metal3-io / **metal3-docs** Public

Code Issues 15 Pull requests 9 Actions Projects Security Insights

metal3-docs / design / baremetal-operator / **bmc-events.md**

tuminoid fix some more spelling issues ✓ 2547fc3 · 5 months ago

145 lines (107 loc) · 5.05 KB

Preview Code Blame Raw

Event Subscription API

Users of bare metal hardware may want to receive events from the baseboard management controller (BMC) in order to act on them in the event of a hardware fault, increase in temperature, removal of a device, etc.

Status

implemented

Summary

The Redfish standard includes the ability to subscribe to events, which will cause hardware events to be sent in a particular format to a destination URI. This design document describes a Metal3 API for configuring a subscription. While Redfish is the primary target for this design, the Ironic API is vendor-neutral and seeks to provide a unified interface for configuring events.

Motivation

Some environments run workloads that need to deal with potential faults or environmental changes quicker than they would get an alert through other channels. For example, some workloads may have a sidecar container that knows how to deal with an alert that a particular network interface went down, or that the CPU temperature reached a certain threshold.

Goals

- Provide an API to manage subscriptions to events

Non-Goals

- Configurable events and thresholds
- Any kind of event polling
- Software for processing the events, i.e. any webhook
- BMC's beyond Redfish for now

Proposal

User Stories

- I'd like to configure my BMC to send events to a destination URI.
- I'd like to provide context to a particular event subscription.
- I'd like to provide arbitrary HTTP headers
- I'd like the BareMetal Operator to reconcile on the BMCEventSubscription resource, and ensure its state is accurate in Ironic.

Design Details

Implementation Details

```
apiVersion: metal3.io/v1alpha1
kind: BMCEventSubscription
metadata:
  name: worker-1-events
spec:
   hostName: ostest-worker-1
   destination: https://events.apps.corp.example.com/webhook
   context: “SomeUserContext”
   httpHeadersRef:
     name: some-secret-name
     namespace: some-namespace
status:
  errorMessage: ""
  subscriptionID: aa618a32-9335-42bc-a04b-20ddeed13ade
```

- A BMCEventSubscription resource represents a subscription to the events generated by a specific BMC.
- Ironic will manage configuring the subscription using a vendor passthru API.
- The BMCEventSubscription will maintain a reference to a BareMetalHost.
- The BMCEventSubscription will allow injection of headers using a reference to a secret, for example to provide basic auth credentials.

- The BMCEventSubscription will reside in the same namespace as the referenced BareMetalHost.
- The BMCEventSubscription will maintain a reference to the subscription ID obtained from the BMC.
- The BareMetal Operator binary will be expanded to include an additional reconciler with a dedicated controller/reconcile loop for BMCEventSubscriptions.

Open Questions

Risks and Mitigations

Thundering herd

Large numbers of events across large numbers of BareMetalHosts could generate a lot of traffic. Users can control how many events their webhook receives by configuring the alert thresholds out of band.

Dependencies

Test Plan

There is some existing POC code for working with Redfish Events; we could build on this to implement a test framework for BMC events. We could also consider modifying sushy-tools to support emulated eventing.

Upgrade / Downgrade Strategy

Not required as this is a new API being introduced

Alternatives

Configurable events and thresholds

This API is for subscribing to events of a pre-defined type. In cases where no particular type is available, users would need to configure it out-of-band. For example, one may want to have a TemperatureOver40C alert that monitors the enclosure's temperature.

The Redfish standard itself does not seem to have a way to specify specific alerts and thresholds. For example, to receive an alert when the temperature exceeds 40C, one would need to configure this manually according to the vendor's recommendations.

Vendors, however, do provide vendor-specific ways to configure these thresholds, but it's hard to abstract to a neutral interface. For example, here is a Dell example for temperature.

In the short term, Ironic has no plans to abstract the various vendor implementations (if they exist at all).

References

- Ironic Vendor Passthru for Subscriptions
- Supermicro Redfish Guide
- DMTF: Redfish Eventing
- Redfish Event Controller (POC)
- Redfish Event Experiment (POC)