Docker image creation walkthrough & commands used

Create container with the following bind mount:
/usr/src/matomo_config/ /NETWORK_SHARE_LOCATION/matomo_mount/config

Network: host

Execute into the container as "root" user and run the following commands:

apt update

apt install cron unzip nano

cd /usr/src/matomo

curl -O
https://raw.githubusercontent.com/Starker3/matomo-plugins-download/main/download-matomo-plugins-only.sh

chmod +x download-matomo-plugins-only.sh

Edit the downloaded script file:

nano download-matomo-plugins-only.sh

Place your license key on the third line inside the quotations (LICENSE_KEY="xyz")

Save and close the file.

Execute the following:

./download-matomo-plugins-only.sh ${MATOMO_VERSION}
mv plugins-downloaded/* plugins/ && rm -R plugins-downloaded
rm download-matomo-plugins-only.sh

Configure the crontab using the following:
crontab -u www-data -e

* * * * * /usr/local/bin/php /var/www/html/console custom-matomo-js:update
* * * * * /usr/local/bin/php /var/www/html/console tagmanager:regenerate-released-containers

Save and close the crontab

Then execute the following:

CURRENT_DATE=`date +%Y-%m`

curl -O https://download.db-ip.com/free/dbip-city-lite-${CURRENT_DATE}.mmdb.gz
gunzip dbip-city-lite-${CURRENT_DATE}.mmdb.gz
mv dbip-city-lite-${CURRENT_DATE}.mmdb misc/DBIP-City.mmdb

If this is the first time setting up Matomo on this database, use the UI to complete the database setup. If this is for an existing database, skip this step.
Once complete execute the following:
mv /var/www/html/config/config.ini.php /usr/src/matomo_config/config.ini.php

If you are using an existing config.ini.php file, copy it to the /usr/src/matomo_config/ folder in the container

Next edit the entrypoint script:
nano /entrypoint.sh

In the last "if" block before the end of the entrypoint script, add this line:
ln -s /usr/src/matomo_config/config.ini.php /var/www/html/config/config.ini.php

Save the entrypoint script and close it

Edit the config.ini.php file to add our Multi-server config settings:

nano /usr/src/matomo_config/config.ini.php

[General]
; Tell Matomo we are running a multi server environment
multi_server_environment = 1

; Disable installing new plugins and (de)activating plugins.
enable_plugins_admin = 0

; Disable the one-click auto-update as we have multiple servers
enable_auto_update = 0

; Hide and disable in the UI the following settings:
; - Archiving settings
; - Update settings
; - Email server settings
; - Trusted Matomo Hostname
enable_general_settings_admin = 0

chown www-data:www-data /usr/src/matomo_config/config.ini.php

On the host machine where the container is running, execute the following to get the name of the container (If you named the container during first setup, then you can skip this step):

docker ps -a

While the container is still running on the host machine, we commit the changes that we made using the following:
docker commit RUNNING_CONTAINER_NAME NEW_IMAGE_NAME

Now you can start up your newly created image in a container with the following bind mounts:

/usr/src/matomo_config/ /NETWORK_SHARE_LOCATION/matomo_mount/config
/var/www/html/misc/user /NETWORK_SHARE_LOCATION/matomo_mount/user

Execute into the running container (That is using the new custom image) as the **www-data** user and execute the following command to set the license key for the Matomo Marketplace:

./console marketplace:set-license-key --license-key="xyz"

Next, activate all of the plugins we downloaded and added to the Matomo codebase with the following bash script:

for PLUGIN_TO_ACTIVATE in Cohorts CustomReports MarketingCampaignsReporting CustomAlerts LogViewer InvalidateReports TasksTimetable DBStats AbTesting MediaAnalytics FormAnalytics Funnels RollUpReporting SearchEngineKeywordsPerformance MultiChannelConversionAttribution HeatmapSessionRecording UsersFlow ActivityLog WhiteLabel TagManager; do echo -e “activating plugin $PLUGIN_TO_ACTIVATE…” && ./console plugin:activate $PLUGIN_TO_ACTIVATE; done

Connect to the Matomo UI through this newly setup container that is running the custom image and check that the UI is usable and that all the plugins are active.

Well done, you should now have a functioning Matomo database and container image that you can use for deployments! Just be sure to mount the correct shared folders for the config.ini.php and user files (I.e. logos, favicon, etc)