Name: Sandesh Kumar M. Yadav Sub: Adv DevOps
Roll: 61
Div: D15C
05/05

Assignment No. 1

Q1. Use S3 bucket and host video streaming?

→ steps to host video on AWS S3 bucket:

1. Download any sample video from the internet.
2. Now break video into smaller segments so that it can be easily transmitted over the internet.
3. To make smaller chunks we would use ffmpeg tool, the general syntax is ffmpeg -i <VIDEO-NAME> -profile: v baseline -level 3.0 -start_numbers 0 -hls_time 35 -hls_list_size 0 -f hls output.m3u8
4. Ensure you download the ffmpeg before executing above command.
5. After that create a new S3 bucket, make it public.
6. Now, change bucket policy to anyone can access.
7. We also need to setup cors policy so that any endpoint can request for resources of bucket.
8. After setting up all necessary configuration, we need to upload video segments that we had created previously
9. Create a new folder named 'hls' inside bucket and upload all the video segments in it.
10. Now we would create a simple HTML document that would be hosted on S3 bucket so that video can be played
11. The HTML file would contain the link on the main playlist of the video segments
12. Open the link provided inside object properties.
13. The video will start streaming.

Q2. Discuss BMW and Hotstar case studies
→ BMW Case study with AWS :
Background :
BMW is a global automobile manufacturer to offer cutting-edge connected car services innovation. with over 100 million connected BMW needed a highly scalable infrastructure the increasing data from their connected and provide seamless customer services
Key challenges :
① Data Handling : BMW's connected cars large amount of data. They needed a system could handle this in real time.
② Global Scalability : BMW needed to scale to serve customers worldwide, providing services like navigation, remote software and infotainment.
③ Agility and Innovation : The company accelerate innovation by reducing infrastructure management overheads and fostering a DevOps
Solution :
BMW choose AWS to power its connected which collects, processes, and analyzes data of vehicles.
1. Scalable infrastructure : BMW uses AWS like Amazon S3, Amazon EC2, and AWS Lambda Store, process and analyze vehicle data at scale This helps them efficiently manage huge datasets

Key challenges:
1. Massive Traffic spikes: Hotstar faces unpr spikes in traffic during live events like the where millions of users stream simultaneously
2. Low Latency and High Availability: For live Streaming, low ~~for~~ latency and uninterrupted are critical for user satisfaction.
3. Scalability: Hotstar needed the infrastruct support over 100 million concurrent viewers w maintaining quality streaming.

Solution:
1. Auto-Scaling and Elasticity: Using Amazon EC2 auto-scaling features, Hotstar can dynamically resources based on viewer traffic. During liv Hotstar scales up to handle millions of concurr
2. Content Delivery with AWS CloudFront: Hotst Amazon CloudFront to distribute content globally, low latency and smooth streaming for users in remote areas.
3. Data Analytics: Hotstar uses Amazon Redshift time data analytics to monitor user engagement live events. This helps them make real-time adj to improve the streaming experience.
4. Database and storage: Hotstar uses Amazon Amazon S3 to manage large amounts of s and unstructured data efficiently.

Result
1. R
over
live
2. Cos
helped
resource
3. Rel
Elastic
and h
a smo

Conclusi

Both
Scalo
unique
1. P
massi
deliv
2. L
Conten
exper

This
Cater
whet

by Connected vehicles.

2. Data Lakes : Using Amazon S3, BMW built a global data lake to store and analyze terabytes of data from their vehicles in real-time.

3. DevOps and Agility : BMW adopted a microservices architecture with AWS Lambda, enabling faster innovation cycles and reducing time-to-market for new features in their cars.

Results :

Scalability : BMW successfully scaled its Connected car platform, supporting over 14 million connected vehicles globally.

Cost Efficiency : AWS's pay-as-you-go model allows BMW to optimize costs while expanding its platform.

Innovation Acceleration : By automating infrastructure management, BMW's engineering team can focus on innovating new features for customers.

Hotstar's Case Study with AWS

Background :

Hotstar (now known as Disney+ Hotstar) is a leading video streaming platform in India, providing live streaming of sports, TV shows, and movies. During major live sports event, like the IPL, Hotstar sees spikes in viewership that need to be handled seamlessly

Q3. Why Kubernetes and advantages and disadvantages of Kubernetes. Explain How adidas uses Kubernetes.

