## QTemp Monitoring Actuator Node

### Description

This is a custom-built container based on Ubuntu 20.04 incorporating only Python modules for external connectivity rather than Ubuntu client applications such as MQTT Mosquito or OpenSSL.

The primary program is configured to run as a non-privileged user within the container to minimize the impact on the underlying OS if the service is compromised.

**Note:** Minimal networking tools were also included to support connectivity troubleshooting since a default Ubuntu container does not have even basic network testing, such as the ability to ping another server on the network. These packages can be removed if a smaller container is required without impacting the controller function.

**Software build list**

|  |  |
| --- | --- |
| File Name | Function |
| ssat3\_controllerClientV2\_2.Dockerfile | Build instructions for Docker |
| .env\_node*X* | Run time parameter file to be used during container startup where *X*  is the number of the node being created |
| Mcd a-node\_requirements.txt | Text file listing modules that must be installed during container build |
| manodeClient.py | Primary program interacting with the application controller using MQTT broker as the intermediary communication mechanism. All local state values are stored in a custom class object defined in the *controlUnit* module. |
| controlUnit.py | A customized python module to decouple the programing needed to interact with any electrical Input/Output (IO) required for the hardware on which the MA-Node is operating. |
| appEnc.py | A customized python module to decouple encryption key management and encryption functions from the primary program control logic. |

**Container Build Instructions:**

* Copy all files from the build list to a clean directory
* Use the docker build command to create a new image in the server's local Docker repository. Include a version tag to support better version control within the environment. (Note "." at the end of the command, it is needed to include all files within the directory in the build process.)
  + sudo docker build -t ssat3\_manode:2.0 -f ssat3\_manodeV2\_0.Dockerfile.

**Example of build process staging and initial build commands:**

Text

Description automatically generated

The build process can take a few minutes, look for the successfully built message and confirm the image name is correct.

A picture containing graphical user interface

Description automatically generated

An environment file is needed to generate environment variables within the container's operating system that will be used by the program to determine IP addresses of other network services, access credentials and so forth. This makes the container both reusable and allows for simple recovery if a core password is compromised, I.E., change the password on the remote service and restart the container with the updated environment file containing the new credentials.

Docker also supports virtual networking within the docker server itself, permitting nodes to communicate to services outside the server hosting the containers but does not allow communication between networks within the server by default. This docker feature has been implemented in the model to simulate different network VLANs communicating through a perimeter router or firewall.

Create networks using the docker network create command:

* sudo docker network create --subnet=10.10.150.0/24 net150

https://docs.docker.com/engine/reference/commandline/network/

Assign a container to a network during the docker run process:

* sudo docker run --net net150 -h manode5 --ip 10.10.150.50 --name manode5 --env-file .env\_node5 --add-host=mqbroker.qtemp.local:10.100.200.3 -i -t ssat3\_manode:2.0

<https://docs.docker.com/engine/reference/commandline/run/>

(Note, an MQTT user ID and password must be in place on MQTT broker for each MA-Node prior to running the container, the username and password must be defined in the unique env file created for that node.)

Once Docker networks and the required image are in place, instantiating additional containers requires only minor modifications to the docker run syntax. Once created, each container can be enabled using the docker start or docker stop commands. To view the messages printed by the container during active operation, use the docker attach command.

Example: Instantiating four additional MA-Node

sudo docker run --net net140 -h manode4 --ip 10.10.140.40 --name manode4 --env-file .env\_node4 --add-host=mqbroker.qtemp.local:10.100.200.3 -d -t ssat3\_manode:2.0

sudo docker run --net net130 -h manode3 --ip 10.10.130.30 --name manode3 --env-file .env\_node3 --add-host=mqbroker.qtemp.local:10.100.200.3 -d -t ssat3\_manode:2.0

sudo docker run --net net120 -h manode2 --ip 10.10.120.20 --name manode2 --env-file .env\_node2 --add-host=mqbroker.qtemp.local:10.100.200.3 -d -t ssat3\_manode:2.0

sudo docker run --net net110 -h manode1 --ip 10.10.110.10 --name manode1 --env-file .env\_node1 --add-host=mqbroker.qtemp.local:10.100.200.3 -d -t ssat3\_manode:2.0

The screenshot below shows manode3 was not running and was able to be reinitialized using the docker start command.

Graphical user interface

Description automatically generated with low confidence

Viewing the data output from manode3 could then be achieved through using the docker attach command, as shown in the screenshot below.

Text

Description automatically generated