Get the best out of Live Sessions

HOW?

Check your Internet Connection

Log in 10 mins before, and check your internet connection to avoid any network issues during the LIVE session.

Speak with the Instructor

By default, you will be on mute to avoid any background noise. However, if required you will be **unmuted by instructor**.

Clear Your Doubts

Feel free to clear your doubts. Use the "**Questions**" tab on your webinar tool to interact with the instructor at any point during the class.

Let us know if you liked our content

Please share feedback after each class. It will help us to enhance your learning experience.

edureka!

Microsoft Azure Developer Associate (AZ-204)

COURSE OUTLINE
MODULE 10

Introduction to Azure IaaS Compute Solutions

Implementing Azure Batch Service and Disk Encryption

Designing and Developing Applications That Use Containers

Implementing Azure App Service Web Apps and Mobile Apps

Implementing Azure App Service API Apps and Azure Functions

Developing Solutions That Use Azure Table Storage and Cosmos DB

Developing Solutions That Use Relational Database and Azure Blob Storage

Implementing Authentication and Access Control in Azure

Implementing Secure Data Solutions and Integrate Caching & CDN

Instrument Monitoring, Logging and Scalability of Apps & Services

Connecting to and Consuming Azure and Third-party Services

Developing Event-based and Message-based Solutions in Azure

Module 10 – Instrument Monitoring, Logging and Scalability of Apps & Services

edureka!

Topics

- Cloud Monitoring
- Azure Monitor
- Alerts and Metrics
- Activity Log
- Service Health
- Application Insights
- Autoscaling In Azure
- Autoscale – Best Practices
- Common Autoscale Patterns
- Handling Transient Faults
- Transient Fault Handling – General Guidelines

Objectives

After completing this module, you should be able to:

- Understand how Azure Monitor works
- Configure instrumentation in an app or server by using Application Insights
- Analyze and troubleshoot solutions by using Azure Monitor
- Understand Auto-scale patterns and best practices for scaling their solutions
- Handle transient faults in your solution

Monitoring Azure Applications and Resources

edureka!

Cloud Monitoring

- Monitoring is the act of **collecting** and **analyzing** data to determine the performance, health, and availability of your business application and the resources that it ***depends on***

- An **effective** monitoring strategy helps you understand the *detailed* operation of the components of your application

- It also helps you increase your **uptime** by *proactively* notifying you of critical issues so that you can resolve them before they become problems

edureka!

Azure Monitor

Azure Monitor

Azure includes *multiple* services that **individually** perform a specific *role* or *task* in the monitoring space

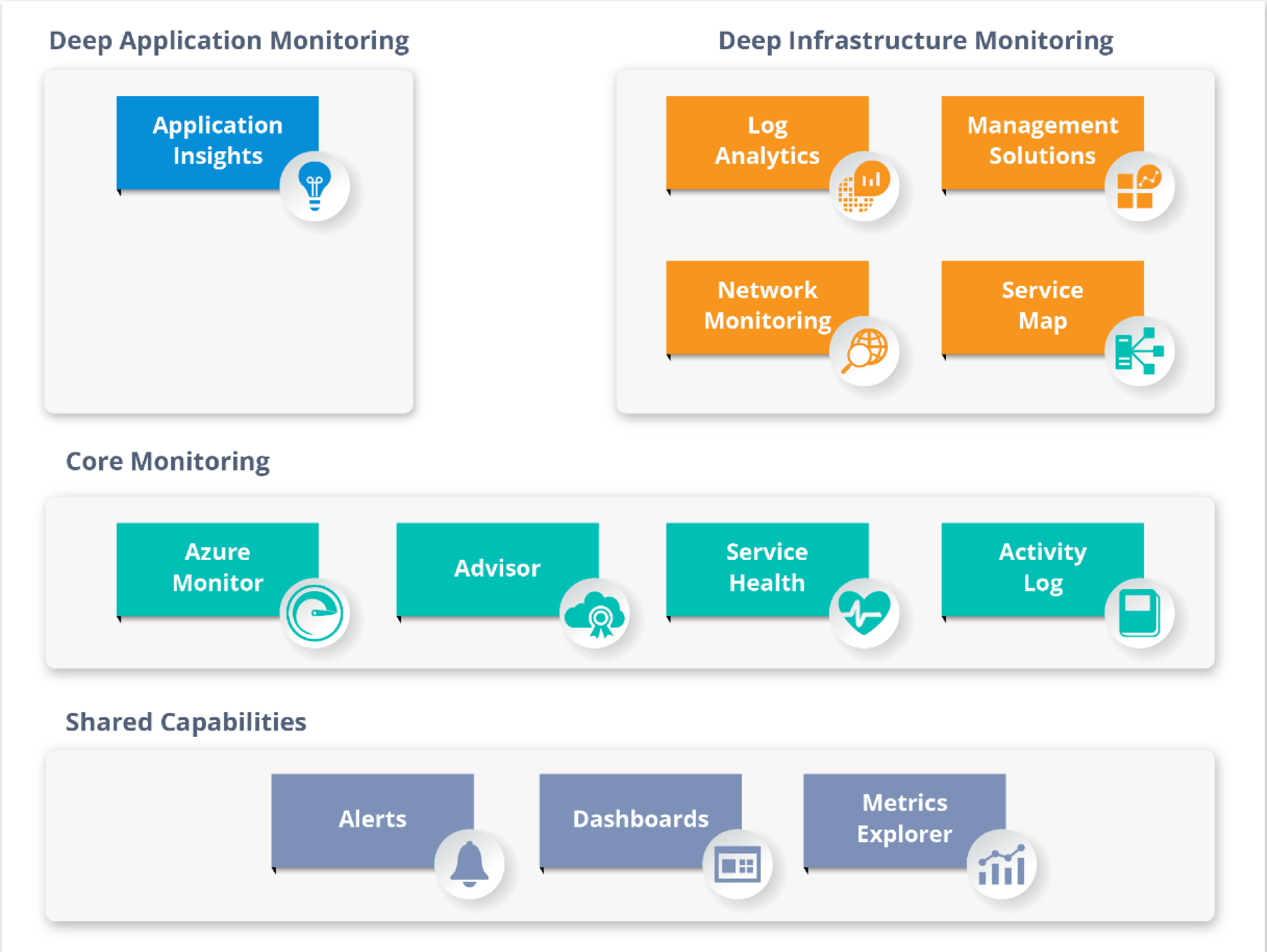
Together, these services deliver a comprehensive solution for collecting, analysing, and acting on telemetry from your application and the Azure resources that support them

They can also work to monitor *critical* **on-premises** resources in order to provide a ***hybrid*** monitoring environment

Understanding the **tools** and **data** that are available is the ***first step*** in developing a complete monitoring strategy for your application

Conceptual View of Azure Monitoring

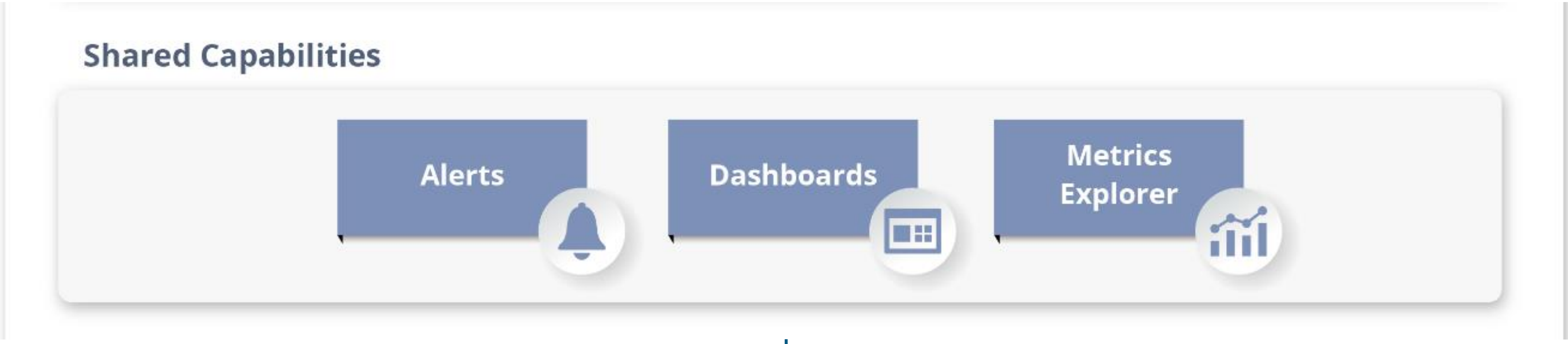
Conceptual view of the components that work together to provide monitoring of Azure resources:



edureka!

Azure Monitor – Shared Capabilities

The core and deep monitoring service share functionality which provides the following capabilities:



Azure alerts proactively *notify* you of critical conditions and potentially take corrective action

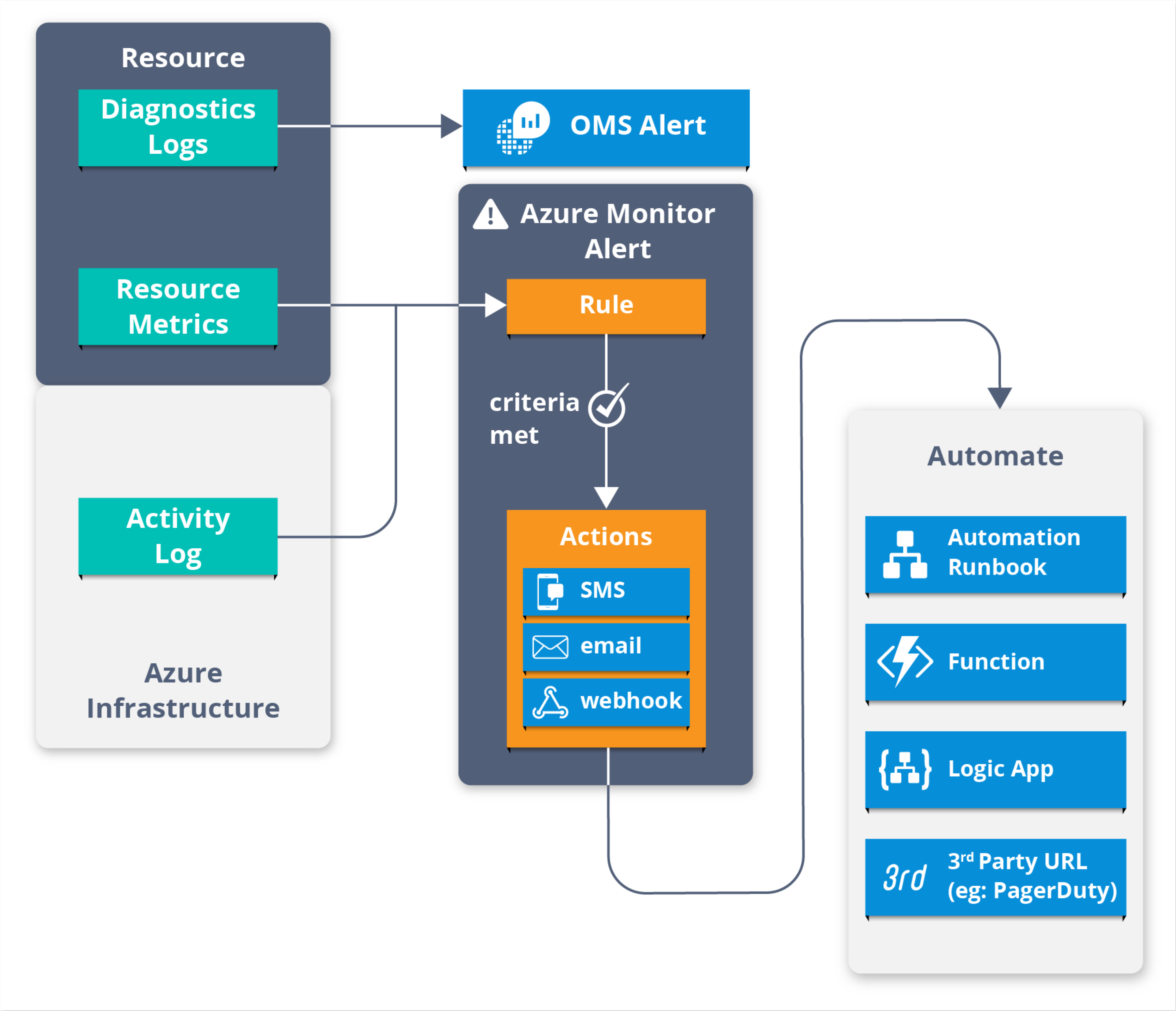
You can use **Azure Dashboards** to combine different kinds of data into a *single pane* in the Azure portal

Metrics are numerical values generated by an Azure resource to help you understand the *operation* and *performance* of the resource

edureka!

