Assignment 5: Resilient Architecture with Availability of 4 9s (99.99%) Scenarios

As a Cloud Architect for an e-commerce company, you are tasked to design a resilient architecture with an availability of 4 9s (99.99%). This architecture should be able to handle high traffic, provide a seamless customer experience, and prevent downtime even during peak shopping seasons.

To accomplish this, you need to perform the following tasks:

1. Identify the critical components of the e-commerce platform that need to be highly available.
2. Determine the service level objectives (SLOs) for each critical component. SLOs stands for Service Level Objectives. It refers to the measurable objectives or targets that define the level of performance or availability that a service must meet. SLOs are used to define expectations and requirements for the quality of service provided by an application or system, and are often used in conjunction with Service Level Agreements (SLAs) that define the expectations and agreements between the service provider and the customer.
3. Develop a resilient architecture design to meet the SLOs for each critical component, using AWS services and best practices.
4. Identify potential failure scenarios and develop a plan to mitigate them.
5. Conduct load testing to validate the performance of the architecture design.
6. Document the resilient architecture design and present it to the stakeholders.

To complete this assignment, you should consider the following questions and checklist:

## Questions:

1. What are the critical components of the e-commerce platform that need to be highly available?
2. What are the current service level objectives (SLOs) for each critical component?
3. What are the potential failure scenarios for the critical components?
4. What AWS services and best practices can be used to develop a resilient architecture design?
5. How can load testing be conducted to validate the performance of the architecture design?
6. How can the resilient architecture design be documented and presented to stakeholders?

## Checklist:

1. Identify critical components of the e-commerce platform that require high availability.
2. Define SLOs for each critical component.
3. Develop a resilient architecture design that meets SLOs, using AWS services and best practices.
4. Identify potential failure scenarios and develop a plan to mitigate them.
5. Conduct load testing to validate the performance of the architecture design.
6. Document the resilient architecture design and present it to stakeholders.

## Deliverables:

1. A comprehensive report on the resilient architecture design, including a list of critical components, SLOs, potential failure scenarios, and mitigation plans.
2. Documentation of the load testing results and performance of the architecture design.
3. A presentation to stakeholders that explains the resilient architecture design and its benefits.

## Questions to ask:

1. What are the critical services or components that require high availability?
2. What is the current availability level of the system?
3. What are the causes of downtime or unavailability of the system?
4. What are the expected or required uptime and availability levels?
5. What are the potential risks and impacts of downtime or unavailability?

## Checklist steps:

1. Identify critical services or components that require high availability.
2. Assess the current availability level of the system.
3. Identify potential risks and causes of downtime or unavailability of the system.
4. Determine the required or expected uptime and availability levels.
5. Develop and implement a resilient architecture that can provide the required availability level, such as:
6. a. Implementing redundancy and failover mechanisms for critical services or components.
7. b. Implementing automatic scaling and load balancing mechanisms to handle sudden spikes in traffic or demand.
8. c. Implementing monitoring and alerting mechanisms to detect and respond to failures or issues quickly.
9. d. Implementing disaster recovery and backup mechanisms to ensure data and system availability in case of disasters or outages.
10. Continuously monitor and assess the system's availability level and performance, and make necessary adjustments and improvements to maintain or improve the availability level.

## Steps to 99.99 % Availability

To achieve a resilient architecture with availability of 4 9s (99.99%) for an e-commerce company, the following steps can be taken:

1. Identify critical services: Identify the critical services and components of the e-commerce application that are essential for the business to operate. These could include the web server, application server, database server, payment gateway, inventory management system, and others.
2. Determine acceptable downtime: Determine the maximum acceptable downtime for each critical service. This will help in determining the required availability percentage for each service.
3. Determine current availability: Determine the current availability percentage of each critical service. This will help in identifying the areas that need improvement.
4. Identify potential points of failure: Identify potential points of failure in the architecture and take steps to mitigate them. For example, redundant hardware and software, load balancers, and failover mechanisms can be put in place to minimize downtime.
5. Design for scalability: Ensure that the architecture is designed for scalability to handle sudden spikes in traffic and load. This can be achieved through the use of auto-scaling groups, load balancers, and elastic infrastructure.
6. Design for resiliency: Ensure that the architecture is designed for resiliency to handle failures of individual components or services. This can be achieved through the use of redundant hardware and software, multi-zone deployments, and disaster recovery mechanisms.
7. Monitor and measure: Implement monitoring and measurement tools to track the availability of critical services and identify any issues before they impact the users. This can be achieved through the use of tools like Amazon CloudWatch, New Relic, and Datadog.
8. Continuously improve: Continuously improve the architecture by identifying areas of improvement and implementing changes to increase the availability percentage.
9. Test and validate: Test and validate the resilience and availability of the architecture through various scenarios like load testing, failover testing, and disaster recovery testing.

## Example:

For an e-commerce company like Amazon, ensuring high availability is crucial for its business operations. Amazon has designed its architecture to be highly available and resilient by implementing a number of strategies such as:

* Multi-zone deployments: Amazon deploys its critical services across multiple availability zones (AZs) to ensure high availability and resilience. Each AZ is an isolated location with its own power and network infrastructure, providing redundancy and failover capabilities.
* Auto-scaling groups: Amazon uses auto-scaling groups to automatically adjust capacity based on demand. This ensures that the application can handle sudden spikes in traffic and load without impacting the availability of the services.
* Load balancing: Amazon uses load balancers to distribute traffic across multiple instances of its services. This ensures that the load is evenly distributed and helps to prevent individual instances from being overloaded.
* Redundant hardware and software: Amazon uses redundant hardware and software to ensure high availability and resilience. This includes using multiple servers and storage devices to store data and running services in redundant configurations.
* Disaster recovery mechanisms: Amazon has implemented disaster recovery mechanisms to ensure that critical services can be recovered quickly in case of a disaster. This includes using backups and replicating data to multiple locations.

By implementing these strategies, Amazon has been able to achieve a highly available and resilient architecture with availability of 4 9s (99.99%).