# P0 - Monitoring

**APPROVED, delivered on Oct 31.**

**EPIC (**[**HRM-1050**](https://jira.mheducation.com/browse/HRM-1050)**)**: As an application team, I need an automated solution that enables monitoring of cloud-based resources so that I'm enabled to create custom dashboards, reports and alerts and respond in a way that reflects the needs of my customer.

## ****What is the corporate strategic benefit?****

Standardized monitoring enables reliable and repeatable processes. Application teams can use standardized monitoring to generate a unified view of their entire stack. This enables application teams to optimize their performance and responds in a way that reflects the needs of their customers.

## ****What is the problem being solved?****

MHE does not have a standard for enabling application monitoring for cloud-based resources.

## ****What is the customer (developer/application team) benefit?****

1. DevOps delivery of standardized monitoring for all application teams.
2. Monitoring across the entire stack - from the hypervisor and host through the container and to the application itself.
3. With monitoring enabled across the entire stack, metrics are gathered and teams can use reports to evaluate the performance of their software via:
   1. Using default dashboards - provided out-of-the box.
   2. Creating customized dashboards - edit the existing dashboard or create a new dashboard to focus on specific metrics.
   3. Creating alerts - proactively set resource thresholds and configure notifications (email, IM, etc).

## ****What is the capability?****

1. Full-stack monitoring occurs at three levels account, container and application.
   1. Account-level monitoring will be enabled by DevOps engineering via automation with zero work done by application teams. Specifically, this is monitoring at the virtual host level (hypervisor, OS).
   2. Container-level monitoring will be enabled by DevOps via 'sample' code. This code provides automated agent installation for a container and is intended to be modified by application teams for use with their specific AWS infrastructure (account ID, credentials, etc) to enable monitoring at the container-level (OS, container, application). Ex:
      1. Cluster Utilization Metrics
      2. Cluster Reservation Metrics
      3. Service Utilization Metrics
   3. Application-level monitoring will be enabled by DevOps via documentation. This enables instrumentation of the application to send custom metrics to a monitoring service.

## ****What are we doing?****

1. For the Account-level, DevOps is creating automation, process and documentation for:
   1. Updating existing MHE Turbot AWS accounts to enable monitoring.
   2. Provisioning new MHE Turbot AWS accounts during the on-boarding process to enable monitoring.
2. For the Container-level, DevOps is creating sample code and documentation for application teams.
3. For the Application-level, DevOps is creating sample code and documentation for application teams.

## ****How are we doing it?****

[Datadog](https://www.datadoghq.com/product/)is a monitoring service for cloud-scale applications, bringing together data from servers, databases, tools, and services to present a unified view of an entire stack. These capabilities are provided on a SaaS-based data analytics platform.

1. Enable metrics collection by Datadog at the account-level using automation:
   1. Updating - Python.  **DONE:**
      1. [**HRM-1053**](https://jira.mheducation.com/browse/HRM-1053)**, code run via a Jenkins job with approval from application team (non-Prod) and CAB (Prod).**
      2. [**Automating DataDog AWS Integration**](file:///C:\display\CICD\Automating+DataDog+AWS+Integration)
   2. Provisioning - Python.  **DONE:**[**Automating DataDog AWS Integration**](file:///C:\display\CICD\Automating+DataDog+AWS+Integration)**.**
2. Enable metrics collection by Datadog at the container-level using automation:
   1. Terraform sample code  **DONE -**
      1. [**Configuration of Data Dog for EC2 Container Service (ECS) Using Terraform**](file:///C:\display\CICD\Configuration+of+Data+Dog+for+EC2+Container+Service+(ECS)+Using+Terraform)
      2. [**Datadog-Docker Integration**](file:///C:\display\CICD\Datadog-Docker+Integration)
3. Enable metrics collection by Datadog at the host-level using automation:
   1. Automated Datadog Agent Installation via User Data script - **DONE**:  [HRM-1082](https://jira.mheducation.com/browse/HRM-1082), [HRM-1083](https://jira.mheducation.com/browse/HRM-1083), [HRM-1084](https://jira.mheducation.com/browse/HRM-1084), [HRM-1085](https://jira.mheducation.com/browse/HRM-1085), [HRM-1086](https://jira.mheducation.com/browse/HRM-1086). Instructions for applying script in Terraform listed under '[Code Snippets/Template Library](https://confluence.mheducation.com/display/CICD/Datadog)'.
   2. Also fulfills Datapipe on-boarding requirements by installing the following agents:  **DONE**
      1. Datadog
      2. Alertlogic IDS
      3. NOD32 Antivirus
      4. Tivoli BigFix BES
4. Enable metrics collection by Datadog at the application-level using automation:
   1. Terraform sample code - **Won't Fix as teams are using New Relic for APM.**