

# Monitoring/Diagnostic machinery for EVE

## Problem statement

We are required to offer the customer a mechanism for monitoring and diagnostics in case the downloaded application hangs in EVE.

## 1st iteration

At the moment we have a mechanism for obtaining logs from a running application, for ex.: ``eden pod logs --fields=app --format=json <application>``

Test for getting logs in case of kernel panic:

<https://github.com/lf-edge/eden/pull/492>

For monitoring, the simplest option is to put labels in the logs, eg. using the ``logger`` command by the ``cron`` service in the application's image.

## 2nd iteration

Embedding some monitoring agents in virtual machine images (for example, QEMU Guest Agent, Chef, Puppet, Ansible, SaltStack, Terraform, etc.). This generally does not solve diagnostic problems in case of kernel panic.

One of the possible monitoring implementations can employ a custom ACPI SSDT table which will have a table region along with code filling it with diagnostic data. Qemu can generate periodic events which will be handled by the code in question.

## 3rd iteration

Developing of a monitoring / diagnostics subsystem within EVE.

For this, it will make sense to add a procedure for binding monitoring/diagnostic scenarios to running applications.

Further, the monitoring system in EVE launches the corresponding monitoring scenario with the specified frequency and waiting time. If it does not complete successfully, a diagnostic script is run.

These scenarios can use mechanisms for interacting with VMs through one or another transport (ranked according to the degree of availability):

1. Console
2. Disk drive
3. Network

For example, in the case of providing access to the VM via the console, the simplest monitoring tool can be, for ex. periodically run ``uptime``. If there is no response to the entered command, a diagnostic script is launched that can receive certain data from the Linux kernel (task-states, blocked-tasks, backtrace-all-active-cpus, dump-ftrace-buffer, memory-usage, timers, registers...) via the console SysRq and/or GDB call, the state of the virtual machine (eg via QMP, some data about the state of the host EVE itself (eg about the state of the same devices).

The received data is sent to the controller where it can be processed by some customized analyzer.