**International Academy of Science,
Engineering and Technology**
Connecting Researchers; Nurturing Innovations

IASET

BUILDING SCALABLE MICROSERVICES ARCHITECTURES ON AWS: BEST PRACTICES AND LESSONS LEARNED

Ravi Laudya[1], Rahul Arulkumaran[2], Dr S P Singh[3], Ravi Kiran Pagidi[4], Shalu Jain[5] & Prof. (Dr) Punit Goel[6]

[1]Indian Institute of Science, Bangalore, India

[2]University at Buffalo, New York, Srinagar Colony, Hyderabad, 500073, India

[3]Gurukul Kangri University, Haridwwar, Uttarakhand, India

[4]Jawaharlal Nehru Technological University, Hyderabad, India

[5]Maharaja Agrasen Himalayan Garhwal University, PauriGarhwal, Uttarakhand, India

[6]Maharaja Agrasen Himalayan Garhwal University, Uttarakhand, India

ABSTRACT

Building scalable microservices architectures on AWS has become essential for organizations aiming to achieve agility, reliability, and rapid deployment. This paper explores best practices and lessons learned from designing and managing microservices on AWS. A microservices architecture breaks down complex applications into loosely coupled, independently deployable services, each focusing on specific business functionalities. AWS offers a robust cloud ecosystem with services like Amazon ECS, EKS, Lambda, and API Gateway, which streamline the development and deployment of microservices.

This study highlights the importance of adhering to core architectural principles such as decoupling, statelessness, and resilience to build scalable systems. It discusses the role of auto-scaling, load balancing, and serverless functions in optimizing performance under varying workloads. Furthermore, it emphasizes the significance of observability through monitoring tools like AWS CloudWatch and distributed tracing with AWS X-Ray for effective troubleshooting. Best practices for securing microservices using identity management tools like AWS IAM and securing communications with encryption protocols are also covered.

Challenges, including managing service dependencies, ensuring data consistency, and minimizing latency, are explored along with effective mitigation strategies. Additionally, adopting a DevOpsmindset, implementing CI/CD pipelines, and leveraging Infrastructure-as-Code tools such as AWS CloudFormation are presented as essential strategies for continuous improvement and automation. The paper concludes by providing insights into lessons learned from real-world scenarios, helping businesses overcome common pitfalls while building robust and scalable microservices on AWS. This exploration aims to serve as a valuable resource for organizations striving to enhance their cloud architectures through microservices.

KEYWORDS: *Microservices Architecture, AWS, Scalability, Auto-Scaling, Serverless, API Gateway, Cloud-Native, Observability, Resilience, CI/CD Pipelines, Infrastructure-As-Code, AWS CloudWatch, DevOps, Service Decoupling, Performance Optimization.*