

Application Name:		Version Number:	
Environment	Steps to Reproduce	Constraints	Result
Resource Limits	1. Compressed image file size <100MBytes		
Resource Limits	2. De-compressed image size (in docker images list) <500MBytes		
Resource Limits	3. Configuration file size <1MB		
Config	Steps to Reproduce	Constraints	Result
Certificates	1. Application should arrive to EA with a valid development certificate to ensure EA is able to perform tests efficiently and with the correct certificate awarded to the Third Party at the development stage <i>For more detailed information see Section 7, "Developing with the LV-CAP Virtual Machine" and the supporting document on where to find certificates</i>	Third Party obtained certificate	
Input Data	1. Ensure devtools is unpacked and play_csv.py is within the dev tools, this is done within the Virtual Machine <i>For more detailed information see Section 9, "Developing with the LV-CAP Virtual Machine" and the supporting document on where to find certificates</i>		
Playback Data	1. Suitable data has been supplied to EA <i>For more detailed information see Section 9.2, "Developing with the LV-CAP Virtual Machine"</i>	Real data	
Images	Steps to Reproduce	Constraints	Result
Image Limits	1. Run the command in the container docker images <ID> (e.g. eatl/utility or cse/m2collation), please note that docker replaces _ with a / 2. Observe the SIZE column and find the image value 3. The image value is within the 500MB resource limitations	Configuration has been performed correctly	
Image Name	1. Run the command in the container docker images <ID> (e.g. eatl/utility or cse/m2collation), please note that docker replaces _ with a / 2. Observe the TAG column and locate the associated image ID 3. The image version matches the file name		
Image Name	4. Run the command in the container docker images <ID> (e.g. eatl/utility or cse/m2collation), please note that docker replaces _ with a / 5. Observe the REPOSITORY column and locate the associated image ID 6. The Docker ID matches the file name (eatl/utility)		
MQTT.fx	Steps to Reproduce	Constraints	Result
Connectivity	1. Access the dev tools file using the ls and cd commands 2. Connect to the MQTT.fx marketplace by using the following command: ./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt --key eatl_tlsdevtools.key -i <filename>.csv 3. The MQTT.fx market place has connected successfully. Please remain connected to the marketplace for the rest of the test cases	MQTT configured correctly (See section 8, Developing with the LV-CAP VM)	
Errors	Steps to Reproduce	Constraints	Result
Subscribe	1. Using MQTT.fx, in the subscribe tab, subscribe to the topic "storage/response/<IID>" 2. Remain subscribed to this topic for the duration of your tests (you may colour code this black to ensure it stands out) 3. No errors are published <i>For more detailed information see Section 8.2.4, "LV Common Application Platform Public API"</i>	Connected to MQTT.fx and configured correctly	
Subscribe	4. Using MQTT.fx, in the subscribe tab, subscribe to the topic "storage/data/error/<IID>" 5. User is subscribed to topic		
Publish	1. Publish to the following (for step 1 in subscribe ONLY) "storage/request/<IID>" 2. Use the following payload: { "InstanceID": "error" } 3. Select "Publish" next to the text box 4. Topic is accurately and successfully published		
Observe	1. Switch to the Subscribe tab 2. There are no errors		
Observe	3. If there are errors, check the time stamp. These errors are pre-patches to fix said errors and have not been replicated	Errors displayed	
Status	Steps to Reproduce	Constraints	Result
Errors	1. Ensure you are still subscribed to the topic "storage/response/<IID>" 2. Unsubscribe to any other topic 3. No errors have published yet	Connected to MQTT.fx and configured correctly	
Subscribe	1. Using MQTT.fx, in the subscribe tab, subscribe to the topic "status/response" (you may colour code this so it stands out) 2. User is subscribed to topic <i>For more detailed information see Section 8.2.2, "LV Common Application Platform Public API"</i>		
Publish	1. Using MQTT.fx, in the publish tab, publish to the topics "status/request/<IID>" 2. Do not fill the payload 3. Select "Publish" next to the text box 4. Navigate back to the subscribe tab 5. Status request is published		
Observe	1. The container manager will send a status request every minute 2. Ensure The application is sending out a status response every minute		
Observe	3. The application does not send out status messages arbitrarily or randomly		
Observe	4. Observe the payload in the subscribe tab. The status message contains Valid JSON		
Observe	5. Observe the payload in the subscribe tab. The status message contains the correct members and types, <i>documented in Table 6 (Section 8.2.2), the LV-CAP API</i>		
Observe	6. Observe the payload in the subscribe tab. The status message containers the correct status value of 1 ("OK"), <i>as described in Table 7 (Section 8.2.2) of the LV-CAP API</i>		
Observe	7. Run the command: sudo service rc.local stop in the Virtual Machine (command line) 8. Wait one minute 9. Status responses halt		
Observe	10. Run the command: sudo service rc.local start to restart the service in the Virtual Machine (command line) 11. Status messages start sending again		
Observe	12. Real-time error subscription 1 is subscribed again upon restart		

Configuration	Steps to Reproduce	Constraints	Result
