✦ Member-only story

Top 10 scenario-based interview questions with answer related to AWS Part-5

Deep · Follow

6 min read · Dec 28, 2023

▷ Listen   Share   ••• More

It took me a lot of time and effort to provide these detailed answers, and Medium doesn't pay for technical articles like this. If you find these notes helpful, **please support me**!

👉 Click here to **buy me a coffee**, and don't forget to **follow me on Medium!**

On a free Medium plan? **Click here** & **Please Do Follow Me!**

```
I would really appreciate it
if you could please stay on the page
for at least 30 seconds to help me support
my work as per Medium's new changes.
```

Scenario 1: Question: You are responsible for optimizing costs in an AWS environment. How would you go about identifying and reducing unnecessary expenses?

Answer: To optimize costs in AWS, I would start by utilizing AWS Cost Explorer to analyze the cost and usage data. I would identify underutilized or idle resources and either modify, resize, or terminate them. Additionally, I would implement AWS Budgets and set up alerts to receive notifications when costs exceed predefined thresholds. Implementing auto-scaling for resources based on demand can also help optimize costs by automatically adjusting capacity.

Scenario 2: Question: Your company is planning to migrate its on-premises database to Amazon RDS. Outline the key steps you would take to ensure a successful database migration.

Answer: The key steps for a successful database migration to Amazon RDS include:

1. Assessment: Understand the existing database schema, dependencies, and performance metrics.
2. Schema Conversion: Modify the schema to be compatible with the target RDS engine.
3. Data Migration: Use AWS Database Migration Service (DMS) to migrate data with minimal downtime.
4. Testing: Conduct thorough testing to ensure data integrity, performance, and functionality.
5. Cutover: Plan and execute the cutover to switch production traffic to the RDS instance.
6. Post-Migration Validation: Perform post-migration checks to ensure everything is functioning as expected.

Scenario 3: Question: Your application, hosted on AWS, is experiencing a sudden increase in traffic. How would you handle this surge in demand and ensure that the application remains responsive and available?

Answer: To handle a sudden increase in traffic, I would:

1. Auto-scaling: Configure auto-scaling groups to automatically adjust the number of EC2 instances based on traffic.
2. Content Delivery Network (CDN): Utilize AWS CloudFront to cache and deliver content closer to end-users, reducing the load on the origin server.
3. Load Balancing: Implement an Elastic Load Balancer (ELB) to distribute incoming traffic across multiple instances.
4. Caching: Implement caching mechanisms, such as Amazon ElastiCache, to reduce the load on the database and improve response times.

5. Monitoring: Set up CloudWatch alarms to receive notifications and automatically trigger scaling actions based on predefined thresholds.

Scenario 4: Question: Your team is working on a microservices architecture, and you need to implement communication between microservices. How would you approach this using AWS services?

Answer: I would implement communication between microservices using the following AWS services:

1. Amazon API Gateway: Use API Gateway to create RESTful APIs that expose microservices to clients.

2. AWS Lambda: Deploy serverless functions for business logic, allowing for scalable and cost-effective execution.

3. Amazon SNS (Simple Notification Service): Implement SNS for event-driven communication between microservices.

4. Amazon SQS (Simple Queue Service): Use SQS to decouple and manage messages between microservices.

5. Amazon EventBridge: Utilize EventBridge for event-driven architectures, enabling communication through events and rules.

Scenario 5: Question: Your application requires a highly available and fault-tolerant storage solution. How would you design a scalable storage architecture using AWS services?

Answer: For a highly available and fault-tolerant storage architecture, I would:

1. Amazon S3 (Simple Storage Service): Use S3 for durable, scalable, and low-latency object storage.

2. Cross-Region Replication: Implement cross-region replication for redundancy and disaster recovery.

3. Versioning: Enable versioning in S3 to track and recover previous versions of objects.

4. Amazon Glacier: Utilize Glacier for long-term archival of infrequently accessed data.

5. Amazon EFS (Elastic File System): Implement EFS for scalable and shared file storage across multiple instances.

Scenario 6: Question: Your team is planning to deploy a highly available web application on AWS. How would you design the architecture to ensure resilience and minimize downtime?

Answer: To design a highly available web application on AWS, I would:

number of EC2 instances based on demand, ensuring availability.

3. Multi-AZ Deployments: Deploy resources across multiple Availability Zones to ensure resilience against failures in a single zone.

4. Elastic Load Balancer (ELB): Use ELB to distribute incoming traffic across multiple instances, improving fault tolerance.

5. Amazon RDS Multi-AZ: Deploy the database using Multi-AZ configuration for automatic failover and increased availability.

Scenario 7: Question: Your organization is looking to enhance security by implementing encryption for data at rest. How would you implement encryption in AWS?

Answer: To implement encryption for data at rest in AWS:

1. Amazon S3 Server-Side Encryption: Enable server-side encryption for S3 buckets using AWS Key Management Service (KMS) or Amazon S3-managed keys (SSE-S3).