→ Kubernetes is an open-source platform that automates deployment, scaling and management, of containerized applications, providing a consistent way to manage workloads across various environments.

Advantages of Kubernetes:

(1) Automated Scaling: Adjusts resources based on demand.
(2) portability: Runs across any infrastructure (cloud, on-premises).
(3) High availability: Self-heals and ensures uptime.
(4) Load Balancing: Distributes traffic efficiently across containers.
(5) Automated updates: Smooth rollouts and rollbacks of applications.
(6) Resource Efficiency: Maximizes hardware usage.
(7) Extensibility: Supports plugins for additional features like monitoring and security.

Disadvantages of Kubernetes:

(1) Complexity: Difficult to learn and set up.
(2) overhead: Adds resource overhead for small projects.
(3) Requires Expertise: Demands skilled engineers for management.
(4) Cost: Can be expensive to operate at small scale.
(5) Networking complexity: Customization may be complex.

Results:

1. Record-Breaking Concurrency: Hotstar achieved over 25 million concurrent viewers during a single live sport event, setting a global record.
2. Cost optimization: AWS's auto-scaling features helped Hotstar optimize infrastructure costs by scaling resources up and down based on demand.
3. Reliable streaming Experience: With AWS cloudfront and Elastic Load Balancing, Hotstar maintained low latency and high availability during critical live events, ensuring a smooth viewing experience for millions of users.

Conclusion:

Both BMW and Hotstar have successfully utilized AWS's scalable and flexible cloud services to address their unique challenges:

1. BMW built a connected car platform that can handle massive amounts of real-time data, accelerating its digital transformation.
2. Hotstar leveraged AWS's elastic infrastructure and Content delivery Services to provide a reliable streaming experience, even during peak traffic events like IPL.

This case studies highlight the versatility of AWS in catering to different industries with unique demands - whether it's connected cars or live sports streaming.

Q4. What are Nagios and explain how Nagios used in E-services

→ Nagios is an open-source monitoring tool oversee the health and performance of sy networks, and applications. It tracks uptime, usage, and service availability, providing al email or SMS when issues arise, allowing f resolution to prevent downtime.

Key features:

(1) Monitoring: Tracks servers, applicatio network devices.

(2) Alerts: Notifies administrators about issue service outages.

(3) performance metrics: Monitors CPU, memory, network traffic.

(4) plugins: Supports custom plugins for s monitoring.

(5) web-interface: provides a user-friendly d

Nagios in E-services:

E-services, like e-commerce and online banking, high uptime and performance. Nagios helps ensu reliability by providing real-time monitoring, proact alerts, and resource optimization.

(1) Real-Time Monitoring: Tracks the status of

for hybrid or multi-cloud environments.

How Adidas Uses Kubernetes:
Adidas uses Kubernetes to manage its e-commerce platform, providing:

1. Scalability: Handles traffic spikes during events.
2. Microservices: Shifts from monoliths to independently deployable services.
3. Multi-cloud: Supports a flexible multi-cloud strategy.
4. Development-efficiency: streamlined environments and speeds up releases.
5. Cost Efficiency: optimizes resource usage and reduces cloud costs.

Adidas benefits from Kubernetes scalability, flexibility, and efficient resource management to meet global demand and enhance their development process.

databases, and networks, ensuring smooth operation.

(2) Proactive Alerts : Sends alerts for issues such as downtime or high resource usage, allowing for immediate response.

(3) SLA Monitoring : Ensures compliance with service-level agreements by tracking uptime and performance.

(4) Resource optimization : Helps avoid crashes by monitoring performance.

(5) Security Monitoring : Detects potential security threats, such as unauthorized access or abnormal traffic.

(6) Scalability : Easily scales as the e-service grows, supporting larger environments.

(7) Historical Reporting : Stores data for reporting and future planning.

Example in E-services :

In an e-commerce platform, Nagios monitors web servers, databases, payment gateways, and user transactions, ensuring high availability and performance for customers.

Conclusion :

Nagios ensures reliability and performance in e-services by monitoring infrastructure, providing real-time alerts, and optimizing resource usage, ensuring a seamless user experience.