



Assignment NO: 1

Q.1.

Use S3 bucket and host video streaming.



1) creation of s3 bucket

1] Go to Aws academy and search for S3 bucket. Then click on it there you will get a button to click to create a bucket.
2] Give your bucket a unique name to your bucket. Here I have given name "video-storage-streaming". There is no other changes made in general configuration.

3] Also there is no change in object ownership and keep block public access on.

4] Then Enable Bucket versioning . After that enable Default encryption also.

5) Do not enable Advanced settings. Keep it disabled by default.

6] After clicking confirm this , "video - storage-streaming" named bucket got created.

7] click on upload button.

2) Uploading A video

1] Then you can drag and drop the video.mp4 file or you can add it by using Add files.

2] Then click on the upload button. Then check the status of video file whether it uploaded successfully

3] Thus, video got uploaded successfully.

4) Then click permissions and then off the block public access ie. providing public access. then click on save changes.

5] Then again go to permissions of the project of the project and look at Access Control List (ACL) whether it has read write access to it. Then go back to S3 main page select your video and click FOR EDUCATIONAL USE

on properties where you will get link click on it but it shows access denied.
Thus we need to update bucket policy by using the following code. In this click on Edit button and then add script for adding the code for bucket - policy.

{ "version": "2012-10-17",

"Statement": [

1

"Effect": "Allow",

"principal": "*",

"Action": "s3.GetObject",

"Resource": "arn:aws:s3:::video-storage-streaming/*"

2

]

3

7] go back on s3 main page the refresh the page

8] Then select the video that you uploaded and click on properties where you will get this type of link (video-storage-streaming.s3.amazonaws.com/321448-4hd_3840_2160_25fps.mp4)

copy this link and paste in new tab, Now you can see the video you have uploaded.

9] The video stream hosted using s3 bucket successfully.

Q.2. Discuss BMW and Hotstar case studies using AWS.

→ BMW, is one of the leading global manufacturers, which use AWS to enhance their digital transformation efforts and improve customer experiences. BMW uses AWS through connected drive platform, which provides cloud-based services to millions of BMW vehicles worldwide. Using AWS IoT, BMW connects and manages millions of car sensors and system in real-time, delivering personalized driving experiences and optimizing vehicle performance. Additionally, BMW uses AWS's data analytics capabilities, including Amazon S3 and Redshift, to collect and analyze large amount of vehicle data for predictive maintenance, improving safety and efficiency. AWS's machine learning services, such as Amazon SageMaker, helps BMW to deploy advanced AI technologies in automated driving, voice recognition and intelligent in-car assistants. By utilizing Amazon EC2 and Elastic Load Balancing, BMW can innovate faster, rolling out software updates globally with greater ability. The benefits for BMW include enhanced global scalability, faster innovation, reduced operational costs, and data-driven insights that improve both vehicle performance and the overall customer experience.

Hotstar now known as Disney+ Hotstar, is one of the India's largest streaming platform and relies heavily on AWS to manage its massive audience, especially during live-streaming events like IPL. Hotstar faced the challenge of handling traffic during major events, such as in 2020 IPL final, there were more than 30 million viewers. So as to manage such high demand, Hotstar uses AWS auto-scaling, allowing its infrastructure to automatically adjust in real-time to support traffic, ensuring uninterrupted service during peak times.

AWS's CloudFront provides Hotstar with a reliable content delivery network, ensuring low latency and high transfer speeds to views over worldwide. Hotstar also utilizes AWS's data analytics services like Amazon Kinesis to analyse real-time viewer data, which helps in delivering personalized content recommendations, improving engagement and retention. Additionally, AWS's Elastic Load Balancing and global data center network help Hotstar ensure high availability and a seamless streaming experience, even during high-traffic events. Hotstar maintains cost efficiency by scaling infrastructure based on real-time needs.

In conclusion, both BMW and Hotstar have successfully utilized AWS to scale their operations, improve customer experience. BMW enhanced its connected car technology with AWS's cloud infrastructure, while Hotstar handled massive live stream traffic using AWS's auto-scaling.

FOR EDUCATIONAL USE

Q.4. what is Nagios and explain how Nagios are used in E-services?

→ 1) Nagios is an open-source monitoring system designed to monitor the performance and availability of IT infrastructure components, including servers, applications, and network services. It provides real-time insights into the status of these components, enabling system administrators and IT professionals to identify and address issues before they get converted into major problems.

2) Features of Nagios:

1] Nagios can monitor various system metrics, including CPU load, disk usage, memory consumption and network bandwidth.

It can check the availability of services such as HTTP, SMTP, FTP.

2] Alerting systems: Nagios has a robust alerting system that notifies users via email, SMS, or other methods when issues arise, ensuring timely responses to critical problems.

3] Web Interface: It offers a web-based interface that provides easy access to monitoring data, making it simple for users to visualize the health and performance of their systems.

4] Plugins: Nagios supports a wide range of plugins which provides functionality, allowing users to customize monitoring for specific applications and services.

