

## Assignment 1

Use S3 bucket to host video streaming.

- Step to host video on new S3 bucket
- ① Download any sample video from internet
  - ② Now break whole video into smaller segments so that it can be easily transmitted over network.
  - 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-number 0 -t 10 3.5-hls-list-eo -f hls output -m3u8`.
  - ④ Ensure you download the ffmpeg before executing above command.
  - ⑤ After that create a new S3 bucket, name it & configure it. give public access.
  - ⑥ Now change bucket policy so that anyone can access bucket object.
  - 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 previous.
  - ⑨ Now we would create a sample HTML document that would be hosted on S3 bucket so that video can be played.

- (11) The HTML file would contain the link of master playlist of video segments.
- (12) Open the link provided. Inside properties
- (13) The video will start streaming.



## Adu DevOps Assignment 1

Q2] Discuss BMW & Hotstar case study using AWS.

Ans:

Leveraging AWS for Scaling BMW's Connected Car Services & Hotstar's Live Streaming Infrastructure.

Overview:

Both BMW & Hotstar faced challenges in scaling their digital platforms to meet growing demands. BMW needed a solution to handle real-time data for millions of connected cars, while Hotstar sought infrastructure capable of supporting millions of concurrent users during live sports events. AWS provided each with scalable, flexible, & cost-effective solution to address their unique challenges.

The Challenges:

BMW Group:

BMW aimed to improve its connected car platform, which collect & process data from millions of vehicles. They needed to ensure:

- Real-time telemetry data process from vehicles.
- Efficient handling of services like over-the-air update & predictive maintenance.
- Scalable infrastructure to manage growing vehicles data.

Hotstar

Hotstar, a leading streaming platform in India, faced the challenge of supporting live sports events with massive viewer spikes. During cricket matches & other popular events, Hotstar needed to:

- Serve millions of concurrent users.
- Maintain low latency & high video quality.

- Scale their infrastructure on demand without incurring excessive costs

### The Solution (Using AWS)

BMW's challenge.

To meet these challenges, BMW turned to AWS for cloud-based infrastructure that could process & store massive amounts of data in real-time. The key services included:

- Amazon S3: For scalable, secure storage of vehicle telemetry data
- AWS Lambda: To process real-time data from vehicles without the need for dedicated servers.
- Amazon RDS & DynamoDB: For managing & accessing vehicle & customer data at scale, ensuring availability & performance.

BMW's connected cars use these AWS services to offer features like predictive maintenance, improved vehicle performance, & remote software updates, enhancing the user experience.

Holster:

To manage its massive user base during match seasons or events.

- Amazon CloudFront: A content delivery network (CDN) to provide low-latency video streaming for millions of concurrent viewers.
- Amazon EC2: To dynamically scale computing power during traffic spikes for live encoding and streaming.
- Amazon S3: For storing video content to stream both live & on demand.
- AWS Lambda & Elastic Load Balancing: To automatically adjust infrastructure based on demand, ensuring smooth performance during



lead viewership  
• Amazon Kinesis: To process real-time data from user behaviour, enabling insights into viewer engagement & personalised recommendations.

### Results & Benefits

Bmw Group.

By using AWS, BMW achieved:

- Scalability: AWS allowed BMW to scale its Connected Car platform to handle data from millions of vehicles globally.
- Real-Time Processing: Vehicle data is processed in real-time, enabling features like predictive maintenance & improved vehicle diagnostics.
- Focus on Innovation: BMW reduced the complexity of managing infrastructure, allowing its team to focus on new services & features.

### Hotstar

Hotstar experienced significant benefits from AWS:

- Massive Scalability: Hotstar successfully supported over 25 million concurrent views during peak live sports events like cricket matches.
- Global Availability: AWS's global infrastructure allowed Hotstar to stream content reliably to users worldwide, ensuring high-quality video with minimal latency.
- Cost-Efficient: The pay-as-you-go pricing model helped Hotstar scale up during events & reduce costs during off-peak hours, optimising their spending.



### Conclusion:

Both BMW & Motostar leveraged AWS to address their unique challenges - BMW in handling real-time connected car data, & Motostar in managing massive live-streaming traffic. AWS provided a scalable, reliable, & cost-effective infrastructure that allowed both companies to focus on delivering enhanced user experiences, innovate rapidly, & ~~seamlessly~~ scale their platform globally.

Q3] Why Kubernetes? Advantages & disadvantage of Kubernetes. Explain how ~~adidas~~ uses Kubernetes.

Ans: Kubernetes is open source platform for automating the deployment, scaling & management of containerized applications. It orchestrates & manages containers, which allows for efficient utilization of resources, scalability & resilience.

Why Kubernetes?

- 1] Container Orchestration: Kubernetes helps manage a large no. of containers across multiple environments, making it easier to deploy & scale applications efficiently.
- 2] Scalability: It can automatically scale applications up & down based on demand.
- 3] Self-Healing: Kubernetes ensures that failed containers are automatically restarted & unhealthy containers are replaced.
- 4] Declarative Configuration: You define the desired state of your



- Application, & Kubernetes ensures it is always maintained
- 1] Automation: It automates many operational tasks, such as deployment, scaling, load balancing & monitoring, reducing manual intervention.
  - 2] Multi-cloud & Hybrid: Kubernetes allows you to run workloads across multiple environments whether on-premises or the cloud or both.

