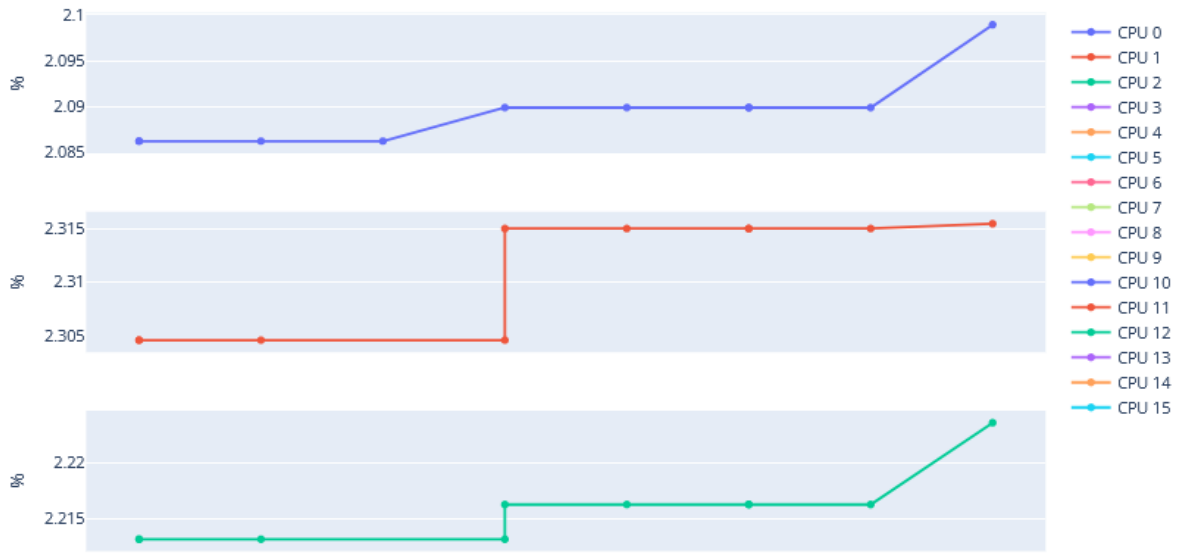
Some images of the live dashboard can be viewed below. It is a long page, as we have one graph for each CPU, so here we only show part of those graphs.

