4. Monitor and troubleshoot Azure solutions

└ 4.1 Monitor and troubleshoot solutions by using Application Insights

└ 4.1.1 Monitor and analyze metrics, logs, and traces

1. What is Application Insights and what core telemetry does it collect?
2. What is the difference between metrics, logs, and traces in App Insights?
3. How do you instrument code to send telemetry to Application Insights?
4. How do you configure Application Insights in an Azure App Service or Function?
5. How do you analyze telemetry data using the Azure portal?
6. What is Kusto Query Language (KQL) and how is it used with App Insights?
7. How do you filter and visualize logs using Log Analytics?
8. How can you track request dependencies and failures?
9. How do you correlate telemetry across distributed services?
10. What are performance counters and how are they monitored?

---

1. What is Application Insights and what core telemetry does it collect?

Application Insights is an APM tool in Azure Monitor that automatically collects telemetry like requests, exceptions, dependencies, traces, custom events, and performance metrics from applications.

---

2. What is the difference between metrics, logs, and traces in App Insights?

- Metrics: Numeric time-series data (e.g., CPU, request count)
- Logs: Structured records from telemetry (e.g., exceptions, requests)
- Traces: Developer-written debug/log messages for tracing app behavior

---

3. How do you instrument code to send telemetry to Application Insights?

- Use SDKs like `Microsoft.ApplicationInsights.AspNetCore`
- Initialize `TelemetryClient` to send custom events
- Use auto-instrumentation via Application Insights extension in App Services or Azure Functions

---

4. How do you configure Application Insights in an Azure App Service or Function?

Enable App Insights from the Azure Portal by turning on "Application Insights" in the Monitoring section. The connection string / instrumentation key is injected into the app's environment variables.

---

5. How do you analyze telemetry data using the Azure portal?

Use the Application Insights blade to view built-in charts, failures, performance, and live metrics. Use "Logs" to query telemetry with KQL and "Failures" or "Performance" tabs for drill-down diagnostics.

---

6. What is Kusto Query Language (KQL) and how is it used with App Insights?

KQL is a read-only query language for analyzing telemetry in Azure Monitor. You use it in the "Logs" section of Application Insights to query tables like `requests`, `exceptions`, `dependencies`, `traces`.

7. How do you filter and visualize logs using Log Analytics?

Use the "Logs" tab in Application Insights to write KQL queries (e.g., `requests | where duration > 1s`). Use the Chart button to visualize results and pin them to Azure dashboards.

---

8. How can you track request dependencies and failures?

Use the "Application Map" and "Failures" tab in Application Insights. Dependency telemetry captures outbound calls (SQL, HTTP, etc.) with duration and result codes for diagnosing bottlenecks and errors.

9. How do you correlate telemetry across distributed services?
Application Insights uses `operation_Id` and `parent_Id` to link related telemetry. Use the "End-to-End Transaction Details" view or join telemetry tables in KQL using `operation_Id`.

---

10. What are performance counters and how are they monitored?
Performance counters like CPU, memory, and request duration are automatically collected in supported environments. View them under the "Performance" tab or query the `performanceCounters` table via KQL.

4. Monitor and troubleshoot Azure solutions

└ 4.1 Monitor and troubleshoot solutions by using Application Insights

└ 4.1.2 Implement Application Insights web tests and alerts

1. What are Application Insights web tests and their core types?
2. How do you create a URL ping test in Application Insights?
3. What are multi-step web tests and when should you use them?
4. What are availability test locations and why are they important?
5. How are alerts configured for availability tests?
6. How do you use metric-based alerts in Application Insights?
7. What is the difference between classic alerts and new Azure Monitor alerts?
8. How can you configure alerts for failed requests or dependencies?
9. How do action groups work with alerts in Application Insights?
10. What best practices apply to monitoring app availability with web tests?

---

1. What are Application Insights web tests and their core types?

Application Insights web tests are synthetic availability tests. The two core types are:

- URL ping tests: Periodically hit an endpoint to verify availability.
- Multi-step tests: Scripted tests that simulate user journeys (deprecated for new use).

---

2. How do you create a URL ping test in Application Insights?

Go to the “Availability” tab in App Insights, choose “Add test”, select “URL ping test”, configure the test name, URL, locations, frequency, success criteria, and alert settings.

---

3. What are multi-step web tests and when should you use them?

Multi-step tests use .webtest files uploaded to App Insights to simulate complex scenarios. They're useful for emulating real user workflows but are deprecated and replaced by Azure Load Testing or Playwright tests in GitHub Actions.

---

4. What are availability test locations and why are they important?

Availability tests run from multiple Azure datacenters (e.g., US West, Europe North). This validates global uptime and helps detect region-specific failures or latency issues.

---

5. How are alerts configured for availability tests?

When creating a web test, you can enable alerts on failure. Alternatively, use Azure Monitor to create an alert rule that triggers when the availabilityResults signal detects failed tests.

---

6. How do you use metric-based alerts in Application Insights?

Navigate to Azure Monitor > Alerts > New Alert Rule. Choose an App Insights resource, then select metrics like requests/failed, availabilityResults/availabilityPercentage, define condition, and assign an action group.

---

7. What is the difference between classic alerts and new Azure Monitor alerts?

Classic alerts are legacy and limited in flexibility. Azure Monitor alerts support advanced logic, dynamic thresholds, metric-based conditions, and integration with action groups.

8. How can you configure alerts for failed requests or dependencies?
Create an alert rule in Azure Monitor using Application Insights as the resource. Select signals like requests/failed or dependencies/failed, define a threshold (e.g., count > 5 in 5 mins), and link to an action group.

---

9. How do action groups work with alerts in Application Insights?
Action groups define how alerts notify users or systems. You can trigger emails, SMS, webhooks, Azure Functions, or Logic Apps. They are reusable across multiple alert rules.

---

10. What best practices apply to monitoring app availability with web tests?

- Test from multiple regions
- Set up alerts for sustained failures, not transient ones
- Use short intervals for mission-critical endpoints
- Use secure URLs (HTTPS) and validate content in the response

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.