# Defining Observability & Monitoring

## Objective:

To clearly define the differences, objectives, and results of sound observability and monitoring.

## Questions:

* Have you ever thought about the differences between observability and monitoring?
* What do you think about these definitions?
* Do you agree with the definitions, or do you disagree? Why or why not?
* Do you or your company have your own definitions? If so, what are they?
* How well do you feel that everyone on your team understands the differences?
* In terms of monitoring and observability practices, in general, where do you feel your teams align? Where do they diverge?
  + Why do you feel this is the case?

## Lead:

When it comes to observability, we want to help the customer understand the various architectural layers of monitoring their workloads.

# Understanding the Full-Stack

## Objective:

To understand all components that make up a workload and the architectural layers of what should be monitored.

## Questions:

* How well do you believe your team fully understands all elements that comprise each workload within your company?
* Who “owns” each level?
  + Is it app team responsible for their entire stack?
  + Are there dedicated teams that are responsible for each level (e.g., infrastructure team monitoring services and networking, while app teams solely focused on application).
  + Or is it a hybrid?
* Do you have clearly defined boundaries of ownership and responsibility?  
  If so,
  + What are they?
  + How well do your teams honor boundaries and respect responsibilities?
    - Why do you feel this way?
* Have RACI charts been defined for each level?  
  If so,
  + Is everyone aware of the proper procedures for escalation?
  + How well do your teams follow these procedures?
    - Why do you feel this way?
    - When do you feel that they don’t?

If not,

* + Is this something you are in the process of defining?
    - If so, how far along are you and when do you anticipate completion?
    - If not, what is preventing you from doing so and how can Microsoft help?

## Lead:

Workloads are complicated and, depending on their components, various services should be monitored—not just the application.

# E2E Monitoring

## Objective:

To be aware that a single application may involve multiple services, and therefore each service should be monitored.

## Questions:

* Are you aware of Azure Landing Zones?
  + If yes, how much are you currently using Azure Landing Zones?
* Are you familiar with the Microsoft Azure Well-Architected Framework?
* Are your teams familiar with the Microsoft Azure Well-Architected Framework?  
  If yes,
  + How well do you believe they are following its principles?
  + How often do they reference Microsoft’s Cloud Design principles?
* In general, what Azure services do you typically use?
* Besides individual services such as VMs or App Services, what other advanced workloads are you running in Azure (e.g., SAP, AVD, AVS, etc.)?
* How confident are you that each service and/or workload are logging telemetry at their various levels?
* What would you say in terms of percentage that all of your production workloads are adequately instrumented for observability and monitoring?
  + Why do you feel that the percentage is so low/high?
  + What do you feel is needed in your teams to increase that percentage?
  + How can Microsoft help?

## Lead:

Microsoft offers many tools and resources to help customers monitor their workloads.

# Microsoft’s Observability Suite

## Objective:

To gain awareness of the tools and resources that Microsoft provides for monitoring their workloads.

## Questions:

* How familiar are you and your teams with Microsoft’s monitoring stack?
* How well do you feel that you and your teams understand the differences between Azure Monitor, Log Analytics, and App Insights?
* What tools are you currently using to monitor your cloud and on-premises environments?  
  If not Azure Monitor/Log Analytics:
  + What made you choose [tool] over Azure Monitor?

For any tool:

* + How happy are you with it? Why?
  + What are some of the pros and cons you’ve found with it?
  + In what areas do you believe it could be improved?
* What tools are you currently using to notify you of incidents within your environment?
* Have you configured alerts within your Azure tenant?

If so,

* + What types of alerts have you configured?
  + How well is your infrastructure covered?
  + What gaps do you believe still exist in your alert coverage?
  + How well does your team respond to alerts?

If not,

* + Why not?
  + What plans do you currently have to implement alerts?

## Lead:

Once the workload is instrumented properly, companies must ensure they are using the right tools to report telemetry.

# Operational Insights vs. Business Insights

## Objective:

To understand the differences between operational and business insights, and to understand what tools should be used.

## Questions:

* How well do you believe your teams understand the differences between operational insights and business insights?
  + Why do you feel this way?
  + In what areas can they improve?

These questions *assume* the customer is currently monitoring their workload.

* How well do you feel like the right tools are being using for reporting operational telemetry?
* What tool(s) are you using to report your telemetry?
  + How happy are you with it? Why?
  + What are some of the pros and cons you’ve found with it?
  + In what areas do you believe it could be improved?
* Do you feel you have collected adequate telemetry to provide enough visibility into the workload?
  + What insights do you feel your teams do well at collecting?
  + What insights do you feel your teams need to pay more attention to?
* What are some incidents that you felt your team responded well to based on the telemetry that has been collected?
* What are some incidents that you felt your team didn’t respond well to based on the lack of telemetry?

