# SECTION 1

**ALGORITHM TEST**

Given a string *str* consisting of letters only and an integer *n*, the task is to replace every character of the given string by a character which is n times more than it. If the letter exceeds ‘z’, then start checking from ‘a’ in a cyclic manner.

Examples:
*Input: str = “abc”, n = 2*
*Output: cde*
*a is moved by 2 times which results in character c*
*b is moved by 2 times which results in character d*
*c is moved by 2 times which results in character e*

*Input: str = “abc”, n= 28*
*Output: cde*
*a is moved 25 times, z is reached. Then the 26th character will be a, 27th b and 28th c.*
*b is moved 24 times, z is reached. 28-th is b.*
*c is moved 23 times, z is reached. 28-th is e.*

***Question:***

a/ Write an algorithm to solve the above issue. Please consider the complexity of the algorithm.
b/ What is the disadvantage of using the ASCII value of the letters to solve this problem?

**SOLUTION**

```
function charShift(str, n) {
 let res = "";
 n = n % 26;
 for (let i = 0, j = str.length; i < j; i++) {
   let nextCode = str.charCodeAt(i) + n;
   if (nextCode > 122) {
     nextCode = 96 + (nextCode - 123 + 1);
   }
   res += String.fromCharCode(nextCode);
 }
 console.log(res);
 return res
}
charShift('abc', 28)
```

Time Complexity: O(n)
Space Complexity: O(n)

**Problems with using the ASCII value of letters**

ASCII code includes non-printable characters, which prints ' ', use decimal or hexadecimal formats to avoid losing information on the display. Also using ASCII value of letters do not cater for letter cases

# SECTION 2

**AWS Questions**

1. **What is the relationship between an instance and AMI?**
ANSWER
An AMI is a template utilized to make an EC2 instance on AWS based on necessities. It is essentially the OS and pre-installed softwares to dispatch with the instance. AMI provides the data required to launch an instance. AMIs varies to serve different use cases

2. **What does an AMI include?**
ANSWER
AMI includes

- Software Information
- Operating System Information
- Volume Information
- Access Permissions

3. **What is the difference between Amazon S3 and EC2?**
ANSWER

In loose terms, EC2 is a server/machine while S3 a storage container. EC2 is a hosted cloud-based machine meanwhile S3 is a storage service. EC2 is majorly used for hosting applications over cloud. within the case of S3 it's highly useful for storing data and managing it in an exceedingly systematic way for the applications hosted on AWS.

4. **How many buckets can you create in AWS by default?**
ANSWER

By default, you can create up to 100 buckets in each AWS account. You can increase your account bucket limit to a maximum of 1,000 buckets by submitting a service limit increase.

5. **What is Buffer in Amazon web services?**
ANSWER
Buffer is used to synchronize different components of an application.It manages the balance between various components. This usually is applied in maintaining the speed and providing a faster service.

6. **What are the storage classes available in Amazon S3?**

ANSWER

- S3 Standard
- S3 Standard-IA
- S3 Intelligent-Tiering
- S3 One Zone-IA
- S3 Glacier
- Glacier Deep Archive
- RRS

7. **What is the importance of buffers in Amazon Web Services?**

ANSWER

- Providing faster service
- It ensures efficiency over traffic or load.

8. **When should you use the classic load balancer and the application load balancer?**

ANSWER

Classic Load Balancer is intended for applications that were built within the EC2-Classic network, It operates Layer 4 (The Transport Layer) and is used when using Virtual Private Cloud. For TCP/SSL or EC2-Classic.

Application Load Balancer operates at the request level routing traffic to targets – EC2 instances, containers, IP addresses and Lambda functions based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications. Application Load Balancer simplifies and improves the security of your application, by ensuring that the latest SSL/TLS ciphers and protocols are used at all times

9. **Can you change the instance type of the instances that are running in your application tier and are also using autoscaling? If yes, then how?**

ANSWER

AWS does not allow the edit of a launch configuration so any modification will require creating a new one. But if one needs to modify an instance running on autoscaling, You will have to create a new launch configuration with the new instance type and update the ASG with the new launch configuration. When you apply the new launch configuration to the ASG, any new instance created will have the new instance type. However the old instances will still have the old instance type so you'll have to manually terminate them slowly if you want to refresh the whole ASG with the new instance type.

10. **Any advantages and disadvantages when do autoscaling on running instance especially**
**on production instances?**
ANSWER

Auto-Scaling assists to choose the correct amount of instances to adjust in with the workload of an application. Autoscaling saves costs as the servers are added on demand, also it helps in reducing down time especially during peak periods.

On the downside, Auto-Scaling makes it much more complicated to operate servers, issues like how to update code and configuration when the number of servers is always changing, Admin tools become useless, sometimes you can not even login to the server because it is getting scaled down.

