

CNTF - Containerized Network Testing Framework





Business Relevance



CNTF is a testing environment designed for developers to evaluate open-source 5G cores.

- For **developers**, CNTF offers valuable resources to enhance 5G core development and make informed comparisons of existing solutions. It will give valuable insights into these solutions, potentially reducing the cost of provisioning private 5G network(s) by taking advantage of open-source applications.
- For **enterprises**, CNTF provides assurance that they have selected the right 5G core for their specific needs.



Project Refocus



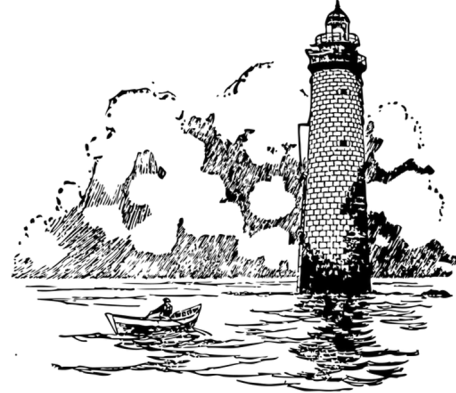
Originally, this BOAT aimed to build test frameworks around three different open-source 5G cores: Open5GS, Magma, and Free5GC. However, after thorough research, it became clear that only one was a viable option: Open5GS.

Review of Magma and Free5GC:

- Magma: Magma is no longer actively developed or supported by either Facebook or the Linux Foundation. Additionally, it relies on a deprecated version of Kubernetes.
- Free5GC: While potentially valuable, Free5GC proved challenging to integrate with Kubernetes due to its numerous dependencies, making it difficult to test effectively.



Customer Focus



Customer: LiveView

Main takeaways from conversations with LiveView:

- Desires for a 5G network:
 - Rapid UE Connections (< 60 seconds).
 - Consistent Upload Speeds (3MBps - 12MBps) regardless of network traffic.
- Other:
 - Automated Testing (Simulate VPN connections, latency, traceroutes, network connection times, cradlepoint reboots, and carrier change speed).
 - Real-time, observable network data.



Objective Key Results



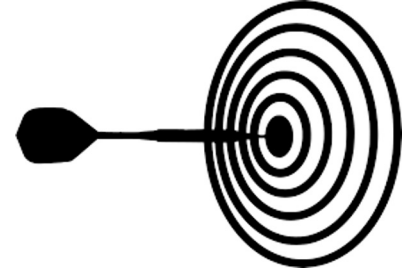
Hypothesis: "Testing Open5GS for load, latency, and QoS will uncover its inadequacy in meeting the network requirements of our customer, LiveView."

OKRs:

- 1: Create an automated process of deploying open-source 5G core(s), UERANSIM (test-suite), and out of the box tests.
- 2: Develop six reusable test cases focused on load, latency, and QoS, to evaluate Open5GS as a potential 5G core for LiveView.
- 3: Create three dashboards visualizing network data in real-time.



OKR Outcomes



OKR 1: 70%

- Successfully automated the deployment of infrastructure, apps, and test cases via Gitlab CI pipelines.
- Some test cases that LiveView had in mind were not accounted for, such as VPN connections and trace routing, speed of carrier changes, etc.

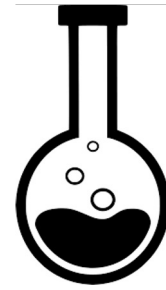
OKR 2: 75%

- Successfully developed five reusable test cases to evaluate the load and latency performance of Open5GS
- Could not test QoS because Open5GS does not support this functionality.

OKR 3: 85%

- Effectively visualized live test data by creating five Coralogix dashboards.
- The visualization of test data is near real-time but could be further optimized

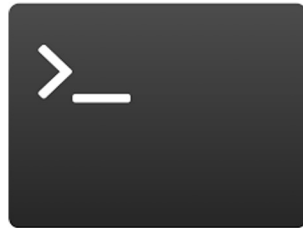
Test Cases





Methodology

CNTF test cases were crafted and executed using bash scripts and UERANSIM. Originally designed for Open5GS, with minor adjustments, they can also run on other open-source networks that support UERANSIM integration and containerized environments with bash shells.