Dashboard

Live monitoring dashboard with data from Redis

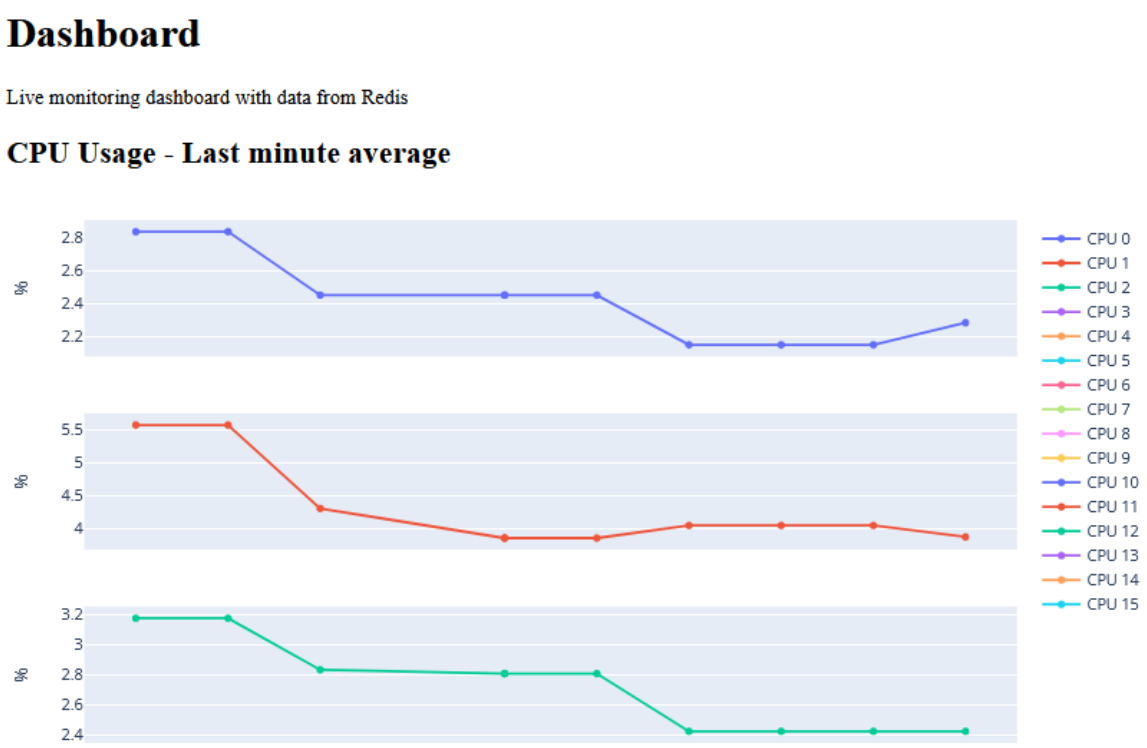
CPU Usage - Last minute average

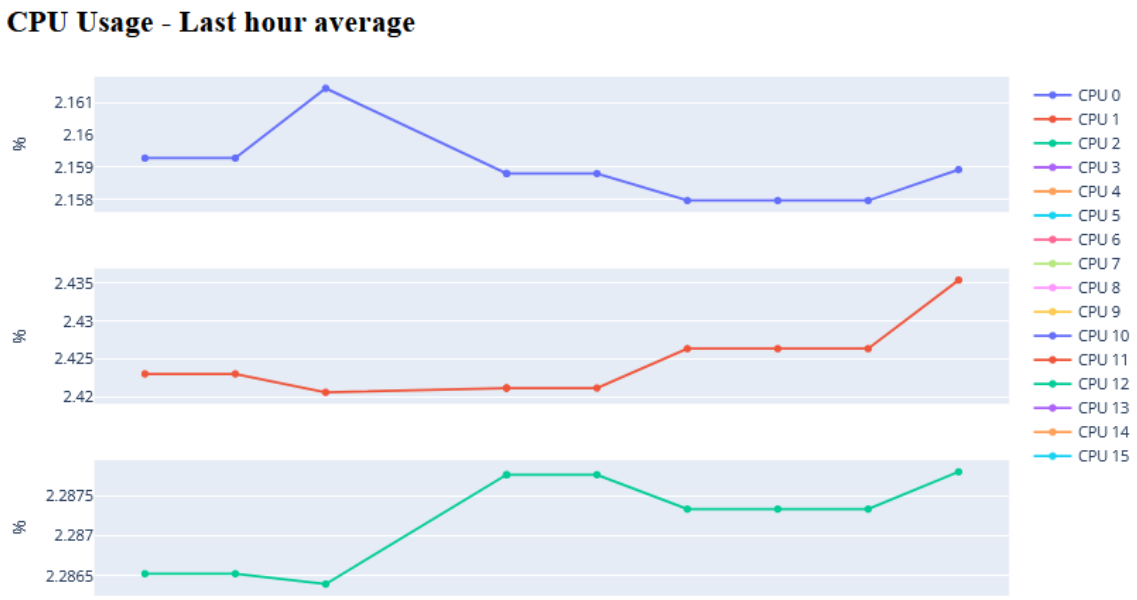
CPU Usage - Last hour average



Further down below, we find the last graph, with monitors the virtual memory usage:

Considering that the callbacks are done every five seconds, the first values take some seconds to be actually plotted. Initially, the last minute average is very similar to the hourly one, however, after around a minute, they become different:





As it is a machine with a lot of CPUs, during the test conducted, we rarely found a single core with more than 15% of usage at a time, highlighting the good job of the processor in trying to keep loads uniform and low. Another reason for this result is that no highly intense computation was performed during the testing. The memory usage follows the same observations, with a very constant read throughout the monitoring.

One implementation detail worth noting is that running the dashboard inside a kubernetes pod adds a step of complexity in verifying if the service is working. In this case, we had to do two SSH tunnels, one between the local machine and the cloud, and another between the cloud and the pod, so we could use the page in the local browser.

Running inside the docker only had a similar problem, but was resolved more easily. There was a need to use a parameter “net” when running, and connect the container to the “bridge” network, as already defined in the docker networks available.