Shared Capabilities – Alerts

- **Alert rules** can use data from *multiple* sources, including **metrics** and **logs**
- They use **action groups**, which contain *unique* sets of **recipients** and **actions** in *response* to an alert
- Based on your *requirements*, you can have alerts start *external actions* by using **webhooks** and *integrate* with your **ITSM** (IT Service Management) tools



edureka!

Shared Capabilities – Alerts Scenario

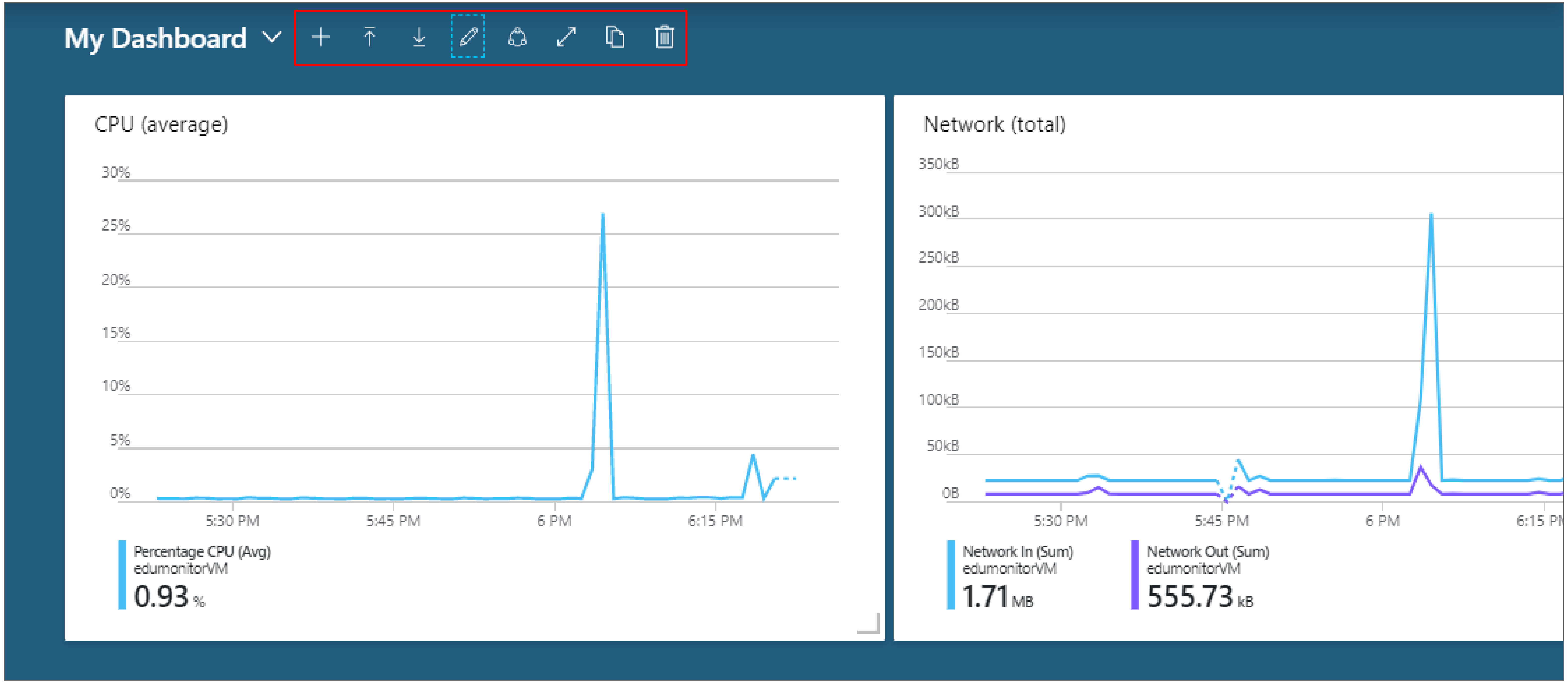
- Consider, You have a Virtual Machine or a Storage Service running in your Resource Group reaches it’s **upper limit**, you might want it to upgrade to it’s **higher Configuration** or **Tier**
- In this case, you have to create an **Alert Rule** in that resource to perform the above **Operation** for you
- In the next demo, you will learn how to create an Alert Rule based on **Metric**

Azure Monitor – Shared Capabilities – Dashboards

- You can use **Azure dashboards** to *combine* different kinds of data into a single pane in the Azure portal
- You can then **share** the dashboard with other Azure users
- For example, you can create a dashboard that combines:
  - Tiles that show a graph of metrics
  - A table of activity logs
  - A usage chart from Application Insights
  - The output of a log search in Log Analytics

Azure Portal – Dashboard

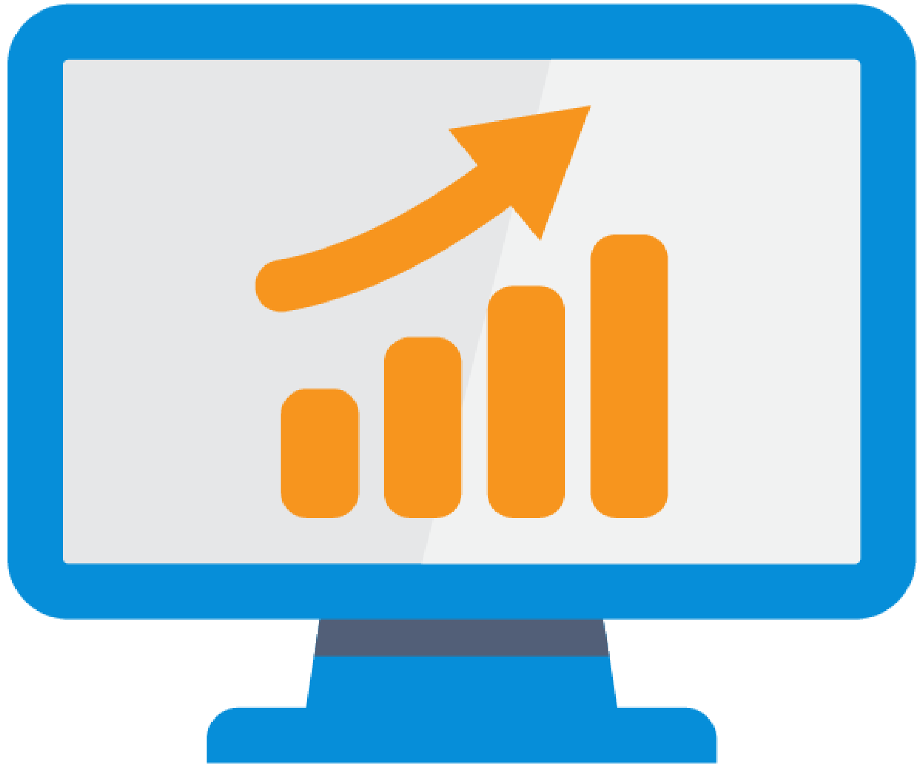
You can **Add a New Dashboard**, **Upload/Download** Dashboard in **JSON** format, **Customize**, **Share** your Dashboard with other Users, **Clone** and **Delete** Dashboards:



edureka!

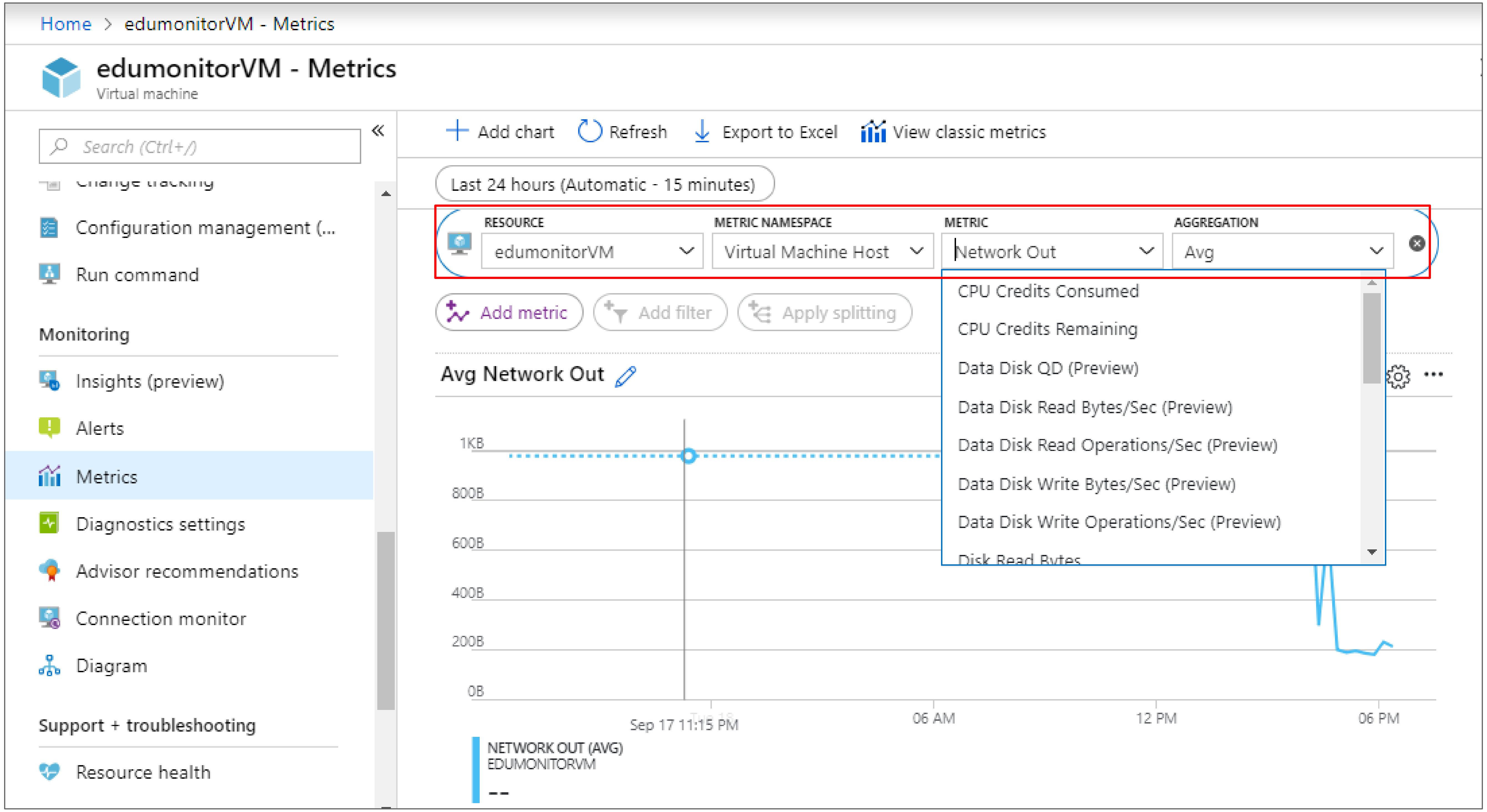
Azure Monitor – Shared Capabilities – Metrics