Test Case - UE Subscription Test

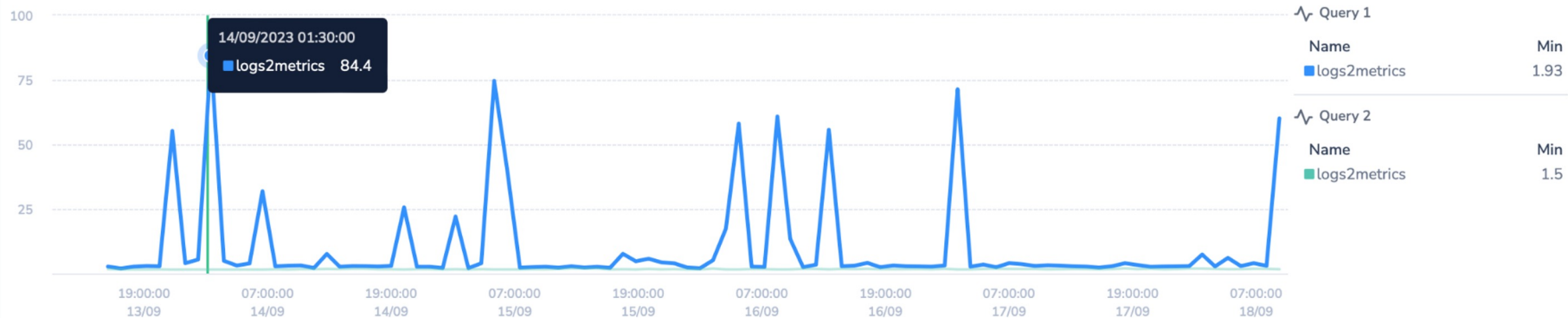


- **Description:** Connect new UEs to Open5GS every five minutes.
- **Purpose:** Evaluate Open5GS's response to continuous new connections and its ability to meet LiveView's requirement of connecting to a new network in under 60 seconds.
- **Outcomes:**
 - Open5GS inconsistently subscribed UEs in under 60 seconds.
 - Open5GS limits its simultaneous UE connections to approx. 3,000 devices. After that, its database blocks new connections.



METRICS

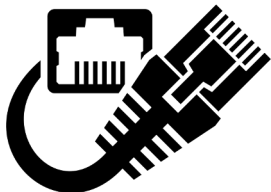
UE Creation Time





Test Case - Smoke Test

- **Description:** Compare the speed of Open5GS to the internet by simultaneously executing curl commands over 5G network interfaces and an ethernet interface.
- **Purpose:** Evaluate Open5GS's baseline latency performance.
- **Outcome:** Open5GS consistently showed slightly higher latency compared to the internet when accessing websites.





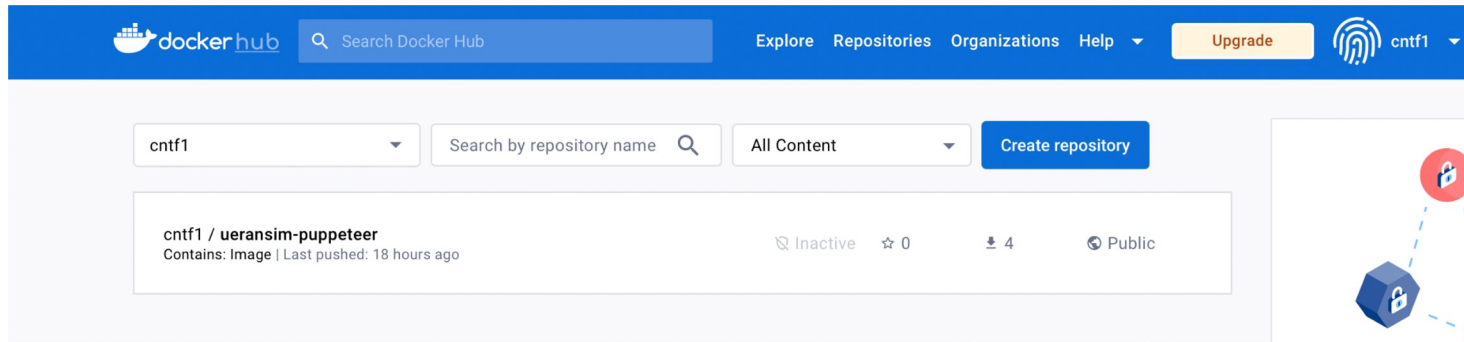
```
:38,"speed upload":0,"ssl verify_resu.  
10421,"time total":0.210511,"url":"ht  
nghttp2/1.47.0","test":"over5g"}
```

```
9,"speed upload":0,"ssl verify_resu.  
093,"time total":0.07333,"url":"https:/  
nghttp2/1.47.0","test":"overinternet"}
```