3) Use of Nagios in E-services:

1] Server monitoring: Monitor the health and performance of servers, including CPU usage, memory utilization, disk space and system load.

2] Networking monitoring: Monitor network devices such as routers, switches and firewalls for availability and performance. It tracks network traffic, bandwidth usage, and detect network anomalies.

3] Service availability: Ensures availability of critical services and

applications. It monitors web servers, databases, email servers and other services to detect and respond to outages.

- 4) Resource Utilization: It tracks resource utilization metrics to identify trends and potential bottlenecks. It monitors the usage of CPU, memory, disk I/O, and other resources on monitored systems.
- 5) Performance Metrics and Trend Analysis: It collects performance metrics over time to analyze trends and forecast potential issues. Use historical data for capacity planning and optimization.
- 6) Log File Monitoring: It monitors log files for specific events, errors or patterns. It also sets up alerts based on log file entries to identify issues in real-time.
- 7) Alerting and Notification: It sets up alerts to notify administrators or support teams when predefined thresholds are breached.
- 8) Application Monitoring: It monitors the performance and availability of custom applications. It integrates Nagios with plugins or agents to collect application-specific metrics.
- 9) It also monitors database servers for performance metrics, query execution times and overall database health.
- 10) We can extend Nagios to monitor cloud-based architecture in platforms like AWS, Azure or Google Cloud to monitor virtual machines, storage and other cloud services.

Q.3. Why kubernetes and advantages and disadvantages of kubernetes. Explain How adidas use kubernetes.

→ Kubernetes is an open-source platform designed to automate the deployment, scaling, and operation of application containers across cluster of hosts. Originally developed by google, kubernetes simplifies the management of containerized applications, ensuring they run efficiently across different environments without manual intervention. kubernetes group containers into logical units called pods, which can be easily scaled and managed. It abstracts the underlying infrastructure, making it easier to run applications in a cloud-native environment.

Advantages:

1. Automated Deployment and Scaling: kubernetes can automatically deploy and scale your applications based on traffic load or predefined rules. This reduces the need for manual intervention in managing app lifecycle changes.
2. Self-Healing: If a container fails, kubernetes will restart it automatically, ensuring high availability. It also reschedules or replaces containers when nodes fail.
3. Load Balancing and Service Discovery: It provides load balancing and service discovery features to distribute traffic to the appropriate containers without manual configuration.
4. Efficient Resource Utilization: kubernetes efficiently manages compute resources by scheduling containers to run on nodes that can fulfill resource requirements. It helps optimize the use of hardware.
5. Portable and Cloud-Agnostic: Applications running on kubernetes can be deployed on any platform. This provides flexibility.

6. Declarative configurations: Kubernetes uses YAML/JSON files to manage deployments. This allows you to declare the desired state of your application and Kubernetes ensures the actual state matches this configuration.

7. Rollback and Rollouts: Kubernetes allows you to easily roll back to previous versions of your application or roll out updates with zero downtime.

8. Extensibility: Kubernetes supports custom resource definitions and various plugins/extensions, which makes it adaptable for specialized workloads.

Disadvantages:

1. complexity: Kubernetes has many features, setting and configuring it is complex.

2. Running Kubernetes can introduce overhead.

3. Managing Kubernetes networking and security policies can be complex. Setting up network policies, encryption and securing communication between containers can be difficult.

4. While Kubernetes automates a lot of tasks, monitoring, debugging and diagnosing can be challenging.

5. For smaller, simpler applications or those that don't need to scale across multiple nodes, Kubernetes might be overkill. It's better suited for large, complex or highly scalable applications.

How adidas uses Kubernetes

- 1) Adidas adopted Kubernetes to enhance its digital platform, primarily supporting its e-commerce website and backend services.
- 2) Adidas shifted from monolithic to microservices, using Kubernetes to manage independent services and ensure faster releases.
- 3) During high-demand events like product launches, Kubernetes scales Adidas services automatically, preventing downtime.
- 4) Adidas streamlined its continuous Integration / continuous Deployment (CI/CD) process with Kubernetes automating deployments and feature updates.
- 5) Adidas leverages Kubernetes to run workloads on cloud platforms like Google Cloud, enhancing flexibility.
- 6) Kubernetes self-healing capabilities ensure Adidas applications stay online, even if components fail.
- 7) Kubernetes allows Adidas development teams to deploy updates and new features more frequently. With containerized microservices, they can work independently and avoid downtime during releases.
- 8) As Kubernetes automatically adjusts the number of running containers based on real-time traffic, so with this Adidas platform handles the increased load without crashing.
- 9) Kubernetes schedules workloads efficiently across clusters.
- 10) Kubernetes enables Adidas to deploy its services across different regions, improving site performance.
- 11) Kubernetes helps Adidas secure its platform by isolating services in containers. This containment minimizes the risk of security breach affecting the entire system.

These highlights how Adidas leverages Kubernetes to maintain a competitive edge in the e-commerce space with highly scalable, secure and efficient platform.