- **Metrics** are numerical values generated by an Azure resource to help you understand the ***operation*** and ***performance*** of the resource
- By using Metrics Explorer, you can send metrics to **Log Analytics** for analysis with data from other sources:



Metrics of a Specific Resource

To check the metrics of your resource, Goto: <<Your VM>> > Click on **Metrics** > Select any Metric:



RESOURCE: edumonitorVM

METRIC NAMESPACE: Virtual Machine Host

METRIC: Network Out

AGGREGATION: Avg

Last 24 hours (Automatic - 15 minutes)

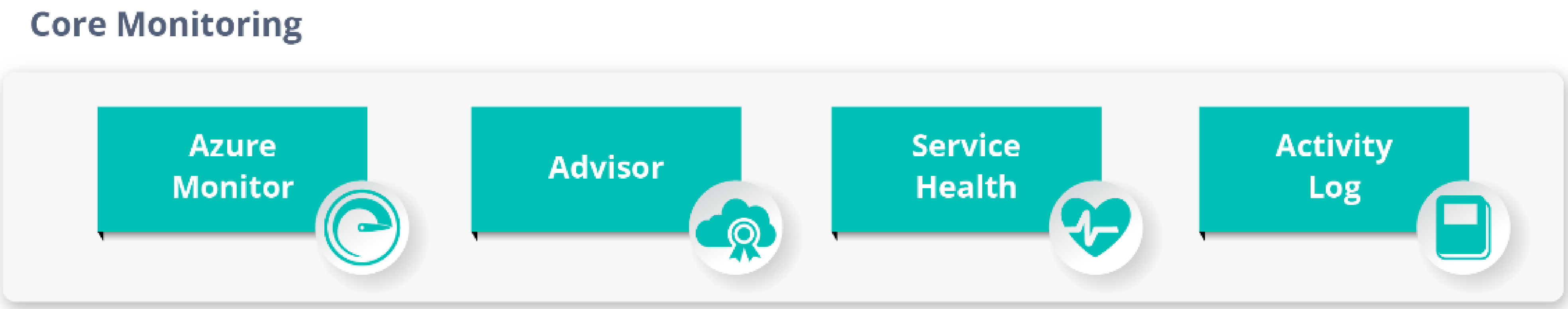
edureka!

Core Monitoring – Azure Monitor

- **Azure Monitor** enables core monitoring for Azure services by allowing the collection of **metrics**, **activity logs**, and **diagnostic logs**
- For example, the activity log tells you when new resources are *created* or *modified*
- You can also send these metrics and logs to **Azure Log Analytics** for trending and detailed analysis

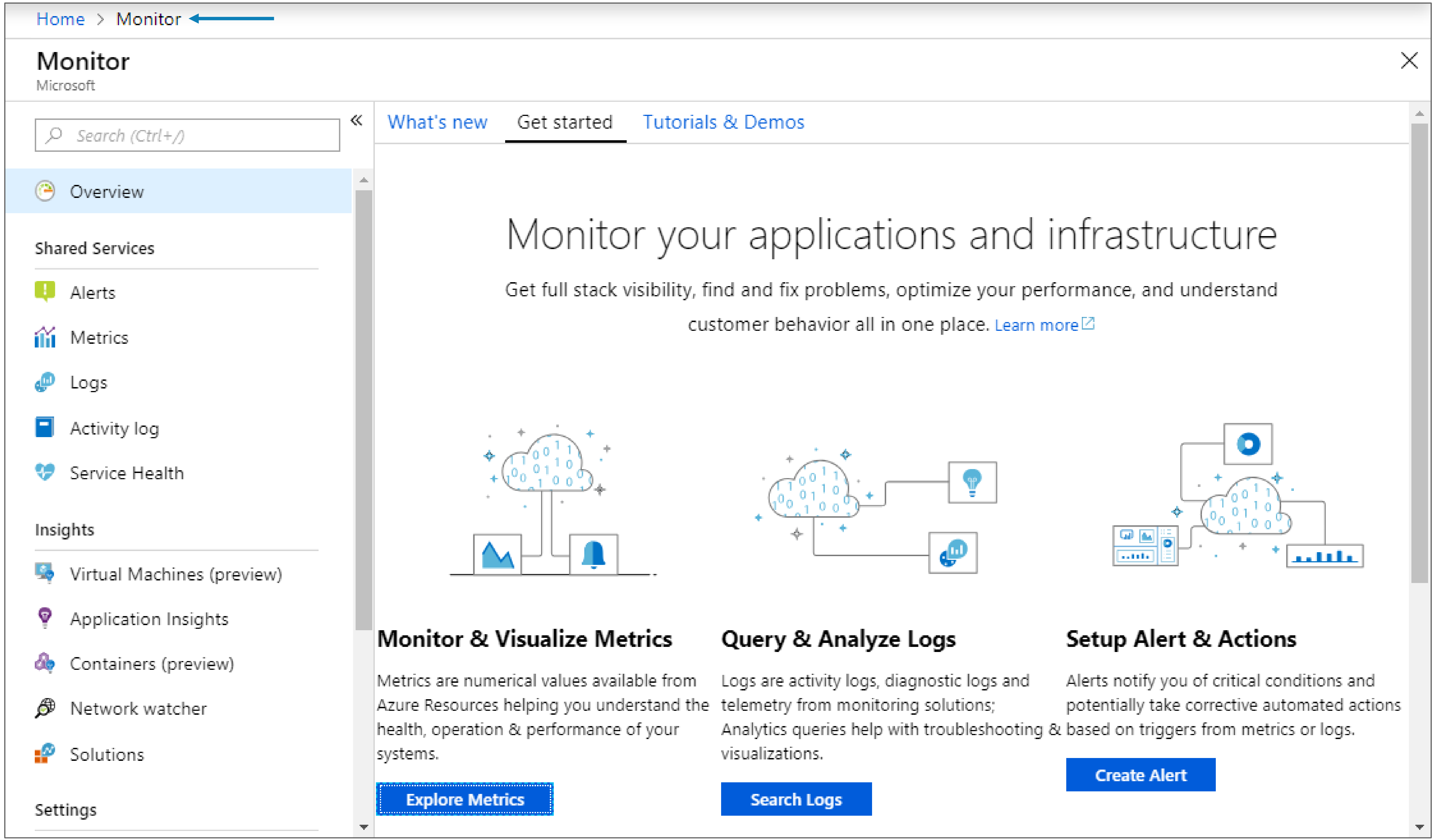
OR

Create additional **alert rules** to proactively notify you of critical issues as a result of that analysis



Azure Monitor – Portal

Find **Azure Monitor** Service on the Main Menu > Click on it:



Home > Monitor

Monitor
Microsoft

Monitor your applications and infrastructure

Get full stack visibility, find and fix problems, optimize your performance, and understand customer behavior all in one place. Learn more

Monitor & Visualize Metrics

Metrics are numerical values available from Azure Resources helping you understand the health, operation & performance of your systems.

Query & Analyze Logs

Logs are activity logs, diagnostic logs and telemetry from monitoring solutions; Analytics queries help with troubleshooting & visualizations.

Setup Alert & Actions

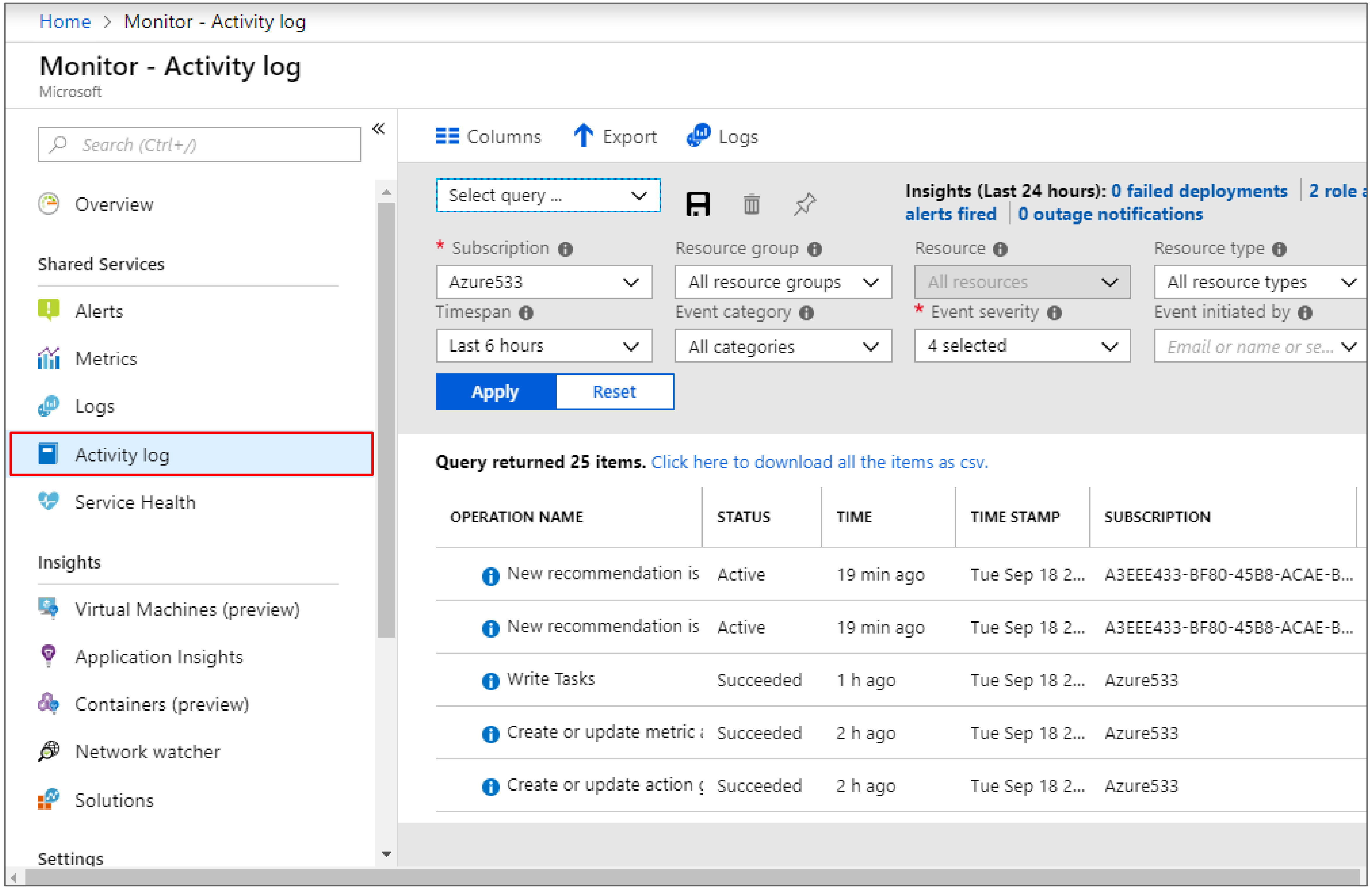
Alerts notify you of critical conditions and potentially take corrective automated actions based on triggers from metrics or logs.

Azure Monitor – Activity Log

- **Activity Log** provides data about the operation of an Azure resource, this information includes:
  - Configuration changes to the resource
  - Service health incidents
  - Recommendations on better utilizing the resource
  - Information related to autoscale operations
- You can also send activity log entries to Log Analytics

Azure Monitor – Activity Log Explorer

Select **Activity Log** in the Azure Monitor window, You can see the below queries and results:



Monitor - Activity log

Microsoft

Select query ...

Insights (Last 24 hours): 0 failed deployments | 2 role a alerts fired | 0 outage notifications

* Subscription: Azure533	Resource group: All resource groups	Resource: All resources	Resource type: All resource types
Timespan: Last 6 hours	Event category: All categories	* Event severity: 4 selected	Event initiated by: Email or name or se...