2. Amazon EBS Volume Encryption: Encrypt Amazon EBS volumes attached to EC2 instances using AWS KMS.

3. Amazon RDS Encryption: Enable encryption for RDS databases using AWS KMS for enhanced security.

4. AWS KMS Customer Master Keys: Manage and rotate Customer Master Keys (CMKs) in AWS KMS regularly for improved key security.

5. AWS CloudHSM: For additional security, consider using AWS CloudHSM for dedicated hardware security module (HSM) protection.

Scenario 8: Question: Your application processes sensitive data, and compliance is a top priority. How would you ensure that your AWS environment meets compliance requirements?

Answer: To ensure compliance in an AWS environment:

1. AWS Artifact: Leverage AWS Artifact to access compliance reports and documentation to support audit requirements.
2. AWS Config: Implement AWS Config to assess, audit, and evaluate the configurations of AWS resources against predefined rules.
3. AWS CloudTrail: Enable CloudTrail to log and monitor all API calls, providing visibility into user activity.
4. AWS Identity and Access Management (IAM): Define and enforce strict IAM policies to control access to resources based on the principle of least privilege.
5. Encryption and Security Best Practices: Adhere to encryption standards, implement network security, and follow other security best practices outlined in AWS documentation.

Scenario 9: Question: Your team is considering the adoption of a serverless architecture for specific workloads. What are the advantages and challenges of using AWS Lambda?

Answer: Advantages:

1. Scalability: AWS Lambda automatically scales based on demand, allowing for efficient resource utilization.
2. Cost Savings: Pay only for the compute time consumed, leading to cost-effective solutions for intermittent workloads.
3. Event-Driven: Easily integrate with various AWS services and trigger functions based on events.
4. Managed Service: AWS Lambda is a fully managed service, eliminating the need for server provisioning and maintenance.

Challenges:

1. Cold Start Latency: There may be initial latency (cold start) when a function is invoked after being idle.
2. Execution Time Limits: Functions are limited by maximum execution times, and long-running processes may require a different approach.
3. Limited State: Designed for stateless functions; maintaining state requires additional considerations.

4. Debugging Complexity: Debugging serverless applications can be challenging compared to traditional architectures.

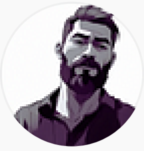
Scenario 10: Question: Your team is planning to deploy a containerized application on AWS. How would you manage container orchestration and scaling?

Answer:

1. Amazon ECS (Elastic Container Service): Use ECS to deploy, manage, and scale containers with ease. ECS provides a fully managed container orchestration service.
2. Amazon EKS (Elastic Kubernetes Service): Consider EKS for Kubernetes-based container orchestration, offering scalability and flexibility.
3. Auto Scaling Groups: Implement Auto Scaling Groups to automatically adjust the number of EC2 instances in the container cluster based on demand.
4. Amazon EC2 Spot Instances: Use Spot Instances to reduce costs for non-critical workloads, taking advantage of spare EC2 capacity.
5. Application Load Balancer: Deploy an Application Load Balancer to distribute incoming traffic across containers, ensuring optimal load balancing.

> ***P*lease consider giving it a clap and follow to help me reach more like-minded individuals. Thank you for being a part of this community!**

