Top AWS SageMaker Interview Questions and Answers (2022)

In this post, questions from AWS SageMaker Interviews will be answered for Experienced and Freshers. We're trying to share our experience and learn how to help you make progress in your career.

1. What is AWS SageMaker?
2. What can SageMaker do?
3. What are the features of Amazon SageMaker?
4. How does AWS SageMaker work?
5. Is AWS SageMaker free?
6. How to call a python file from angular application in AWS Sagemaker?
7. How to utilize ARIMA model in AWS Sagemaker?
8. How to analyze data and evaluate machine learning models on Amazon SageMaker?

Q: What is AWS SageMaker?
Ans:

Amazon SageMaker was launched in November 2017 and is a cloud machine learning platform. SageMaker helps developers to build, train and deploy cloud-based machine learning models. SageMaker allows developers to deploy ML models in embedded systems and edge devices.

Q: What can SageMaker do?
Ans:

ploy ML models. SageMaker eliminates the heavy lifting in each step of the process to make the development of high-quality models simpler.

Q: What are the features of Amazon SageMaker?
Ans:

Amazon SageMaker includes the following features, which comes in the Prepare, Build, Train-Tune and Deploy-Manage Processes:

1. **SageMaker Studio**

   An integrated machine learning environment that allows you to build, train, develop and monitor your models in the same application.

2. **SageMaker Model Registry**

   It helps in versioning, artifact and lineage tracking, approval workflow, and cross account support for deployment of your machine learning models.

3. **SageMaker Projects**

   Use SageMaker projects to build end-to-end ML solutions with CI/CD.

4. **SageMaker Model Building Pipelines**

   Create and maintain machine learning pipelines incorporated directly with SageMaker jobs.

5. **SageMaker ML Lineage Tracking**

machine learning workflows for easy and streamlined data preprocessing and feature engineering with little to no code. To configure your data prep workflow, you can also add your own Python scripts and transformations.

7. **SageMaker Feature Store**

   A centralised store for features and associated metadata, which makes it easy to recognise and reuse features. Two stores, one online or one offline, can be created. The Online Store is for low latency, real-time inference applications, and the Offline Store can be used for training and batch inference.

8. **SageMaker JumpStart**

   Learn about SageMaker features and capabilities through curated 1-click solutions, example notebooks, and pretrained models that you can deploy. You can also fine-tune the models and deploy them.

9. **SageMaker Clarify**

   It help explain the predictions, detecting potential bias that models make.

10. **SageMaker Edge Manager**

    It help in optimizing custom models for edge devices, create and manage fleets and run models efficiently.

11. **SageMaker Ground Truth**

    High-quality training datasets by using workers along with machine learning to create labeled datasets.

12. **Amazon Augmented AI**

    Build the workflows required for human review of ML predictions. Amazon A2I brings human review to all developers, removing the undifferentiated heavy lifting associated with building human review systems or managing large numbers of human reviewers.

13. **SageMaker Studio Notebooks**

    The next generation of SageMaker notebooks that include AWS Single Sign-On (AWS SSO) integration, fast start-up

build on experiments conducted by peers, and trace model lineage for compliance and audit verifications.

15. **SageMaker Debugger**

    Inspect training parameters and data throughout the training process. Automatically detect and alert users to commonly occurring errors such as parameter values getting too large or small.

16. **SageMaker Autopilot**

    Users without machine learning knowledge can quickly build classification and regression models.

17. **SageMaker Model Monitor**

    Monitor and analyze models in production (endpoints) to detect data drift and deviations in model quality.

18. **SageMaker Neo**

    Train machine learning models once, then run anywhere in the cloud and at the edge.

19. **SageMaker Elastic Inference**

    Speed up the throughput and decrease the latency of getting real-time inferences.

20. **Reinforcement Learning**

    Maximize the long-term reward that an agent receives as a result of its actions.

21. **Preprocessing**

    Analyze and preprocess data, tackle feature engineering, and evaluate models.

22. **Batch Transform**

    Preprocess datasets, run inference when you don't need a persistent endpoint, and associate input records with inferences to help the interpretation of results.

