4. Monitor and troubleshoot Azure solutions

└ 4.1 Monitor and troubleshoot solutions by using Application Insights

└ 4.1.3 Instrument an app or service to use Application Insights

1. What are the main ways to instrument an application with Application Insights?
2. How do you install and configure the Application Insights SDK?
3. How does automatic vs. manual instrumentation differ?
4. How do you log custom events and metrics?
5. How do you enable Application Insights in Azure App Service or Azure Functions?
6. What is TelemetryClient and how is it used?
7. How do you set context (e.g., user ID, session ID) in telemetry?
8. How do you integrate App Insights with non-.NET apps (e.g., Node.js, Java)?
9. What is Application Insights Sampling and why is it used?
10. How do you verify instrumentation is working?

---

1. What are the main ways to instrument an application with Application Insights?

- **SDK-based**: Add App Insights SDK to the code (e.g., ASP.NET, Node.js)
- **Agent-based**: Enable from Azure Portal (App Service, Functions)
- **Connection string**: Set `APPLICATIONINSIGHTS_CONNECTION_STRING` in app settings

---

2. How do you install and configure the Application Insights SDK?

For ASP.NET Core:

Install `Microsoft.ApplicationInsights.AspNetCore` via NuGet, then add to `Program.cs` or `Startup.cs`:

```
builder.Services.AddApplicationInsightsTelemetry("<connection-string>");
```

---

3. How does automatic vs. manual instrumentation differ?

- **Automatic**: Captures requests, exceptions, dependencies without code changes
- **Manual**: Use `TelemetryClient` to track custom events, metrics, or exceptions explicitly in code

---

4. How do you log custom events and metrics?

Use `TelemetryClient.TrackEvent("eventName")` or `TrackMetric("metricName", value)`. You can add custom properties using dictionary parameters for additional context.

---

5. How do you enable Application Insights in Azure App Service or Azure Functions?

Go to the resource in the Azure Portal, select "Application Insights", enable monitoring, and link or create an instance. The instrumentation key or connection string is auto-injected.

---

6. What is TelemetryClient and how is it used?

`TelemetryClient` is the core class for sending custom telemetry. Instantiate it via DI or manually, then call methods like `TrackTrace`, `TrackException`, or `TrackEvent`.

---

7. How do you set context (e.g., user ID, session ID) in telemetry?

Use `TelemetryClient.Context` to set properties like `User.Id`, `Session.Id`, or `Operation.Id`. This enables correlating logs by user or session.

8. How do you integrate App Insights with non-.NET apps (e.g., Node.js, Java)?
Install the respective SDK (e.g., `applicationinsights` for Node.js, `applicationinsights-agent` for Java), then initialize with the connection string and enable auto-collection.

---

9. What is Application Insights Sampling and why is it used?
Sampling reduces telemetry volume by sending a subset of data. It helps control cost and data ingestion while preserving statistical accuracy. Enabled via `AddApplicationInsightsTelemetry` or configuration.

---

10. How do you verify instrumentation is working?
Use the Live Metrics Stream and “Logs” tab in the App Insights blade. You should see traces, requests, and dependencies appear within seconds of activity.