### Advantages of Kubernetes.

- 1] Portability: Kubernetes runs containers across various environments, making it applications portable & flexible across cloud.
- 2] Efficient Resource Utilization: It optimizes resource utilization by intelligently scheduling containers across nodes.
- 3] Automated Rollouts & Rollbacks: Applications can be updated without downtime, with automatic rollback in case of failure.
- 4] Self-healing: Failed containers are automatically replaced & restarted, ensuring high availability.
- 5] Service Discovery & Load Balancing: It provides a built-in mechanism to expose containers to external traffic & balances load automatically.
- 6] Extensibility: Kubernetes integrates with many other systems (storage, networking) & support custom workflows with extensions & plugins.
- 7] DevOps Integration: Well-suited for modern DevOps practices like Continuous Integration & Continuous Delivery (CI/CD).

### Disadvantages of Kubernetes:

- 1) Complexity: Kubernetes has a steep learning curve, & its complexity can make it difficult to setup & maintain.
- 2) Resource-Intensive: It requires significant computing resources to run the control plane components (API, etcd, scheduler, etc.) which may be overkill for small applications.
- 3) Management Overhead: Managing Kubernetes clusters, networking, security & storage can require additional expertise & administrative effort.
- 4) Not Always Necessary: For small or simple applications, Kubernetes might be an over-engineered solution when simple orchestration tool like Docker Compose could suffice.
- 5) Latency: The orchestration & scheduling process might add some latency compared to running containers directly on the host machine.
- 6) Cost: Running Kubernetes clusters, especially in cloud environment, can be expensive due to the infrastructure required.



## Case Study: Adidas

Shaping the future culture, adidas got used of its most impactful System Reengineering Kibernetes in 2018

### Challenges:

Adidas engineers faced significant delays and complexity in accessing development tools. The process to get something done was involved multiple internal steps, which would make anyone from half an hour to a week. This inefficiency hindered the productivity of the engineering team who often felt restricted by these time-consuming procedures.

### Solution

To address this, the Adidas platform engineering team adopted a ~~devOps~~ first approach. They implemented containerization, agile development, continuous delivery, and cloud-native platform to integrated Kibernetes to DevOps. Adidas partnered with Google Cloud to help design a secure Kubernetes cluster across a and on various environments. This move simplified the process, significantly speeding up project delivery & infrastructure management.

### Impact

The adoption of Kibernetes to other cloud-native technology had profound impact on Adidas

- Deployment frequency increased from every 4-6 weeks to 3 times daily.
- The lead time for the e-commerce site was cut in half.

user experience

- Kubernetes now manages 40% of Adidas' most critical system, handling 4000 pods, 200 Nodes & 10000 build jobs each.
- This transformation allowed Adidas developers to focus building fast, scalable, secure platform without worrying about infrastructure management.

'David Eichten', a senior director of ~~front~~ platform engineering at Adidas states that:

"I call ~~our~~ cloud native platform the fruit of dreams we built it. I was never anticipated that people would recognize it ~~as~~ <sup>as</sup> ~~moment~~ just love it."

Key Lesson:

Adidas' transformation highlights ~~the~~ the importance of simplifying the developer experience to increase efficiency. Cloud-native tech like Kubernetes & Prometheus enable greater agility, faster development cycle & enhanced performance. Fostering collaboration & education across teams ensure smooth adoption of these technologies. Adidas experience emphasizes that successful digital transformation requires aligning technology solution with company culture, as there is no one-size-fits-all approach.



Q. What are Nagios & explain how Nagios are used in E-Commerce?

Ans:

Nagios is an open source monitoring tool used to monitor systems, networks, & infrastructure. It provides real-time monitoring, alerting, & reporting for services, devices & applications. Nagios helps system administrators track the performance & health of the IT environment, ensuring that any failures, slowdowns or errors are promptly detected & resolved.

Key features

- 1) Monitoring: Nagios monitors various components like services, applications, network devices & servers (HTTP, SMTP, DNS, etc)
- 2) Alerting: When problem is detected, Nagios can send alerts via mail.
- 3) Scalability: It can monitor large, distributed infrastructure with multiple hosts.
- 4) Customizable: Nagios allows users to define custom checks & integrate with specific requirements.
- 5) Plugins: Nagios supports plugins that extend its capabilities for monitoring additional services or devices.
- 6) Web Interface: It provides a web-based interface for viewing the current status of the systems, ~~history~~ historical data & logs.

How Nagios is used in E-Service.

In the context of e-service, such as e-commerce platforms, financial services, & digital applications, Nagios plays a critical role in ensuring uninterrupted service & optimal performance. Here how it is applied:

- 1] Website & Application Monitoring: Nagios monitors web services to ensure that online platforms are running smoothly. It checks if the service is reachable, monitors load times, & flags any downtime.
- 2] Server Monitoring: Track server performance, include memory, CPU, disk space, & network traffic. Alerts are triggered for any threshold breach.
- 3] Database Monitoring: Ensure database health, query performance, & storage, preventing slowdowns & data issues.
- 4] Network Monitoring: Monitor network devices for latency, bandwidth, or packet loss, ensuring smooth connectivity.
- 5] Service Availability: Check critical services like payment gateway & email services to ensure uptime for smooth transactions.
- 6] Security Monitoring: Monitor logs & security activities, helping detect suspicious behaviour.
- 7] Scalability Monitoring: Track the performance of cloud clusters, containers, & VMs as e-services scale.
- 8] Incident Management: Provide real-time alerts & integrate with incident system for quick issue resolution.