Query returned 25 items. Click here to download all the items as csv.

OPERATION NAME	STATUS	TIME	TIME STAMP	SUBSCRIPTION
New recommendation is	Active	19 min ago	Tue Sep 18 2...	A3EEE433-BF80-45B8-ACAE-B...
New recommendation is	Active	19 min ago	Tue Sep 18 2...	A3EEE433-BF80-45B8-ACAE-B...
Write Tasks	Succeeded	1 h ago	Tue Sep 18 2...	Azure533
Create or update metric a	Succeeded	2 h ago	Tue Sep 18 2...	Azure533
Create or update action g	Succeeded	2 h ago	Tue Sep 18 2...	Azure533

edureka!

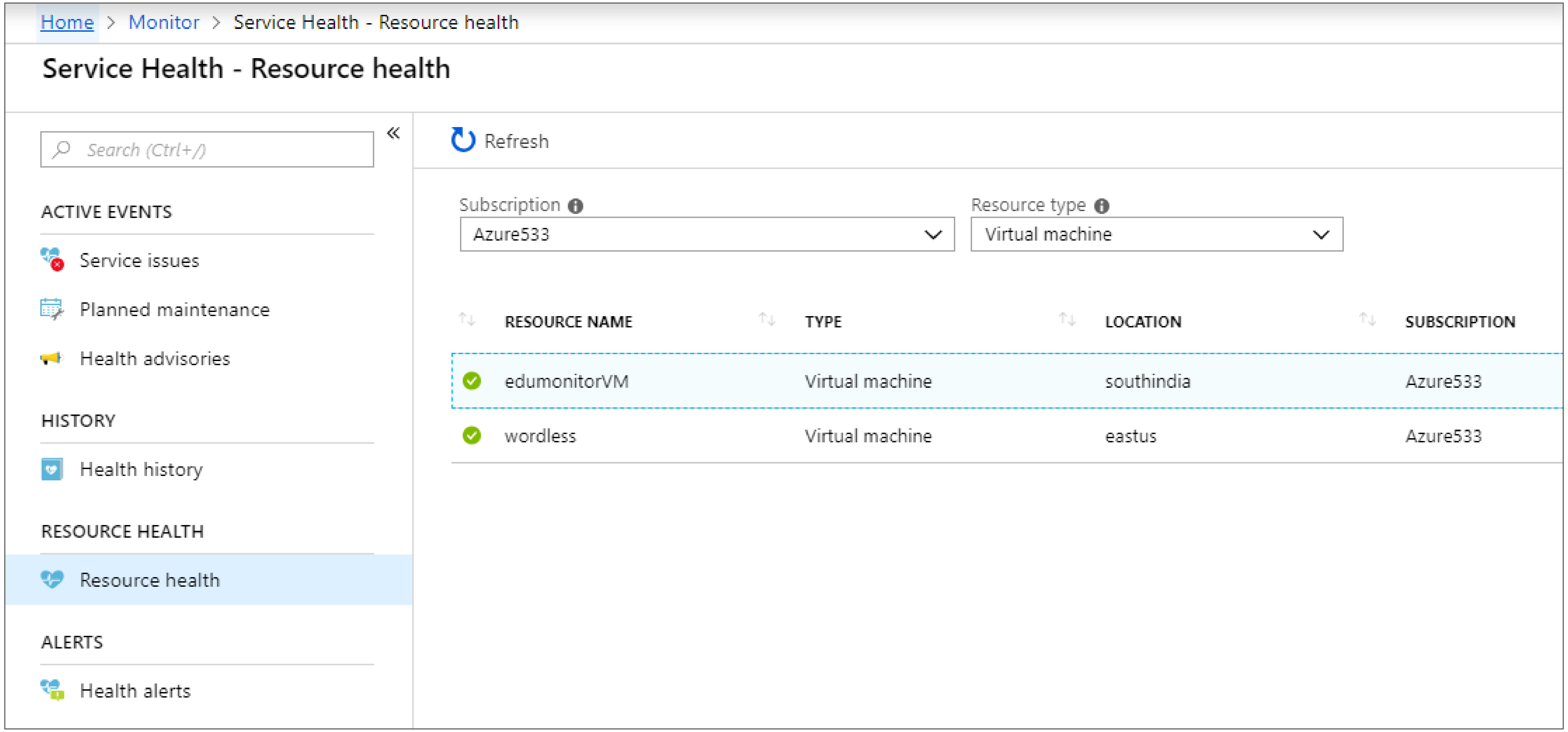
Azure Monitor – Service Health

- The health of your application relies on the **Azure services** that it ***depends on***
- **Azure Service Health** identifies any *issues* with Azure services that might *affect* your application
- Service Health also helps you *plan* for **scheduled maintenance**

edureka!

Azure Monitor – Service Health Explorer

In Service Health, You can check for any Health issues or Planned Maintenance > Resource Health:



Home > Monitor > Service Health - Resource health

Service Health - Resource health

Subscription: Azure533

Resource type: Virtual machine

RESOURCE NAME	TYPE	LOCATION	SUBSCRIPTION
✓ edumonitorVM	Virtual machine	southindia	Azure533
✓ wordless	Virtual machine	eastus	Azure533

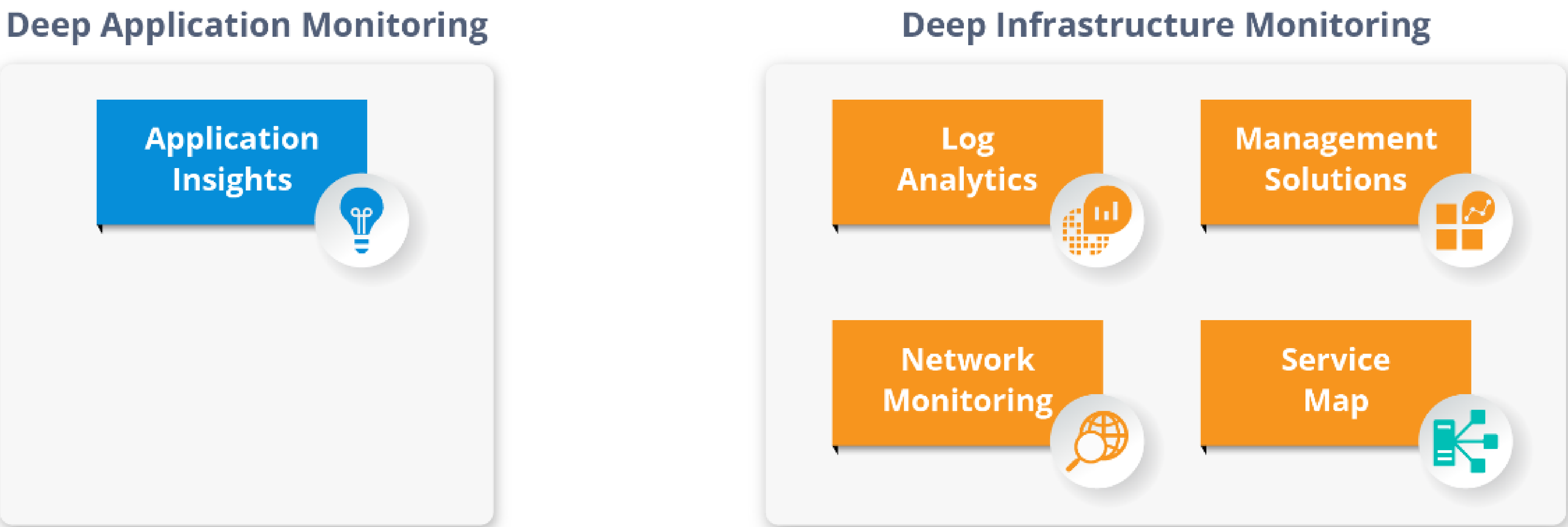
edureka!

Demo 1 – Analyze and Troubleshoot Solutions Using Azure Monitor

edureka!

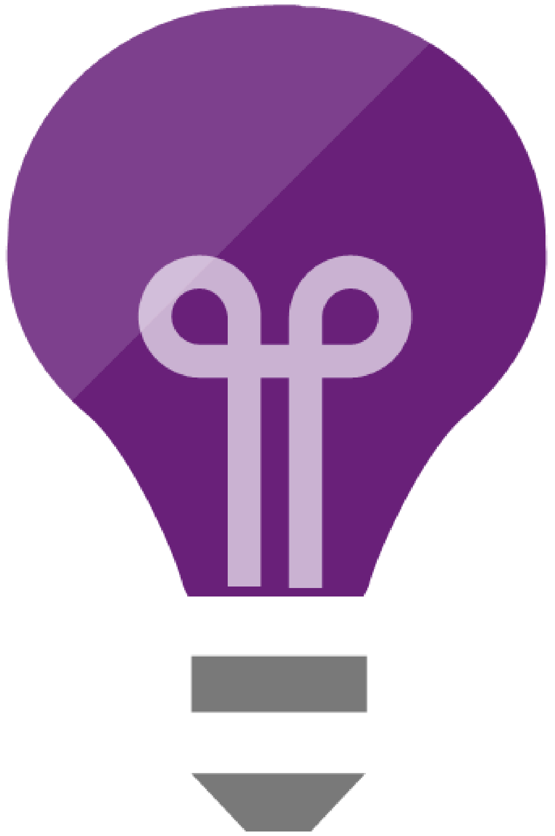
Deep Monitoring Services

- **Deep Monitoring Services** provide rich capabilities for collecting and analyzing monitoring data at a deeper level
- These services build on core monitoring and take advantage of common functionality in Azure
- They provide powerful analytics with collected data to give you unique insights into your applications and infrastructure
- They present data in the context of scenarios that are targeted to different audiences



Deep Application Monitoring – Application Insights

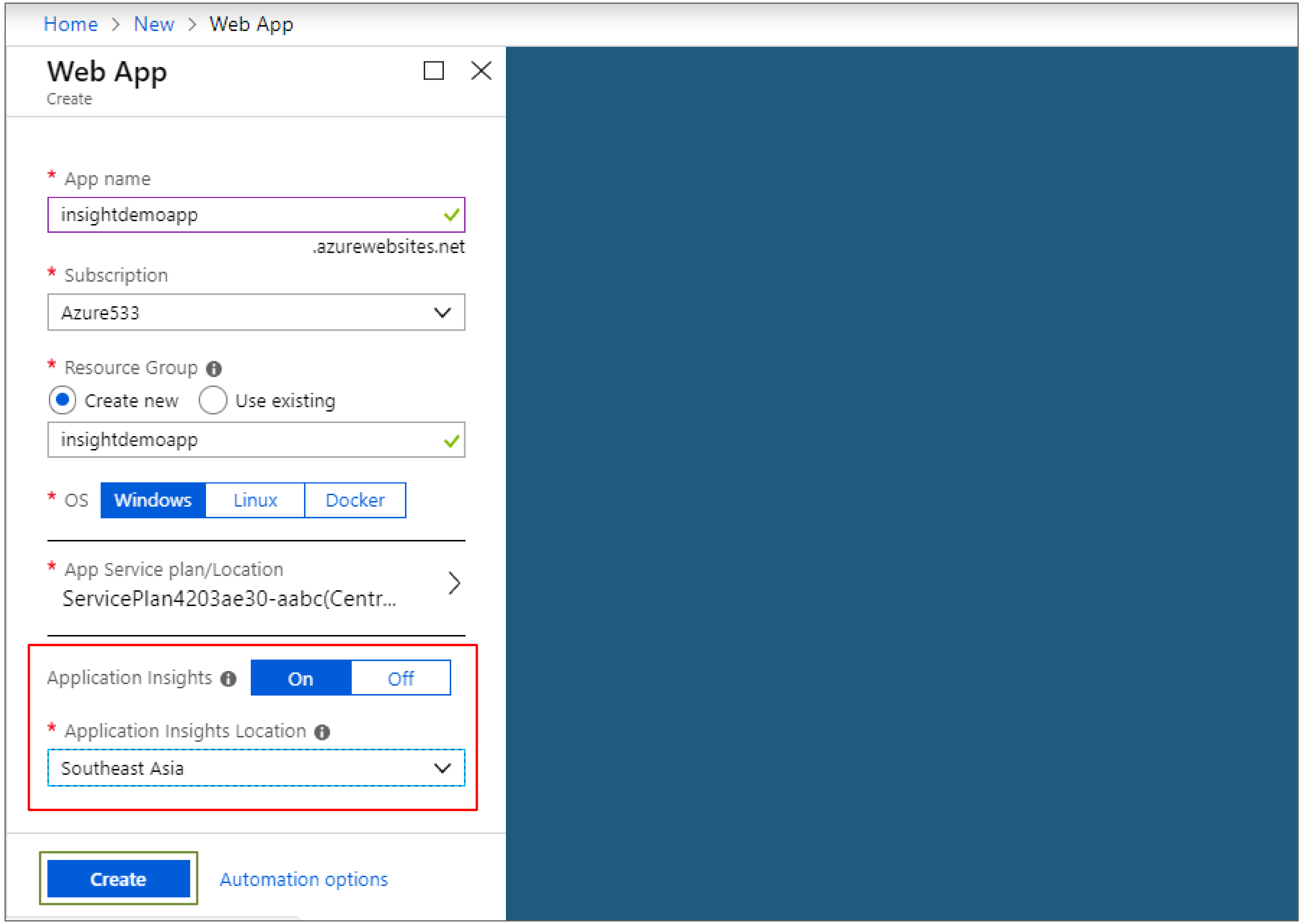
- You can use Azure Application Insights to monitor availability, performance, and usage of your application, whether it's hosted in the cloud or on-premises
- By instrumenting your application to work with Application Insights, you can achieve deep insights and implement **DevOps** scenarios
- You can **quickly** *identify* and *diagnose* **errors** without waiting for a user to report them
- Application Insights has extensive tools for interacting with the data that it collects