11. **What do you do to secure your data in Cloud?**
ANSWER

- On a basic level, the use of IAM to manage users, security credentials, access keys and permission policies to limit access to the application.
- Use virtual firewalls on AWS marketplace such as AWS Shield, Guard Duty, and Cloud Watch on all virtual networks to control and monitor network traffic to secure your infrastructure and the operating system it is running on.
- Use of native AWS security tools such as AWS Shield, Guard Duty, and Cloud Watch readily available that can help secure cloud environments.
- Having a strong password policy in place is critical to the security as it can significantly reduce the chances of a security breach.
- Use AWS encryption with EBS, RDS, and S3 or Azure Secure Server Encryption (SSW) with files and blobs. Furthermore ensure that data stored on S3 via SSL has encrypted endpoints to protect data in transit as well.

12. **Explain Stopping, Starting, and Terminating an Amazon EC2 instance.**
ANSWER

- Starting - it enters the pending state, and then to the running state. In this process we move the instance to a new host computer (though in some cases, it remains on the current host). When you stop and start your instance, you lose any data on the instance store volumes on the previous host computer.

- Stopping - Temporarily shutting down your instance, the instance will be shutdown and the virtual machine that was provisioned for you will be permanently taken away and you will no longer be charged for instance usage. The key difference between stopping and terminating an instance is that the attached bootable EBS volume will not be deleted. The data on your EBS volume will remain after stopping while all information on the local (ephemeral) hard drive will be lost as usual. The volume will continue to persist in its availability zone. The instance can be started at any time.

- Terminating - This action is returning control back to amazon. It destroys the softwares and the AMI created. The instance will be shut down and the virtual machine that was

provisioned will be permanently taken away and you will no longer be charged for instance usage. Any data that was stored locally on the instance will be lost. Any attached EBS volumes will be detached and deleted.

13. **Define regions and availability zones in Amazon EC2**
ANSWER
Regions in Amazon E2 are geographic locations where amazon has servers/huge data centers, and these are possible locations where EC2 instances will be set up on.

Availability zones are isolated locations present within a particular region. Regions typically have more than one availability zone. For data distribution and backup

14. **Is it important to select suitable zones for AWS services? What's the benefits of it?**
ANSWER
Yes it is, Selecting the suitable zones offers the choice of operating from a trusted data center for the scale of application to be deployed, it gives the ability to operate production applications and databases that are more highly available, more fault tolerant. Also launching an instance in a location close to users will improve speed of access of the platform

15. **What are the challenges in microservices debugging and troubleshooting?**
ANSWER
Microservices debugging requires highly organized and appropriately described logs; this is probably one of the biggest problems one might face deploying a microservice architecture application. - not being able to read meaning data from logs. This can make debugging extremely difficult
Monitoring microservice systems can be difficult and would involve a comprehensive APM set up.
Any time there is an update, things could malfunction, and they can break in potentially unpredictable ways due to dependencies mismatch

# SECTION 3 - Architecture (Laravel PHP)

Implementation of the app that provides a service - for example a food ordering application.
Most of the functionalities to be used are provided by the laravel frame work like

- queues,
- creating a job and configuring it,
- mailing,
- migrations for database
- following an MVC pattern,
- outsourcing the view from blade to using VueJs an excellent front end framework

A detailed outline of the proposed framework for development is attached alongside this document named - ‘Service App Flow.pdf’

The document includes a general process flow, an architecture, the database design and a modular process flow to explain the proposed solution. The architecture was designed with industry standards and best practices - SOLID, ACID, DRY and KISS. ensuring an efficient system that is scalable and maintainable.

DEPLOYING THE APPLICATION

AWS will be ideal for deploying the solution: Below are the process outlined:

- Setting a Linux 20.04 AMI on an EC2 instance.
- Configuring the EC2 instance by installing appropriate php version, composer and the latest version of laravel on the server.
- After developing and testing the app locally,
- Create a repository that holds the application source file
- Upload files to EC2 instance, set-up database connection to the DB
- Install nginx to configure as the server.
- Set up a domain name pointing AWS name servers, and configure necessary port to load the app on the base route
- On the EC2 instance, using AWS CodePipeline, set up a CI/CD pipeline that builds, tests, and deploys the app once there are changes, connect the pipeline to the repository for automated deploy and update on pushing updates to the repository.
- Usually the data can be stored on the EC2 instance, but if other reasons an external relational database is one setting one up on aws would be necessary.
- Set up other services like S3 bucket and security services.
- Set up SQS for queue and SNS for notification
- Set up APMs(datadog) for monitoring and analytics. This will be connected to a service that will ensure constant notification - A slack channel for instance.

SERVICES
AWS - EC2-(Linux AMI), SQS, SNS, DATADOG, CODEPIPELINE, CODEDEPLOY
PHP, MySQL