Capstone Synopsis <November 2018> <Group 4>

# Version History:

|  |  |  |  |
| --- | --- | --- | --- |
| ***Version*** | ***Date*** | ***Author*** | ***Key Changes*** |
| *1.0* | *23 mar 2019* | *Rummy Maini* | *First version* |
|  |  |  |  |

# Document References:

|  |  |  |
| --- | --- | --- |
| ***#*** | ***Title*** | ***Link*** |
|  |  |  |
|  |  |  |

# Name / Title of the Project:

Monolith Application to Microservices – Ecommerce platform Domain(AWS)

# Objective / Problem Statement:

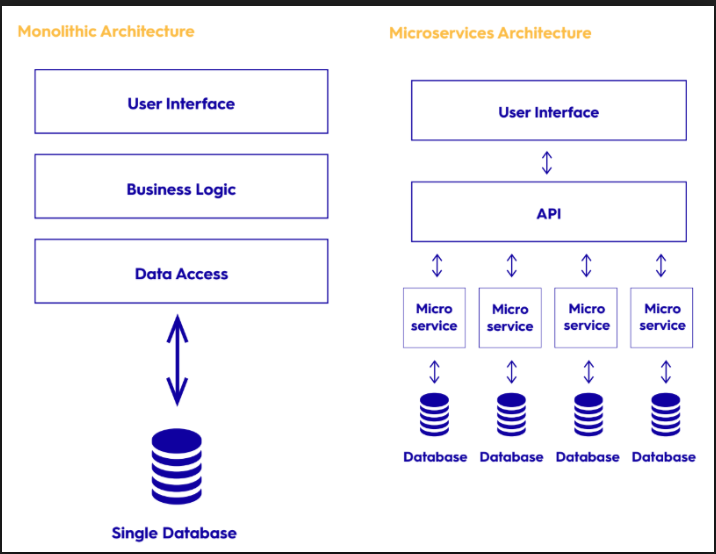
*The idea is to move from Monolith application to Microservices using AWS services.*

*With the traditional monolith architectures it becomes hard to scale and also introducing new technologies /frameworks/languages becomes a constraint resulting in limitation of innovations and ideas. Along with that Application downtime and data security are also two major concerns with the monolithic architecture as it’s results in loss of business credibility and value.*

# Proposed Solution:

* *With the introduction of Microservices each business functionality gets segregated and communicates with others through API.*
* *Microservices are built around business capabilities and thus helps in doing specific task separately and doing them effectively.*
* *This helps in maintaining all the services independently without any deployment dependencies.*
* *Microservices can be written using different frameworks and programming languages.*

We will be implementing the above solution through an Ecommerce website transition from a monolith to microservices application. However the approach is generic and could be applied to any other platforms as well.



MONOLITH MICROSERVICES

User Management

User Management

Search Engine

Search Engine

Product Management

Product Management

# Role of Cloud:

*List out Cloud Services you are planning to use to solve this use case, along with a brief description for selecting that service. Map these services to the Logical Architecture / Flowchart shared in the previous section.*

|  |  |  |  |
| --- | --- | --- | --- |
| *Seq No* | *Cloud Service* | *Reason for selection* | *Business Mapping* |
| *1* | *Amazon EC2* | *For deploying the application* | *Scaling* |
| *2* | *Elastic Container Service* | *Bundling the application* | *Scaling* |
| *3* | *Elastic Container Registry* | *Tagging the bundled application* | *Scaling* |
| *4* | *VPC* | *Having own virtual private cloud network* | *Security* |
| *5* | *CloudFormation* | *Faster Infrastructure onboarding* | *Management* |
| *6* | *Elastic Load Balancing* | *Balancing the application load* | *Networking* |
| *7* | *Route 53* | *DNS Web service* | *Availability* |
| *8* | *IAM* | *Authentication of the users* | *Security* |
| *9* | *RDS* | *Storing the data in RDBMS* | *Storage* |
| *10* | *Elastic Cache* | *Faster retrieval of data* | *Business turnaround* |
| *11* | *WAF* | *Restricting the malfunction access* | *Security* |
| *12* | *Cloud Watch* | *Logging* | *Business process flow* |
| *13* | *Simple Storage Service* | *Storage for website static content* | *Availability* |
| *14* | *API Gateway* | *Triggering Interface to application* | *Security/Reliability* |
| *15* | *Lambda* | *Serverless architecture* | *Scaling* |
| *16* | *Kinesis* | *Analytics* | *Business expansion* |
| *17* | *Content Front* | *Quicker availability of the content* | *Availability/response time* |
| *18* | *Simple Queue Service(SQS)* | *Messaging service as part of integration* | *Integration* |

# Assumptions & Risks:

*List out any assumptions considered for the project or any risks which you foresee along with the mitigation plan.*

1. *Compliance /Regulations-: As part of business expansion to multiple Geographies adhering to the data security regulation could be a risk. Mitigation plan will be getting the business needs and requirements clearly documented.*
2. *Resource Availability - Due to unforeseen scenarios we could have resource issue. Mitigation plan will better distribution of work and regular updates.*

Effort & Cost Analysis:

*Resources-: Team of 4 members*

*Cost-: AWS Services (Free tier) and outside the usage limit additional cost*

*Effort-: 120 hrs*

*Timelines – 1 month*

# Conclusion:

*The synopsis should conclude with identifying the following points:*

1. ***Real world impact*** *– Cloud Migration /Onboarding without any Downtime*
2. ***Any innovation*** *– Lambda Serverless architecture/12 factor applied as Microservices*
3. ***Main achievements*** *– Microservices Implementation /Serverless architecture/ Decoupling of services*

# Team Roles and Responsibilities:

*Clearly mention team’s role & responsibilities in the below format.*

|  |  |  |
| --- | --- | --- |
| *Team Member Name* | *Contribution Till Date* | *Contribution Planned Ahead* |
| *<Name 1>* |  |  |
| *<Name 2>* |  |  |
| *<Name 3>* |  |  |