Application Insights on Web App – Azure Portal

Enable Application Insights during

WebApp creation:



Home > New > Web App

Web App
Create

* App name
insightdemoapp ✓
.azurewebsites.net

* Subscription
Azure533

* Resource Group
● Create new ○ Use existing
insightdemoapp ✓

* OS [✓ Windows] Linux Docker

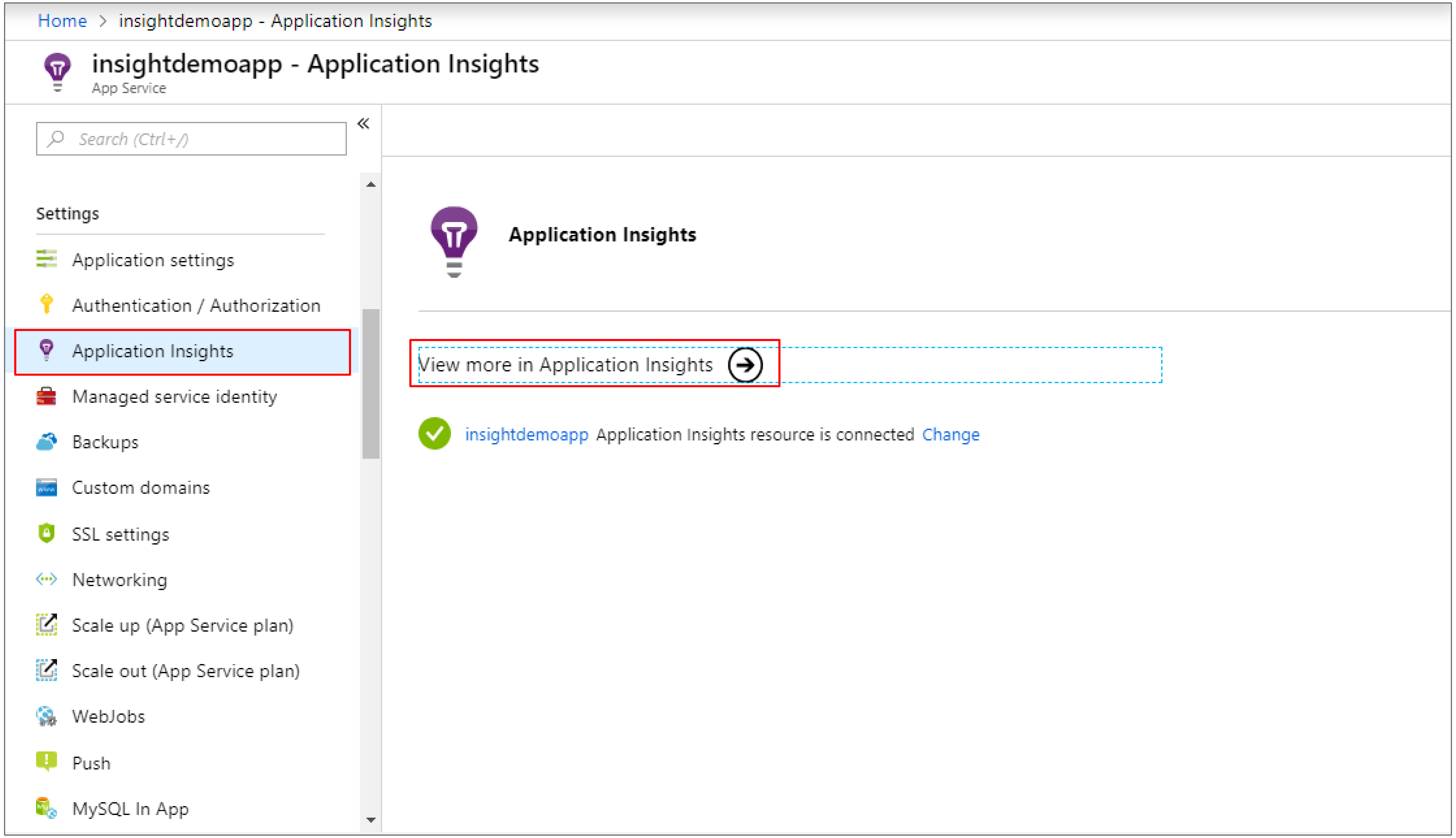
* App Service plan/Location
ServicePlan4203ae30-aabc(Centr...

Application Insights [✓ On] Off

* Application Insights Location
Southeast Asia

edureka!

Click on Application Insights in the App Settings



Home > insightdemoapp - Application Insights

insightdemoapp - Application Insights

App Service

Application Insights

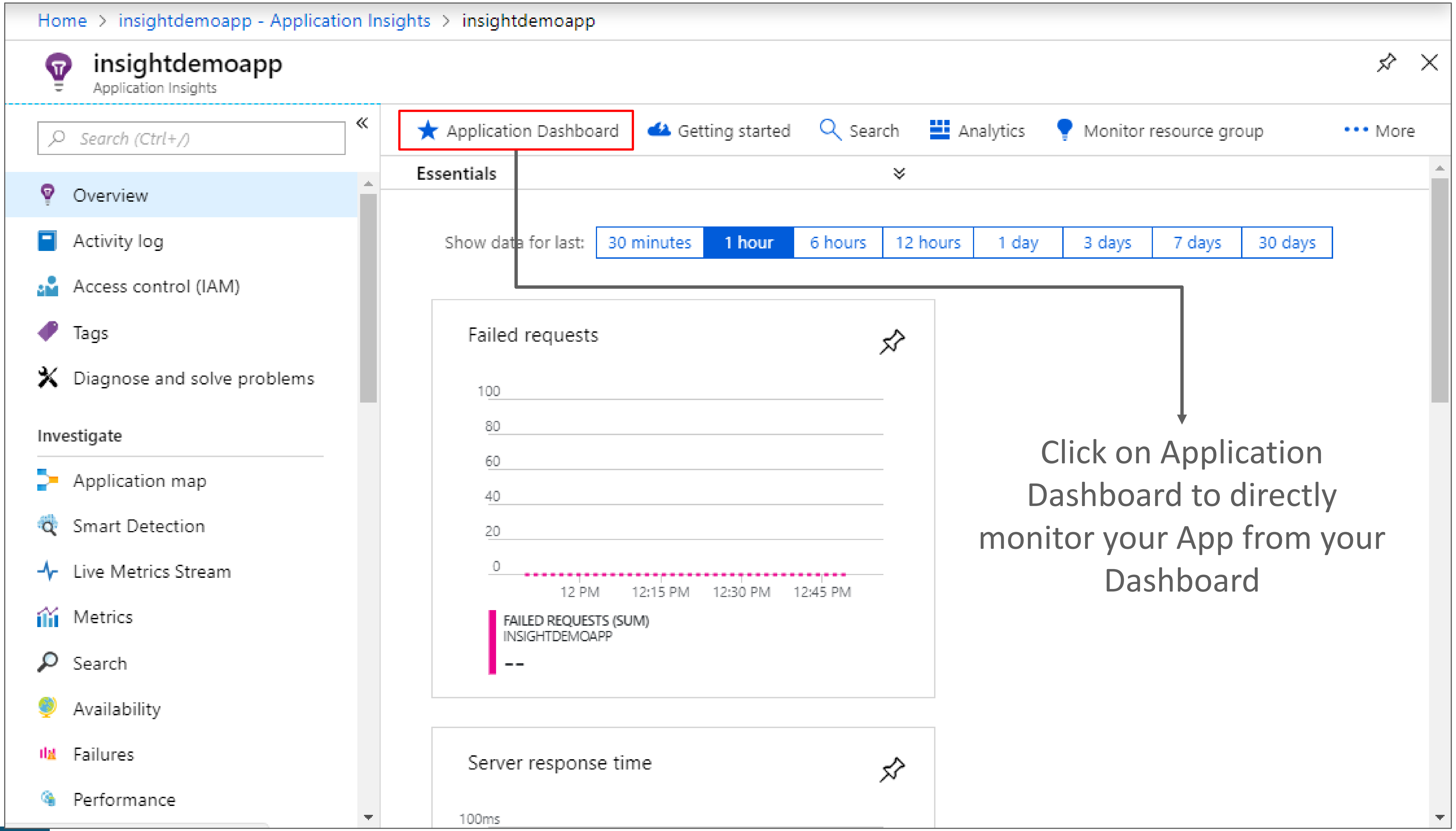
View more in Application Insights

insightdemoapp Application Insights resource is connected Change

edureka!