Errors	1. Ensure you are still subscribed to the topic "storage/response/<IID>" 2. Unsubscribe to any other topic 3. No errors have published yet	Connected to MQTT.fx and configured correctly	
Subscribe	1. Using MQTT.fx, in the subscribe tab, subscribe to the topic "status/response/<IID>" 2. Note down the JSON payload	Ability to note down JSON payload	
Subscribe	Note the JSON output payload here:	JSON payload noted	JSON
Publish	1. Using MQTT.fx, in the publish tab, publish to the topics "config/request/<IID>" 2. Do not fill the payload 3. Select "Publish" next to the text box 4. Navigate back to the subscribe tab 5. Config request is published		
PAUSE	Please wait for another status/response following the above steps	PAUSE	PAUSE
Observe	Note the JSON output payload here:	JSON payload noted	JSON
Observe	1. Observe the payload in the subscribe tab. The status message contains Valid JSON		
Observe	2. Observe the payload in the subscribe tab. The status message contains the correct members and types, documented in Table 6 (Section 8.2.2), the LV-CAP API		
Observe	3. Observe the payload in the subscribe tab. The status message containers the correct status value of 1 ("OK"), as described in Table 7 (Section 8.2.2) of the LV-CAP API		
Compare	1. Compare the two collect JSON payloads 2. The configuration is now additionally added to the status message		
Command	Steps to Reproduce	Constraints	Result
Errors	1. Ensure you are still subscribed to the topic "storage/response/<IID>" 2. Unsubscribe to any other topic 3. No errors have published yet	Connected to MQTT.fx and configured correctly	
Subscribe	1. Using MQTT.fx, in the subscribe tab, subscribe to the topic "command/<IID>" 2. In the subscribe tab, subscribe to the topic "status/response/<IID>" 3. User is subscribed to topic		
Publish	1. Using MQTT.fx, in the publish tab, publish to the topics "command/<IID>" 2. Fill the payload with the following: { "Command": 1 } 3. Select "Publish" next to the text box 4. Navigate back to the subscribe tab 5. Config request is published		
PAUSE	Please wait for another status/response following the above steps	PAUSE	PAUSE
Observe	1. Observe the payload in the subscribe tab. The status message contains Valid JSON		
Observe	2. Observe the payload in the subscribe tab. The status message contains the correct members and types, documented in Table 6 (Section 8.2.2), the LV-CAP API		
Observe	3. Have the correct status value of 4, as described in Table 7 (Section 8.2.2) of the LV-CAP API. This is the numeric code for "Shut Down" status.		
Publish and Subscribe	Steps to Reproduce	Constraints	Result
Errors	1. Ensure you are still subscribed to the topic "storage/response/<IID>" 2. Unsubscribe to any other topic 3. No errors have published yet	Connected to MQTT.fx and configured correctly	
Subscribe	1. Using MQTT.fx, in the subscribe tab, subscribe to the topic "algorithm/data/<IID>" 2. Using MQTT.fx, in the subscribe tab, subscribe to the topic "sensor/data/<IID>" 3. User is subscribed to topics. It is advised to colour code these		
Publish	1. There is no need to publish to anything 2. In order to avoid waiting for data, ctrl+Z on the command line to terminate the connection 3. Then run <code>./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt --key eatl_tlsdevtools.key -i <FileName>.csv</code> whilst remaining subscribed to alorirhtm data		
Observe	1. Check with the user manual or application form what topics require to be pushed and subscribed to (e.g. algotirhm data) 2. Observe the payload in the subscribe tab. The status message contains Valid JSON		
Observe	3. Observe the payload and name match information in the User Guide (e.g. calculating current mean)		
Observe	4. Repeats steps five times each with a different algorithm or status subscription 5. Ensure all results match what is described in the user manual		
Updates	Steps to Reproduce	Constraints	Result
Errors	1. Ensure you are still subscribed to the topic "storage/response/<IID>" 2. Unsubscribe to any other topic 3. No errors have published yet	Connected to MQTT.fx and configured correctly	
Subscribe	1. Using MQTT.fx, in the subscribe tab, subscribe to the topic "config/response/<IID>" 2. Using MQTT.fx, in the subscribe tab, subscribe to the topic "status/response/<IID>" 3. User is subscribed to topics. It is advised to colour code these		
Steps	1. Using an SSH session, open the application's configuration file (e.g. eatl_product_00.json) and edit the data in some way, it is advised to edit the period as this can be checked against timestamps 2. Save the configuration and use the <code>cp</code> command to copy the updated .json file to the /tmp/LVCAP_config/	SSH connection made	
PAUSE	Leave up to five minutes for the container manager to pick this change up	PAUSE	PAUSE
Observe	1. Ensure the new period is being followed (Optional)		
Observe	2. Using MQTT.fx, observe the "status/response/<IID>" outputs, the JSON payload for one of the subscriptions should have a value of 6, aka STATUS_RESTART		
Observe	3. Using MQTT.fx, observe the "config/response/<IID>" subscription		
Documentation	Steps to Reproduce	Constraints	Result
Application Run	1. The Application performs as expected and as described in the Third-Party Application Form	Third-Party has provided documents	
Additional Tests	Steps to Reproduce	Constraints	Result
Restarting	1. Restart the docker using the command "docker retstart <IID>" 2. Status message of 2		
Final Test	Steps to Reproduce	Constraints	Result
Final Test	1. Unsubscribe to everything except "storage/reponse/<IID>" 2. User is left with no messages	Connected to MQTT.fx	