UERANSIM Customization

UERANSIM + Puppeteer

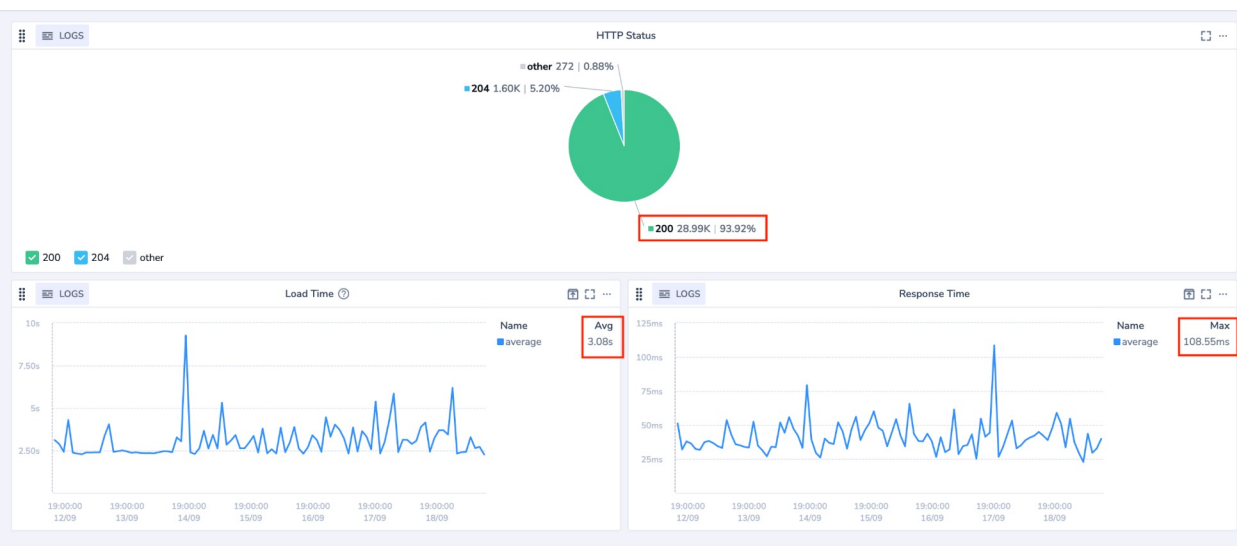
Description: Customized the UERANSIM test suite to enable UEs to perform various user activities (e.g. watching YouTube, browsing websites, etc.). This was achieved by enhancing UERANSIM's Docker image with Chromium, Node.js and unique JavaScript files.





Test Case - YouTube Test

- **Description:** Simulate a UE watching a YouTube video while connected to the 5G network.
- **Purpose:** Determine how well Open5GS supports data streaming activities on a UE (e.g. video playback).
- **Outcome:** Open5GS demonstrated good video streaming performance with minimal HTTP errors and consistent response/load times.
- **Metrics:**
 - screenshot of video
 - total test load time
 - network requests (Records load times for individual network requests during the test, with a focus on critical requests among the approximately 50 made during YouTube video playback)



cntf-open5gs-test-results.s3.us-east-1.amazonaws.com

youtube...

YouTube

Iron Maiden Invaders X

Sign in

T-Mobile
t-mobile.com/home-internet
Switch now
\$50/MONTH
w/ AutoPay discount using eligible payment method
Ad 1 of 2 · 0:10 · T-Mobile.com/t-mobile/enterprise

Iron Maiden - Invaders Lyrics

LyricsWorldHUN
14.8K subscribers
Subscribe

1.2K

Share

123K views 10 years ago



Test Case - Load Test

- **Description:** Load Open5GS with thousands of UEs to simulate heavy network traffic.
- **Purpose:** Determine the impact heavy network traffic has on Open5GS (e.g. do NF's scale?, how many UE subscriptions can be made?, etc.) .
- **Outcome:**
 - Open5GS subscribes up to approximately 3,000 UEs
 - Network interface provisioning for UEs exhibited sporadic behavior
 - No scaling of network functions was identified

The screenshot shows the 'Subscriber' management page in the Open5GS interface. It features a sidebar with navigation options: Subscriber, Profile, and Account. The main area contains a search bar and a table listing 30 subscriber IDs, organized in 10 rows and 3 columns. A red circular button with a white plus sign is located at the bottom right of the table.