Application Insights – Overview

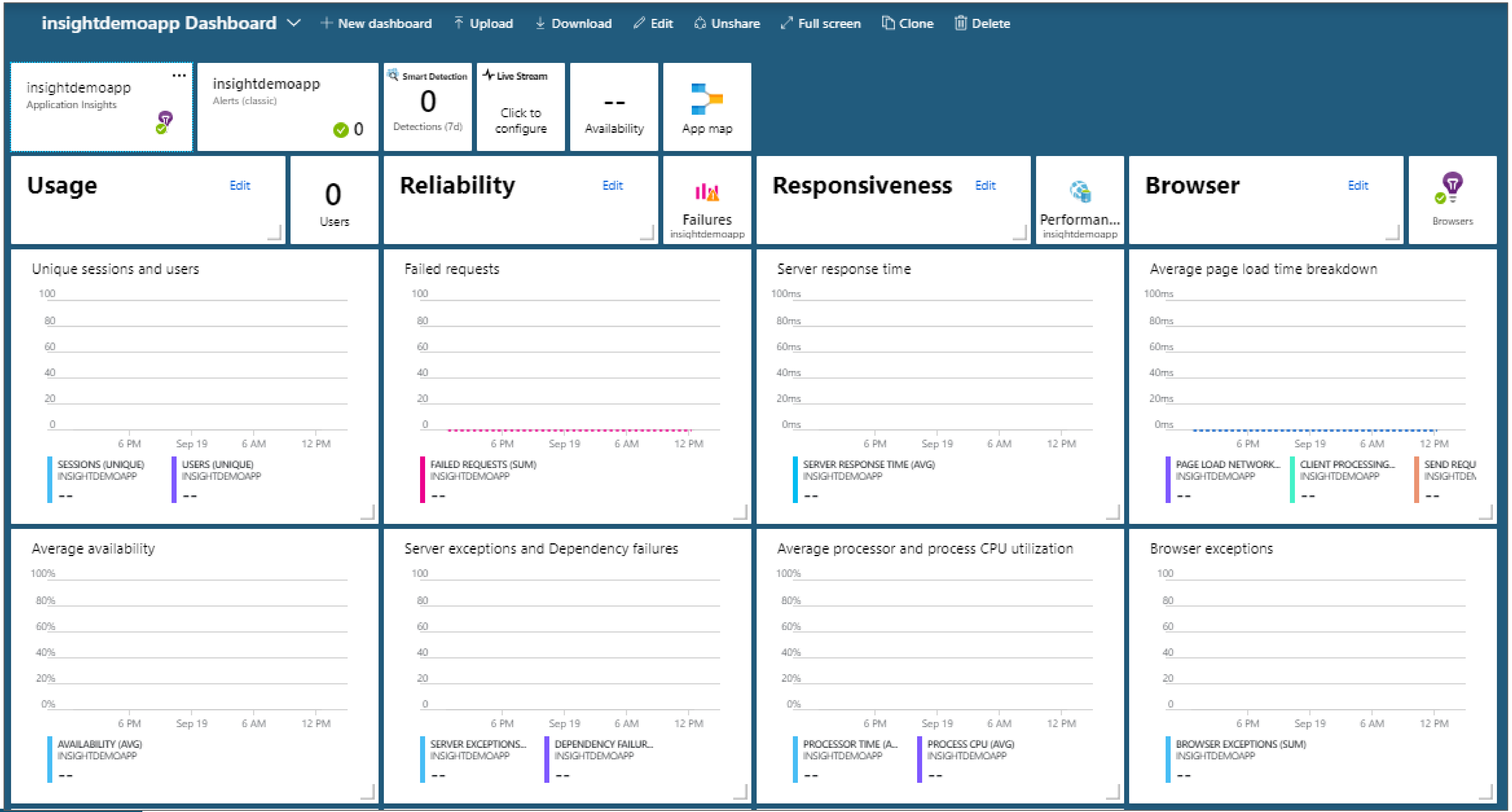
You can perform a lot of Monitoring operations on your WebApp as shown below:



Click on Application Dashboard to directly monitor your App from your Dashboard

edureka!

Application Insights Dashboard



edureka!

Demo 2 – Configure Instrumentation in an App Using Application Insights

edureka!

Autoscaling in Azure

edureka!

Autoscaling in Azure – Overview

Autoscaling is the process of ***dynamically*** allocating resources to match performance requirements

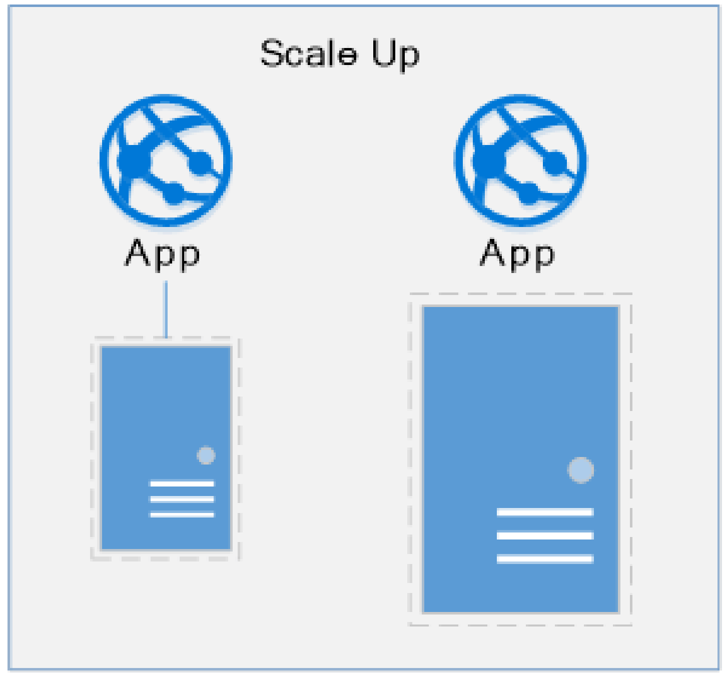
As the volume of work grows, an application may need additional resources to maintain the desired performance levels and satisfy SLAs

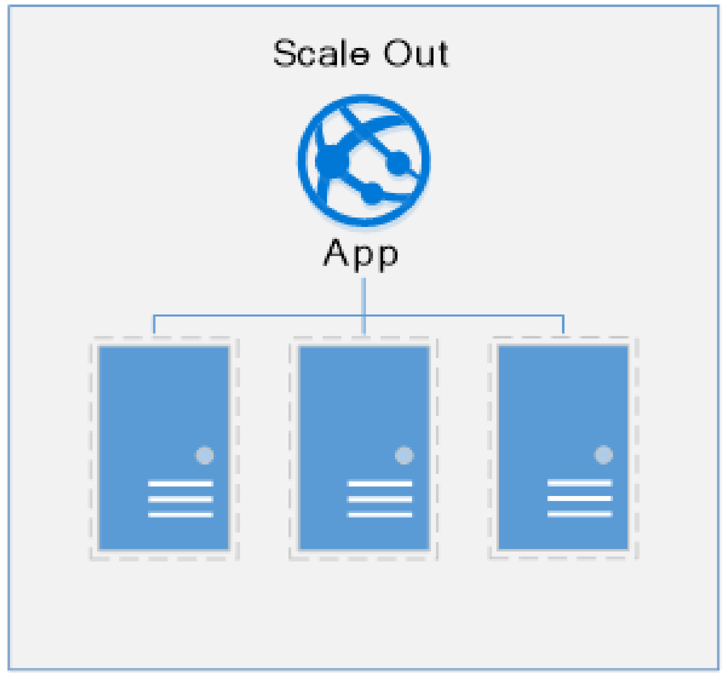
As demand slackens and the additional resources are no longer needed, they can be de-allocated to minimize costs

edureka!

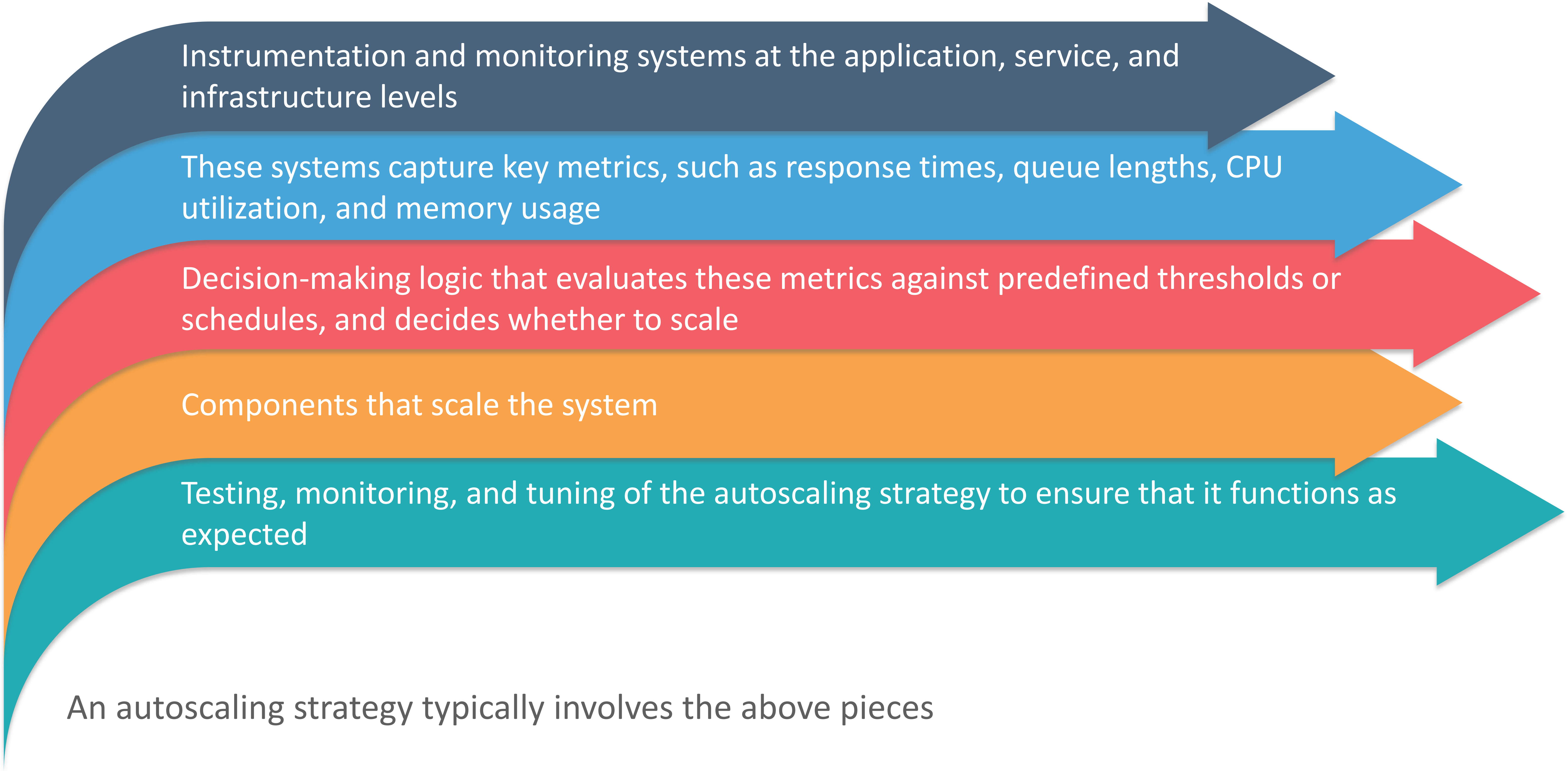
Vertical Scaling Vs. Horizontal Scaling

Vertical Scaling (Scale Up & Down)	Horizontal Scaling (Scale In & Out)
❑ Scale the number and power of resources up and down	❑ Scaling out and in, means adding or removing instances of a resource
❑ For example, you could move an application to a larger VM size	❑ Application continues running ***without*** interruption as new resources are provisioned
❑ Vertical scaling often requires making the system temporarily ***unavailable*** while it is being redeployed	❑ When the provisioning process is complete, the solution is deployed on these additional resources
❑ Less preferred	❑ If demand drops, the additional resources can be shut down cleanly and deallocated





Autoscaling Strategy



Instrumentation and monitoring systems at the application, service, and infrastructure levels

These systems capture key metrics, such as response times, queue lengths, CPU utilization, and memory usage

Decision-making logic that evaluates these metrics against predefined thresholds or schedules, and decides whether to scale

Components that scale the system

Testing, monitoring, and tuning of the autoscaling strategy to ensure that it functions as expected

An autoscaling strategy typically involves the above pieces

Configure Autoscaling for an Azure Solution

- **Azure Virtual Machines** autoscale via Virtual Machine Scale Sets (VMSS), which manage a set of VMs as a group
- **Service Fabric** also supports autoscaling through VMSS (Each Node in a Cluster is setup as a separate VMSS)
- **Azure App Service** Autoscale settings apply to all of the apps within an App Service (built-in autoscaling)
- **Azure Cloud Services** has built-in autoscaling at the role level

These compute options all use **Azure Monitor autoscale** to provide a common set of autoscaling functionality

- **Azure Functions** differs from the previous compute options, because there's no need to configure any autoscale rules
  - Instead, Azure Functions automatically allocates compute power when your code is running, scaling out as necessary to handle load

edureka!