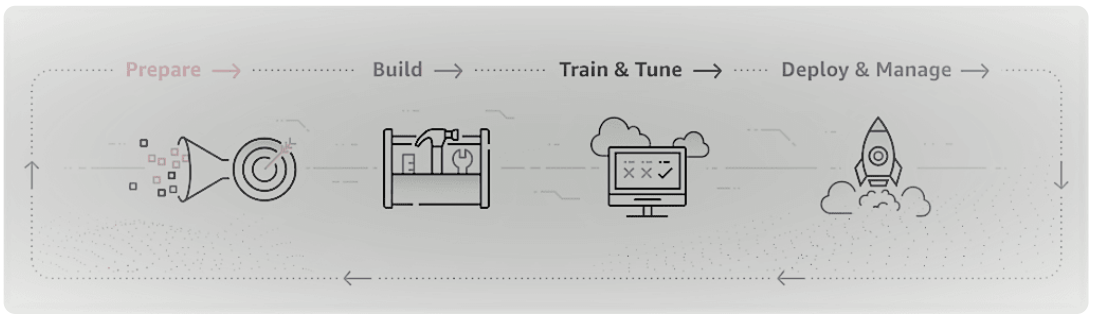
Take a look at our suggested post on AWS :

AWS EC2 Interview Questions

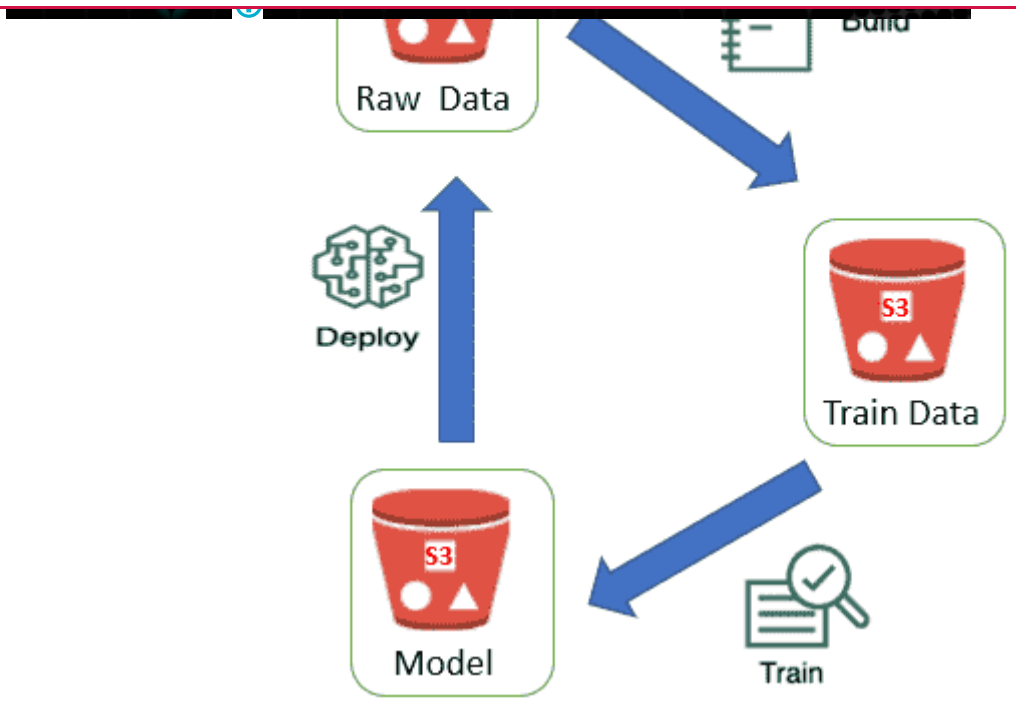
AWS VPC Interview Questions

Q: How does AWS SageMaker work?
Ans:

SageMaker has a three-step process whcih simplifies machine learning modelling. Build, train and tune, deploy models automatically



Amazon SageMaker Autopilot chooses the best prediction algorithm and automatically builds, trains and tunes machine learning models without lack of visibility or control.