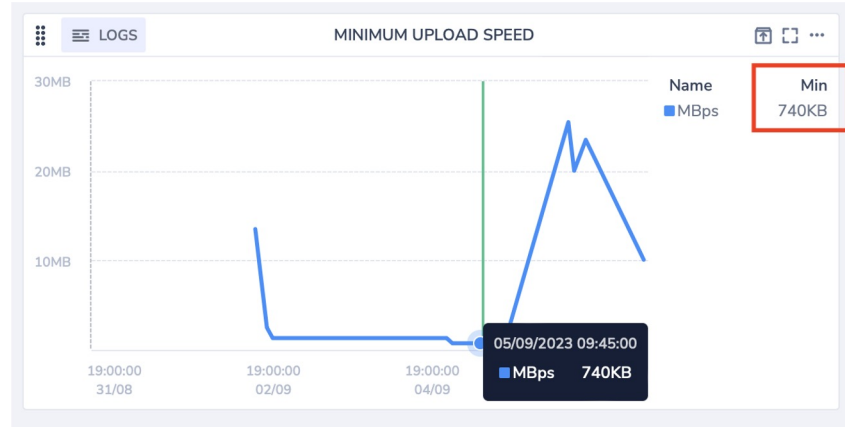
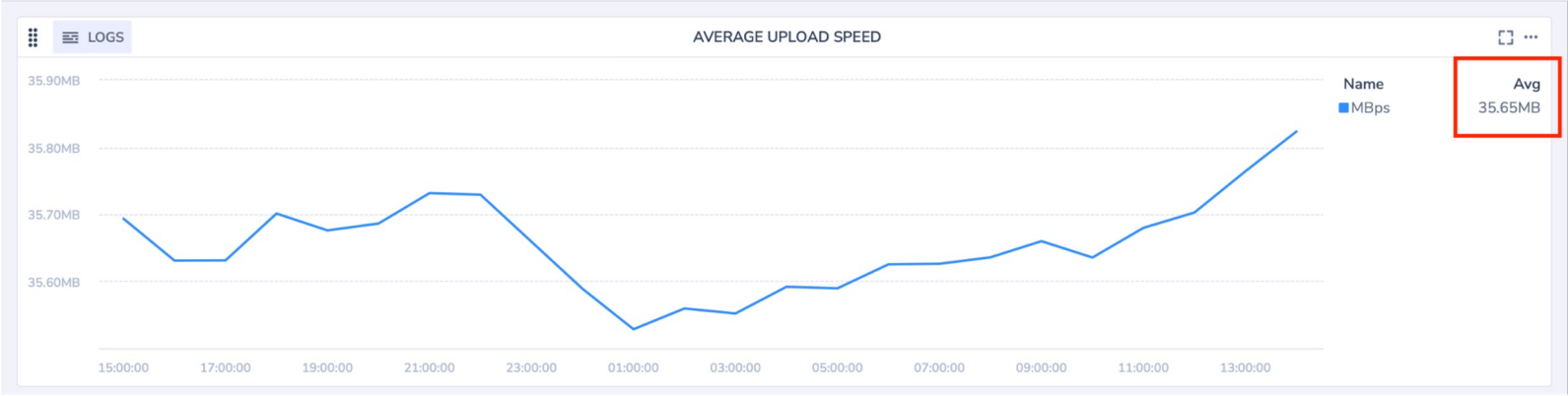
Subscriber		
1002897630	1003236331	1003405610
1004421168	1005267635	1012207388
1013222856	1013900469	1016608206
1019147655	1022363302	1024056153
1027441130	1038781938	1043691179
1046060486	1054524216	1056047738
1058924550	1059432609	1065357053
1066372825	1068742286	1070096767
1075174001	1077035943	1081098770
1086515135	1087700199	1088376910
1093793564	1097178539	1098194878



Test Case - Stress Test



- **Description:** Upload data to a webserver while Open5GS handles thousands of UE connections. Calculated the estimated maximum upload speed per second with a unique formula.
- **Purpose:** Evaluate Open5GS's ability to meet LiveView's minimum upload speed requirement of 3MBps-12MBps under network traffic.
- **Outcomes:**
 - With minimal traffic, Open5GS consistently supports upload speeds of approximately 35 MBps.
 - With heavy traffic, (approx. 3,000 UEs) there was a point in time where upload speeds dropped to 740 kB/s.





Coralogix Dashboards

Methodology:

1. Captured logs and metrics from tests run in Open5GS/UE-RANSIM pods and stored them locally via bash scripts.
2. Utilized Python to transform local files into S3 objects within a dedicated bucket.
3. Set up a Lambda function to continuously transfer data from S3 bucket to Coralogix.
4. Parsed and converted logs into metrics to facilitate the creation of dashboards displaying data from test runs.



Conclusion



Hypothesis: "Testing Open5GS for load, latency, and QoS will uncover its inadequacy in meeting the network requirements of our customer, LiveView."

Conclusion: "After completing our test cases, it is evident that Open5GS does not meet LiveView's customer requirements."



Conclusion

1. Open5GS inconsistently subscribed UEs in under 60 seconds.
2. Upload speeds did not meet the 3MBps requirement when the network faced heavy traffic.
3. The database for Open5GS (MongoDB) consistently had network errors which greatly impacted the efficiency of connecting UEs to the 5G core.

```
17006 MongoNetworkError: connect ECONNREFUSED 10.100.68.248:27017
17007 command terminated with exit code 1
✓ 17008 Cleaning up project directory and file based variables
17009 ERROR: Job failed: command terminated with exit code 1
17010 mand terminated with exit code 1
```