Autoscale – Best Practices

01 Ensure the maximum and minimum values are different and have an adequate margin between them

02 Manual scaling is reset by autoscale min and max

03 Always use a scale-out and scale-in rule combination that performs an increase and decrease

04 Choose the appropriate statistic for your diagnostics metric

05 Choose the thresholds carefully for all metric types

06 Always select a safe default instance count

07 Configure autoscale notifications

Azure Monitor Autoscaling – Web App Metrics

- You can generate a list of the Web Apps metrics by using the following command in PowerShell:

```
Get-AzMetricDefinition -ResourceId <resource_id> | Format-Table -Property Name,Unit
```

- You can generate a list of the Web Apps metrics by using the following command in PowerShell:

Metric Name	Unit
CpuPercentage	Percent
MemoryPercentage	Percent
DiskQueueLength	Count
HttpQueueLength	Count
BytesReceived	Bytes
BytesSent	Bytes

Common Autoscale Patterns

- Scale based on CPU
- Scale differently on weekdays vs weekends
- Scale differently during holidays
- Scale based on custom metric

Demo 3 – Implement Autoscaling Rules and Patterns

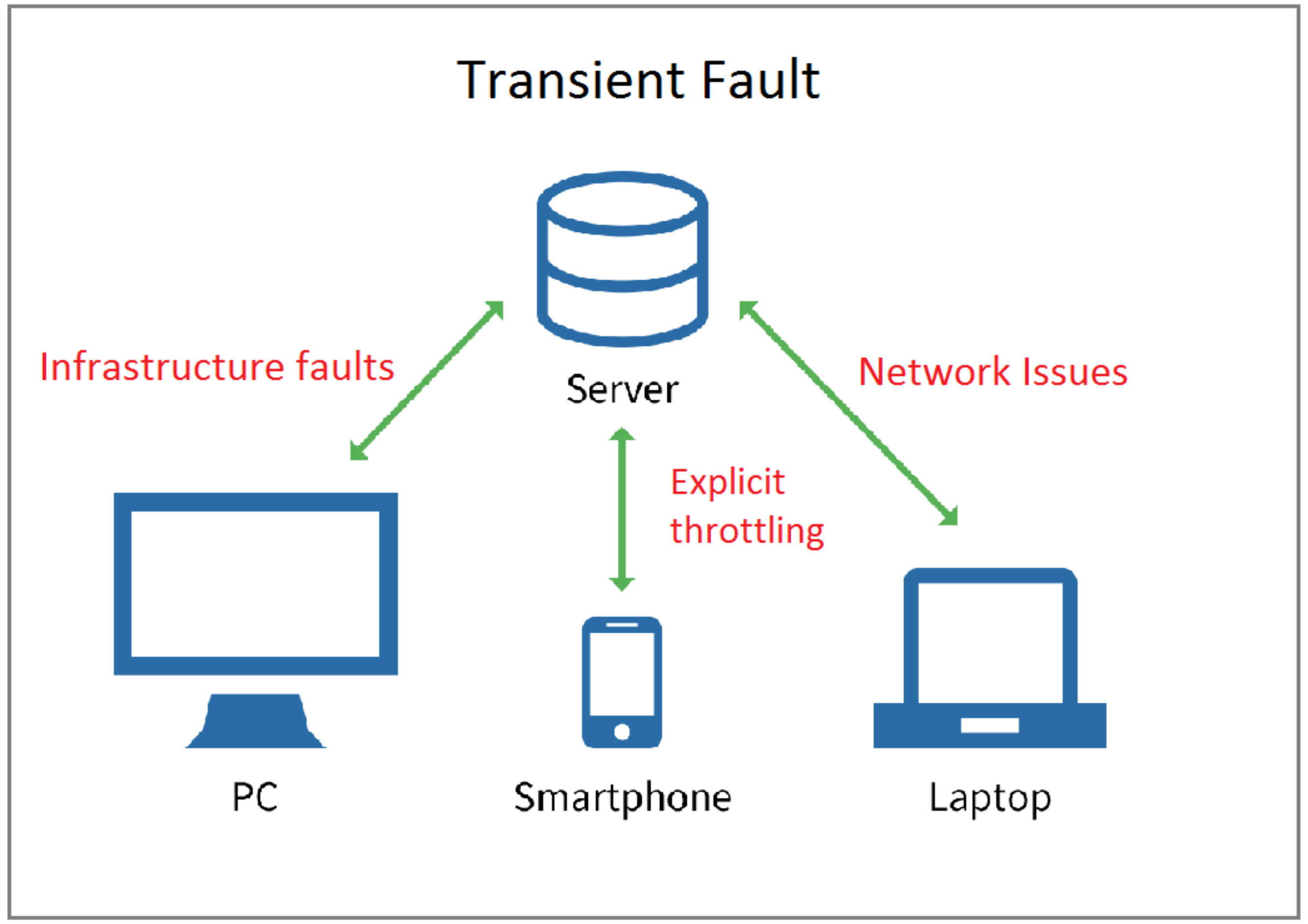
edureka!

Handling Transient Faults

edureka!

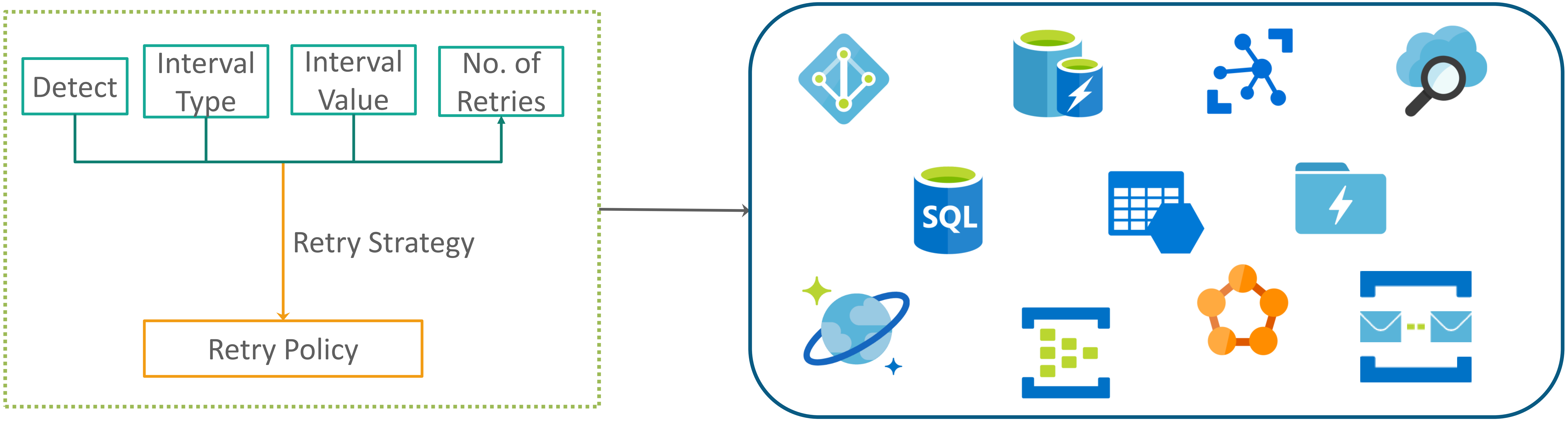
What are Transient Faults?

- When a client makes a request to the server, there may be failure responses because of temporary reasons such as:
  - Network Issues
  - Infrastructure faults
  - Explicit throttling
- These failures are very ***common*** in cloud applications
- **Retrying** the same operation after a short time may result in a successful response
- These errors are called as **Transient Faults**
- These errors occur ***inconsistently*** and no tracking can be done for this error



Transient Fault Handling

- There is no particular way to differentiate transient and non-transient faults
- By ***retrying*** the same server request few more times results in success
- Undergoing the retries based on a ***predefined set of processes*** is known as handing the transient faults
- A **Retry Policy** is a combination of all of the elements of your Retry Strategy



Application

Azure services which include Retry mechanism

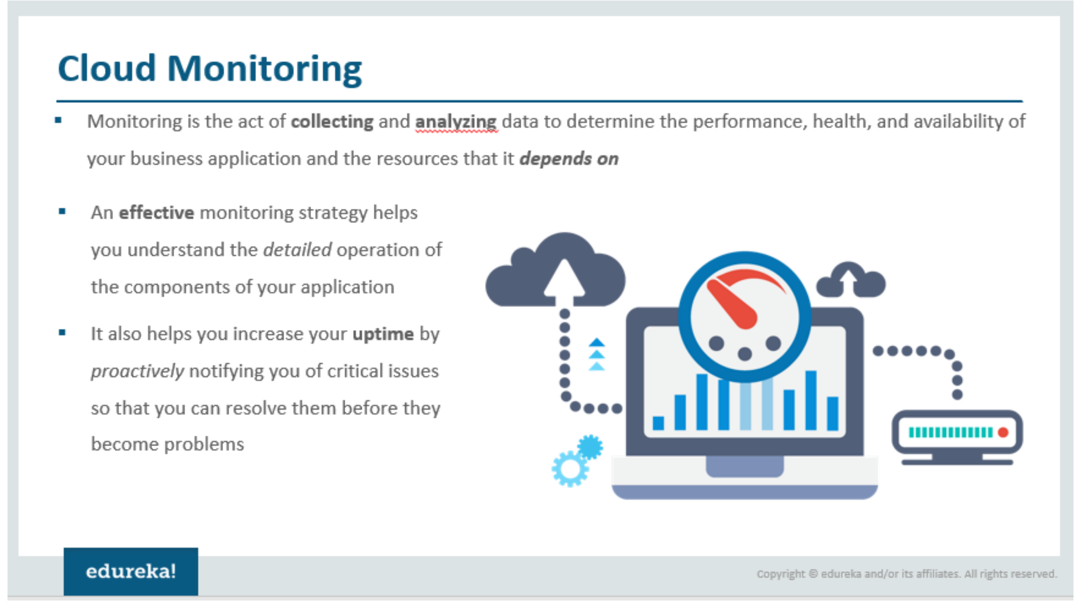
Transient Fault Handling – General Guidelines

- Determine if there is a built-in retry mechanism
- Determine if the operation is suitable for retrying
- Determine an appropriate retry count and interval:
  - Exponential back-off
  - Incremental intervals
  - Regular intervals
  - Immediate retry
  - Randomization
- Avoid anti-patterns
- Test your retry strategy and implementation
- Manage retry policy configurations
- Log and track transient and non-transient faults
- Manage operations that continually fail

Demo 4 – Implement a Code That Handles Transient Faults

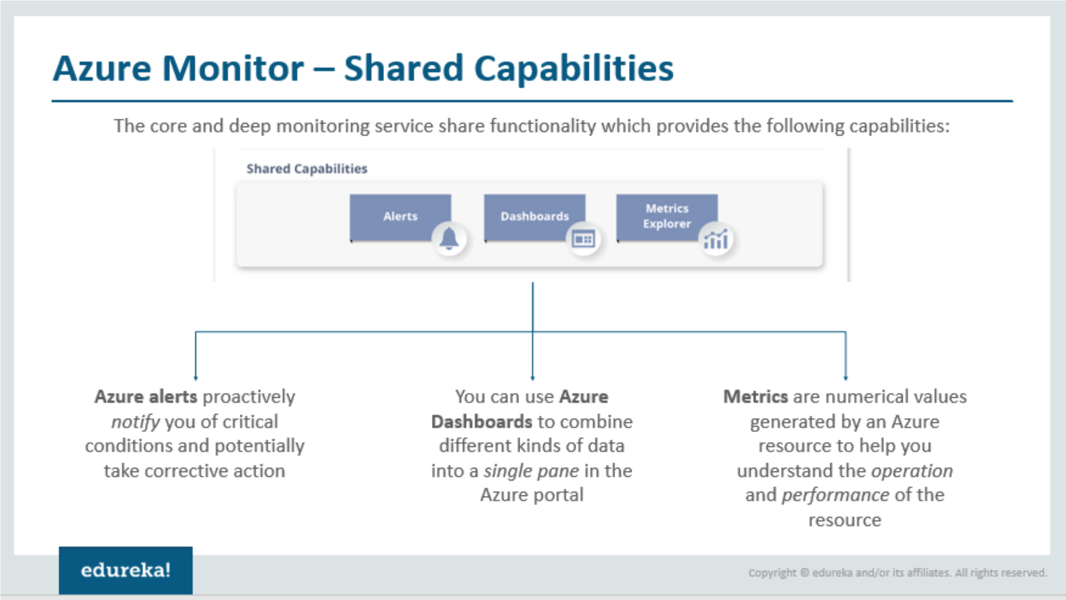
edureka!