1. **Build**

   Jupyter notebook instance can be created with just a few clicks with desirable server size and capacity. The data cleaning and exploration can begin when the Jupyter hub is running. The key feature is for our notebook instance to pick the desired server size. After some periods of inactivity, we can automate the instance shutdown and avoid unnecessary costs.

2. **Train**

   With the ability to choose the size and number of servers we can train our models at the right server capacity. Starting a server is just a line of code and after a model is completed, the server automatically shuts down.

3. **Deploy**

   Again, by defining desired server capacity, we are able to deploy the machine-learning model with only one line of code. To create the application service or serverless function use the endpoint address.

Q: Is AWS SageMaker free?
Ans:

Using SageMaker Studio is free, you only need to pay for the AWS services that you use within Studio. You can make use of many services within SageMaker Studio at no additional charge, including: SageMaker Pipelines to automate and manage auto-mated ML workflows.

Q: How to call a python file from angular application in AWS Sagemaker?
Ans:

We can generate a lambda function to be invoked by the angular app utilising the API gateway. In return that lambda will call the our Sagemaker function.

Can refer AWS documentation
https://aws.amazon.com/blogs/machine-learning/call-an-amazon-sagemaker-model-endpoint-using-amazon-api-gateway-and-aws-lambda/

Q: How to utilize ARIMA model in AWS Sagemaker?
Ans:

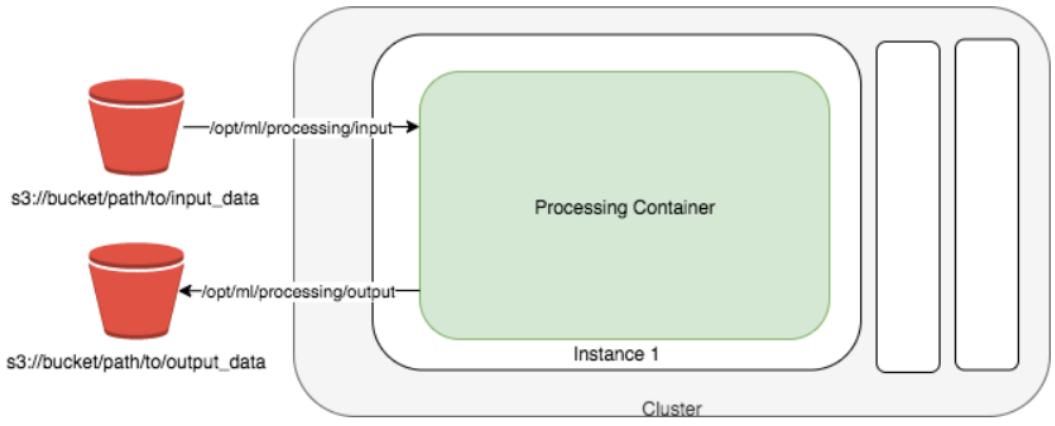
1. Amazon Forecast that have ARIMA built in can be used
2. Or we can create our own Docker container to publish it to ECR, and then can utilize the same in case of AWS Sagemaker.
   Can refer AWS documentation https://sagemaker-examples.readthedocs.io/en/latest/advanced_functionality/scikit_bring_your_own/scikit_bring_your_own.html#The-example

Q: How to analyze data and evaluate machine learning models on Amazon SageMaker?
Ans:

*We can **Amazon SageMaker Processing**, to analyze data and evaluate machine learning models on Amazon SageMaker.*

With Processing, we can perform data processing workloads such as feature engineering, data validation, model evaluation, and model interpretation on SageMaker in a simple, managed

tation phase and after the code is put in production.



The diagram above depicts how Amazon SageMaker begins a Processing job. Amazon SageMaker reads your script, copies your data from Amazon Simple Storage Service (Amazon S3), and then retrieves a processing container. The processing container image can be either an Amazon SageMaker built-in image or a custom image provided by you. Amazon SageMaker fully manages the underlying infrastructure for a Processing job. Cluster resources are provisioned for the duration of your job and then cleaned up when it is over. The Processing job's output is saved in the Amazon S3 bucket you specified.