## Lead:

All companies fall somewhere within the Cloud Observability Maturity Model. It is important to understand where customers are currently and their opportunities for improvement.

# Observability Maturity

## Objective:

To assess a customer’s level within the Cloud Observability Maturity Model and identify opportunities for growth.

## Questions:

* Where do you believe your teams/company fall on this scale?
  + Why do you believe that?
* Do you have SLOs defined for your workloads?  
  If so,
  + What are they (SLAs, RPOs, RTOs)?
  + Are you measuring SLIs (how well your applications and processes are meeting your SLAs)?  
    If so,
    - How well are you doing?
    - What tools are you using to measure your SLIs?
    - What areas need improvement, or where do you recognize gaps?
* What are you currently doing to improve your maturity?

## Lead:

Based on environment, applications should log various types of telemetry to ensure stakeholders have the right depth of actionable insights.

# Severity Levels

## Objective:

To outline the various levels of log severity and inform which levels should be incorporated for the proper environments along with what types of data to collect.

## Questions:

* How well do your teams *know* the various severity levels?
  + Why do you feel this way?
* How well do your teams *utilize* the various severity levels?
  + Why do you feel this way?
* How well do your teams capture the appropriate types of data for each level?
* How well do your teams collect the appropriate types of data for each environment (e.g., production, development, etc.)?
* In what areas can your teams improve on leveraging the appropriate severity levels and data capture?
* What are your retention periods for these log levels?
* What are your security policies for various log levels?

## Lead:

A sample application is available to demonstrate various levels of logging. By interacting with the application, we can observe what type of data and respective severity levels is collected.

# Demo – Logging

## Objective:

To demonstrate various types of data being collected by the application and how to interpret severity levels.

## Questions:

*Talk through the demo and explain what you are doing.*

Talking points:

* Logger levels within the code /app/Controllers/ItemController.cs
* Logging scopes for transactions
* Look at the data in the Log Analytics Workspace

**Look at and drill into the AppTraces data (expand “Properties”)**

|  |
| --- |
| AppTraces  | where TimeGenerated >= ago(1h)  | order by TimeGenerated desc |

**Format AppTraces data**

|  |
| --- |
| AppTraces  | where TimeGenerated >= ago(1h)  | project TimeGenerated, CategoryName = Properties.CategoryName, Message, SeverityLevel,  HttpVerb = Properties.HttpVerb,  Action = Properties.Action,  TransactionId = Properties.TransactionId,  Id = case(Properties.Id != "", Properties.id, parse\_json(tostring(Properties.item)).id),  Name = parse\_json(tostring(Properties.item)).name,  Description = parse\_json(tostring(Properties.item)).description,  IsComplete = parse\_json(tostring(Properties.item)).isComplete,  Category = case(Properties.category != "", Properties.category, parse\_json(tostring(Properties.item)).category)  | order by TimeGenerated desc |

## Lead:

There is a vast amount of data that can be collected and there is some that shouldn’t. Customers should be aware of best practices for data collection along with some of the dangers.

# Logging Best Practices

## Objective:

To communicate best practices for collecting telemetry along with some of the pitfalls of collecting too much.

## Questions:

* How many of these best practices do your teams follow?
  + Which ones?
  + Which ones do they not follow so well?
    - Why do you believe this is the case?
* What other policies do you follow for logging?
* How do you trace events across multiple application layers or microservices?

## Lead:

A demo is available in which a substantial load is placed on a basic, load balanced App Service and Cosmos DB. The dashboards will give some insights in to how the application and the underlying services respond.

# Demo – Grafana

## Objective:

To demonstrate an application under stress and identify areas for improving resiliency and performance.

## Questions:

*Talk through the demo and explain what you are doing.*

Talking points:

* Describe what’s happening (e.g., virtual users are being added every minute)
* View the increased stress on Azure resources
* Increases in CPU, memory (secondary App Service sometimes will scale to meet demand)
* Increases in RU consumption on Cosmos DB
* Notice what happens with Cosmos DB reaches 100% consumption – App Services don’t fail, however, the application fails
  + This is an indicator that the database RUs should be increased to handle load
* Zoom in and show how long Cosmos DB/App was unresponsive

## Lead:

Customers will have need for assistance. Work with the customer to identify needs and next steps.

Some next steps could include:

* Enabling Log Analytics for all production services
* Undergoing a Proactive Reliability or Performance review
* A work-with engagement in which CSAs work with the customer to identify current data collected, gaps where data isn’t collected, and security concerns.
* Improving operational reliability practices (e.g., adding alerting, building RACI charts, defining SLAs, constructing dashboards, etc.)

**Remember:** Build solutions; don’t sell products.