Summary



Cloud Monitoring

- Monitoring is the act of **collecting** and **analyzing** data to determine the performance, health, and availability of your business application and the resources that it ***depends on***
- An **effective** monitoring strategy helps you understand the *detailed* operation of the components of your application
- It also helps you increase your **uptime** by *proactively* notifying you of critical issues so that you can resolve them before they become problems



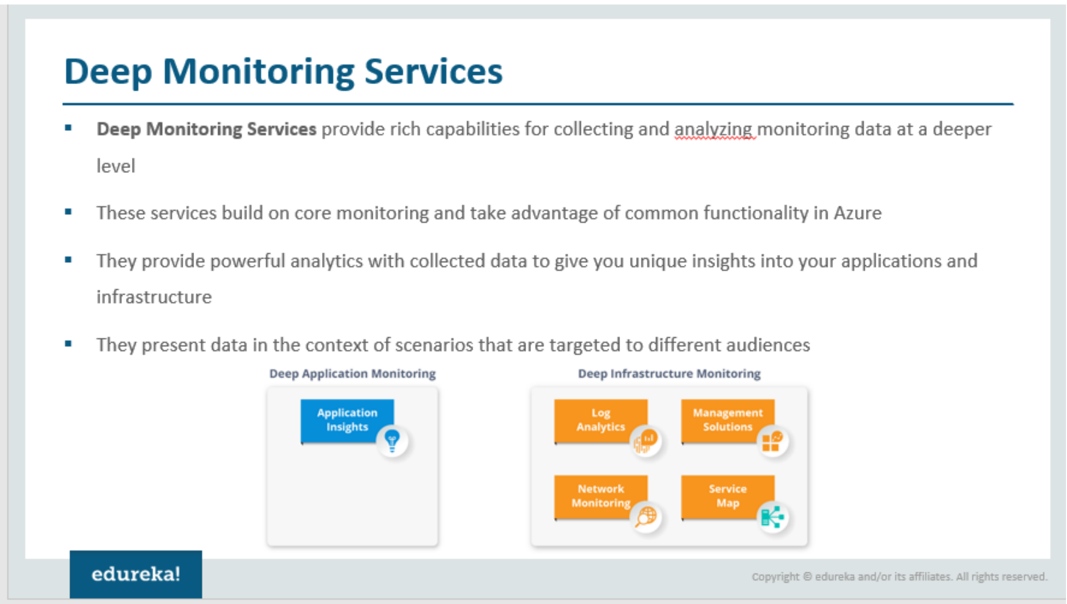
Azure Monitor – Shared Capabilities

The core and deep monitoring service share functionality which provides the following capabilities:

Azure alerts proactively *notify* you of critical conditions and potentially take corrective action

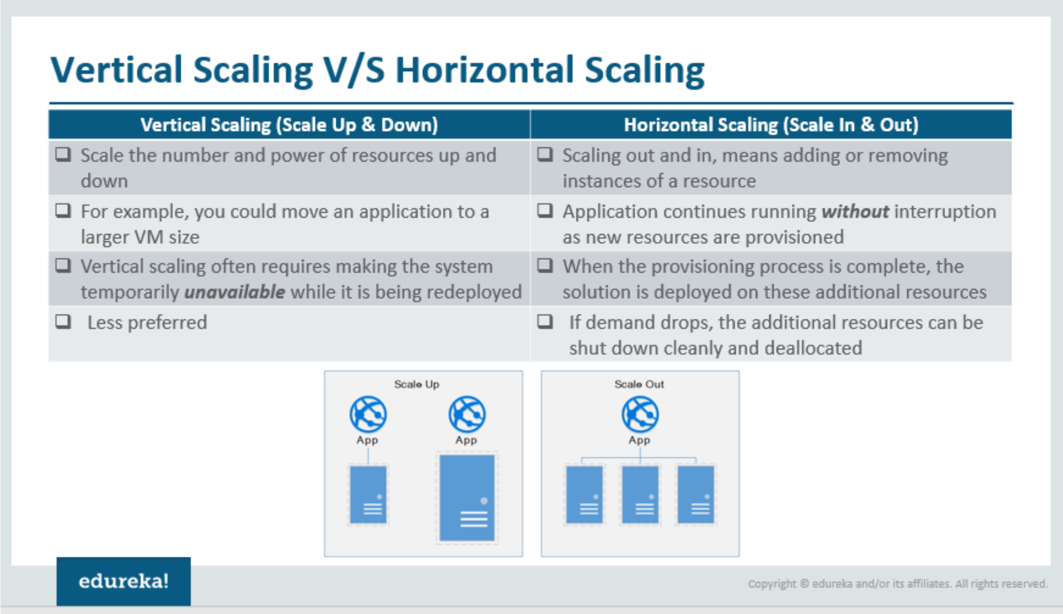
You can use **Azure Dashboards** to combine different kinds of data into a *single pane* in the Azure portal

Metrics are numerical values generated by an Azure resource to help you understand the *operation* and *performance* of the resource



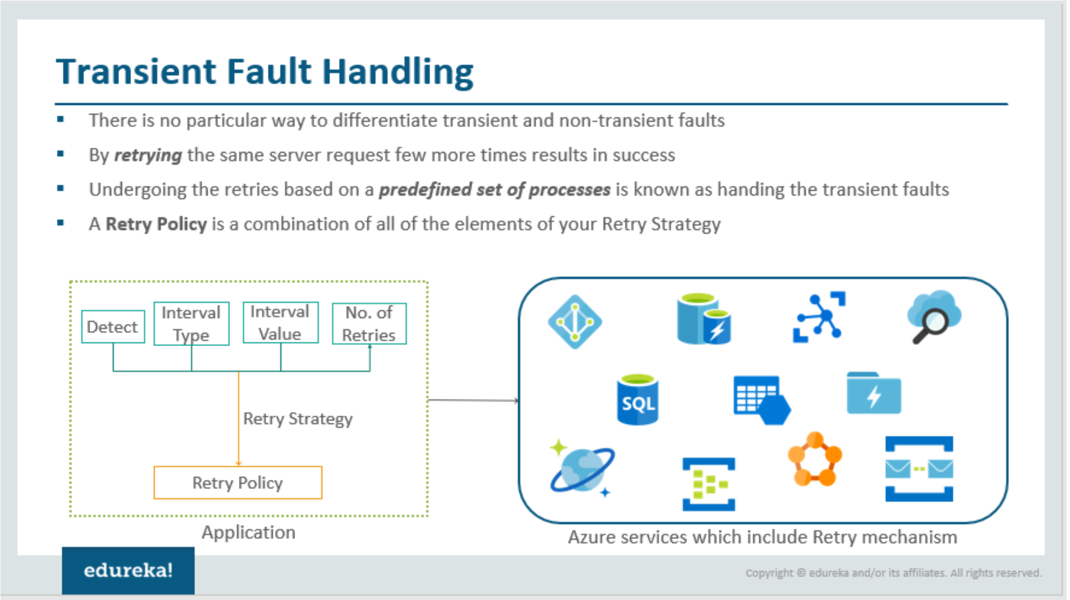
Deep Monitoring Services

- **Deep Monitoring Services** provide rich capabilities for collecting and analyzing monitoring data at a deeper level
- These services build on core monitoring and take advantage of common functionality in Azure
- They provide powerful analytics with collected data to give you unique insights into your applications and infrastructure
- They present data in the context of scenarios that are targeted to different audiences



Vertical Scaling V/S Horizontal Scaling

Vertical Scaling (Scale Up & Down)	Horizontal Scaling (Scale In & Out)
□ Scale the number and power of resources up and down	□ Scaling out and in, means adding or removing instances of a resource
□ For example, you could move an application to a larger VM size	□ Application continues running ***without*** interruption as new resources are provisioned
□ Vertical scaling often requires making the system temporarily ***unavailable*** while it is being redeployed	□ When the provisioning process is complete, the solution is deployed on these additional resources
□ Less preferred	□ If demand drops, the additional resources can be shut down cleanly and deallocated

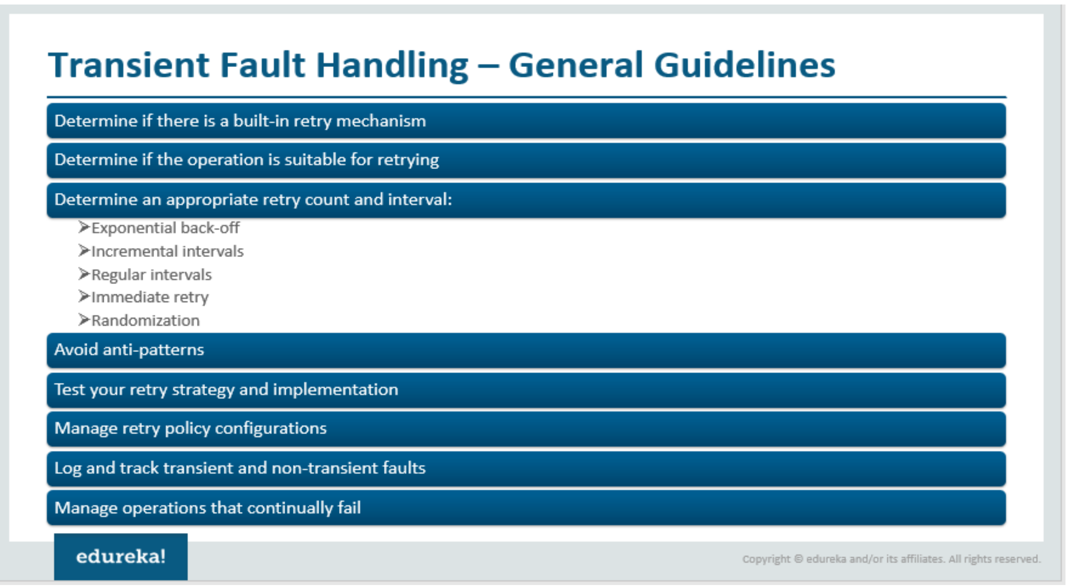


Transient Fault Handling

- There is no particular way to differentiate transient and non-transient faults
- By ***retrying*** the same server request few more times results in success
- Undergoing the retries based on a ***predefined set of processes*** is known as handing the transient faults
- A **Retry Policy** is a combination of all of the elements of your Retry Strategy

Application

Azure services which include Retry mechanism



Transient Fault Handling – General Guidelines

- Determine if there is a built-in retry mechanism
- Determine if the operation is suitable for retrying
- Determine an appropriate retry count and interval:
  - ➢Exponential back-off
  - ➢Incremental intervals
  - ➢Regular intervals
  - ➢Immediate retry
  - ➢Randomization
- Avoid anti-patterns
- Test your retry strategy and implementation
- Manage retry policy configurations
- Log and track transient and non-transient faults
- Manage operations that continually fail

Questions

FEEDBACK

Comments

Ratings

Suggestions

Survey

Likes

Ideas

Thank You

For more information please visit our website
www.edureka.co