2-2-1: Examining Scaling Approaches

After completing this episode, you should be able to:

- Identify and explain the appropriate cloud scaling approach, given a scenario

Description: In this episode, the learner will examine various scaling approaches for cloud resources. We will explore triggered, scheduled, and manual scaling.

- Describe the significance of understanding, identifying, and implementing various scaling approaches
    - The right scaling approach for cloud resources is key to maintaining a reliable, efficient, and cost-effective cloud environment
    - Implementing appropriate scaling methods enhances system resilience, allowing infrastructure to adjust to changing workloads
    - Organizations that focus on scaling can achieve operational efficiency, better user experiences, and greater flexibility to meet evolving business needs.
        - Organizations can ensure their cloud environments are responsive and adapting seamlessly to both expected and unexpected changes in demand
- Describe triggered scaling
    - Triggers
        - Automated scaling based on specific criteria observed within the cloud environment.
    - Examples:
        - Trending - Adjusts resources based on observed trends over time, such as gradual increases in web traffic.
        - Load - Scales resources in real-time based on current load metrics, such as CPU utilization or memory usage, ensuring performance remains stable under varying loads.
        - Event - Reacts to specific events, such as the deployment of new features or the sudden increase in demand due to marketing campaigns.
- Describe scheduled scaling
    - Resources are adjusted based on known or anticipated patterns, such as scaling up during business hours or scaling down overnight.
    - Useful for predictable load variations
    - Can help optimize costs by aligning resource usage with demand cycles
- Describe manual scaling
    - Involves direct intervention to scale resources up or down
    - This method offers precise control
    - Requires manual effort and constant monitoring.
    - Less efficient for dynamic environments
- Role of Automation in Scaling
    - Enhances all these scaling approaches by enabling faster and more accurate adjustments to resource levels
    - Reduces the need for manual intervention
    - Allows systems to respond immediately to changes in demand or predefined schedules
    - Automated scaling can improve operational efficiency
    - Maintain high availability and performance
    - Optimizing costs

Additional References

- Metric - a quantitative measurement used to track specific aspects of a cloud system, such as CPU usage,

memory consumption, or network traffic.

- Log - a record of system activity, typically capturing detailed information about events, operations, and error messages.
- Event - an occurrence or action within a cloud system, like a user request, a change in configuration, or a system warning.
- Load - the demand placed on a cloud system, typically measured by the number of requests, processing tasks, or resource usage.
- Trigger - a condition or event that initiates a specific action or response in a cloud system, such as scaling, alerting, or automated workflows.
- Triggered scaling based on Trends
    - Example - Autoscaling in Azure App Service Based on CPU Utilization
        - ♠ Setup - Deploy a basic web application on Azure App Service.
        - ♠ Configuration - Set up autoscaling to increase or decrease the number of instances based on CPU utilization. You can set the scale-out threshold to a specific percentage (e.g., 70%) and the scale-in threshold to a lower percentage (e.g., 30%).
        - ♠ Demonstration - Simulate traffic to the web app and observe how the instances scale in response to CPU trends. This example demonstrates triggered scaling based on trends in resource usage.
- Triggered scaling based on "Load"
    - Example - Horizontal Pod Autoscaling in Azure Kubernetes Service (AKS)
        - ♠ Setup - Create a simple AKS cluster with a basic application running in pods.
        - ♠ Configuration - Configure the Horizontal Pod Autoscaler to scale based on resource load, such as CPU or memory usage. Set thresholds for scaling out and in.
        - ♠ Demonstration - Apply load to the application to increase CPU usage, and watch as additional pods are created. As the load decreases, observe how pods are removed. This example illustrates triggered scaling based on load.
- Triggered scaling based on "Events"
    - Example - Azure Functions with Event-Driven Scaling
        - ♠ Setup - Create an Azure Function that is triggered by a specific event or a message in a queue.
        - ♠ Configuration - Configure the scaling settings to ensure the function scales out when more events occur. Azure Functions can automatically scale based on the number of incoming events.
        - ♠ Demonstration - Send multiple HTTP requests or add several messages to a queue, and observe how the Azure Function scales to process the events. This example shows how event-driven scaling works in Azure.
- Scheduled scaling
    - Example - Azure App Service with Scheduled Scaling
        - ♠ Setup - Deploy a web application to Azure App Service.
        - ♠ Configuration - Set up autoscaling rules based on a schedule. For example, increase the number of instances during peak hours and decrease them during off-peak times.
        - ♠ Demonstration - Configure the schedule for scaling and then verify that the number of instances changes according to the specified times. This example demonstrates how scheduled scaling can be used to optimize resource usage.
- Manual scaling
    - Example - Manual Scaling of Azure Virtual Machines
        - ♠ Setup - Create an Azure Virtual Machine and choose a basic VM size.
        - ♠ Configuration - Manually adjust the VM size or add additional VM instances to scale. You can change the VM SKU to increase capacity or add more VMs to scale out.
        - ♠ Demonstration - Manually scale the VM by changing its size or adding/removing instances. This example illustrates manual scaling, allowing beginners to understand how to adjust resources manually.