What is serverless?

* Function as service - FaaS
* Serverless computing doesn’t mean there aren’t any servers; there will be a cloud provider allocating the machine resources, but the developer doesn’t have to worry about server management. He just has to focus on building the best web applications.

Faas Provider

* AWS lambda
* Google cloud functions
* Microsoft Azure functions

Advantage

* No server management
* only charged for the server space they use, reducing cost
* Serverless architectures are inherently scalable
* Quick deployments and updates are possible

Disadvantage

* Testing and debugging become more challenging
* Serverless architectures are not built for long-running processes- some application running code continuously, so it may generate high cost compare to traditional server concept.
* Performance may be affected - he serverless provider will keep it ready to be activated – a request for this ready-to-go code is called a 'warm start.' A request for code that hasn't been used in a while is called a 'cold start.'

34. Mention the time span in which the AWS Lambda function will execute.

**Answer:** All the process of AWS Lambda and execution takes place within 300 seconds from placing calls to AWS Lambda. The default timeout is 3 seconds rest you can setup any value between 1 to 300 seconds.

37. What are the Final variables?

**Answer:** Once assigned these variables cannot be changed. In its earlier stage, they are known as effective variables where any form of change is possible and the values are assigned to them. They also play an important role in testing. Most of the local expressions are final.

38. How can performance be improved in Lambda?

**Answer:** There are some of the methods by which performance can be improved in Lambda. You can improve the performance by using Linux software RAID and with the help of RAID, we can be assured of better security.

Take care points for Lambda :

- even though Lambda can scale itself up to the account limit of 1000 instances per region, we have possibility to control this with upper limit

- it is good to set timeout on Lambda function to kill tasks running longer than expected

- if Lambda is triggered from SQS we can use `batchSize` to limit number of messages processed in single Lambda invocation

Serverless npm

npm install -g serverless

*# Create a new Serverless Service/Project*

serverless create --template aws-nodejs --path my-service

#you simply want to deploy all changes within your Service at the same time.

serverless deploy -v

#Deploy particular function

serverless deploy function -f hello

# invoke the function

serverless invoke -f hello

$ remove all function

serverless remove

npm install serverless-http

npm install serverless-domain-manager --save-dev