRAC to send emails.

slvstlappiold01.blamericas.bausch.com – SQL server used to retrieve data needed by TRAC.

## Deploying to Development:

Stop existing server and remove containers: node docker/clean-docker-development.js

Re-create new volumes for development: node docker/create-volumes.js --pw V3fx0si0qjfK --tpw Bausch@1234567 --certdir /iol/certs/ --certfile trac.bausch.com.crt --certkeyfile trac.bausch.com.key --apihost trac-dev.bausch.com --smtphost smtp.bausch.com --smtpport 25 --configdir /iol/deployment-configs -d

Build and start development server: node docker/start-development.js --pw V3fx0si0qjfK --certdir /iol/certs --configdir /iol/deployment-configs/ -r

Restart the development server: docker compose -f ./tmp/docker-compose-trac-development.yml restart

Monitor the development server: docker logs --follow trac-dev-api

Create backups for the development server. The backups are stored /iol/docker/lib/trac-backups - node docker/backup-volumes.js

Restore volumes to development - node docker/restore-volume.js --name trac-dev-pgdata and node docker/restore-volume.js --name trac-dev-uploads

## Deploying to Production:

Re-create new volumes for development: node docker/create-volumes.js --pw V3fx0si0qjfK --tpw Bausch@1234567 --certdir /iol/certs/ --certfile trac.bausch.com.crt --certkeyfile trac.bausch.com.key --apihost trac-dev.bausch.com --smtphost smtp.bausch.com --smtpport 25 --configdir /iol/deployment-configs -d --production

Build and start production server: node docker/start-production.js --useVersion 0.9.15 --pw V3fx0si0qjfK --certdir /iol/certs --configdir /iol/deployment-configs/ -r

Restart the production server: docker compose -f ./tmp/docker-compose-trac-production.yml restart

Monitor the production server: docker logs --follow trac-api

Create backups for the production server. The backups are stored /iol/docker/lib/trac-backups - node docker/backup-volumes.js --production