Interview Jobs AWS DevOps Sre



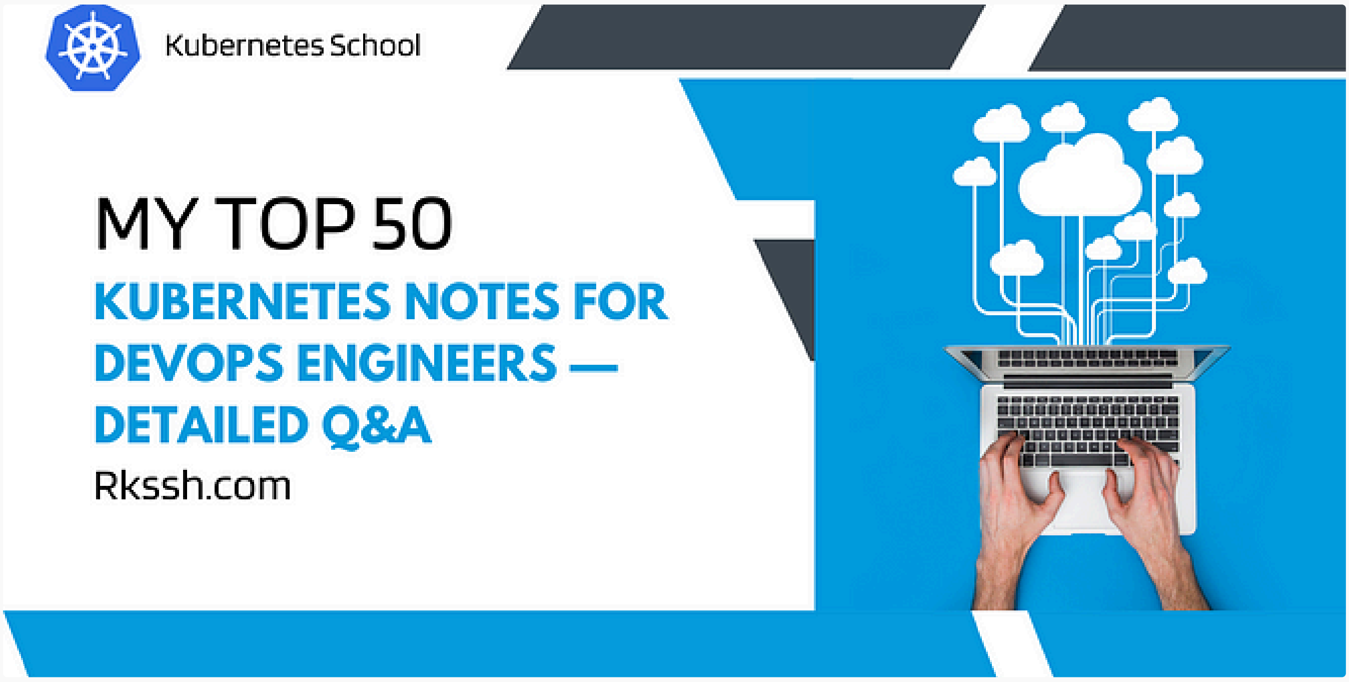
Follow

Written by Deep

1K Followers

Cloud / DevOps Engineer | I write on Cloud, AWS, K8s, DevOps and CI/CD Buy-Me-A-Coffee https://buymeacoffee.com/deepdevops

More from Deep



Deep in CodeX

My Top 50 Kubernetes Notes for DevOps Engineers—Detailed Q&A

My Notes Series Kubernetes

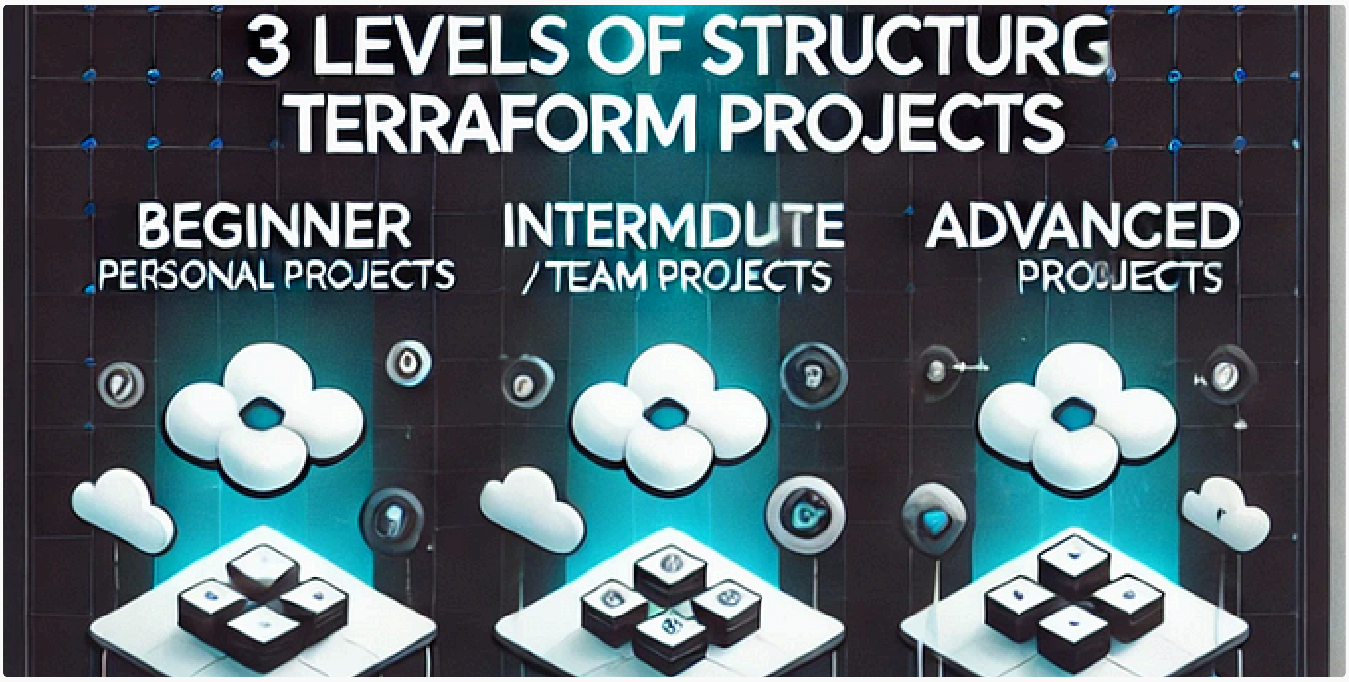
✦ Aug 9 477 2

Deep

My Top 50 Terraform Notes for DevOps Engineers — Detailed Q&A

On a free Medium plan? Click here

Aug 28 162



Deep in DevOps.dev

3 Levels of structure terraform projects

The Evolution of a Terraform Project - From Tiny Side Project to Big-Time Enterprise

Aug 13 70 4

Deep in CodeX

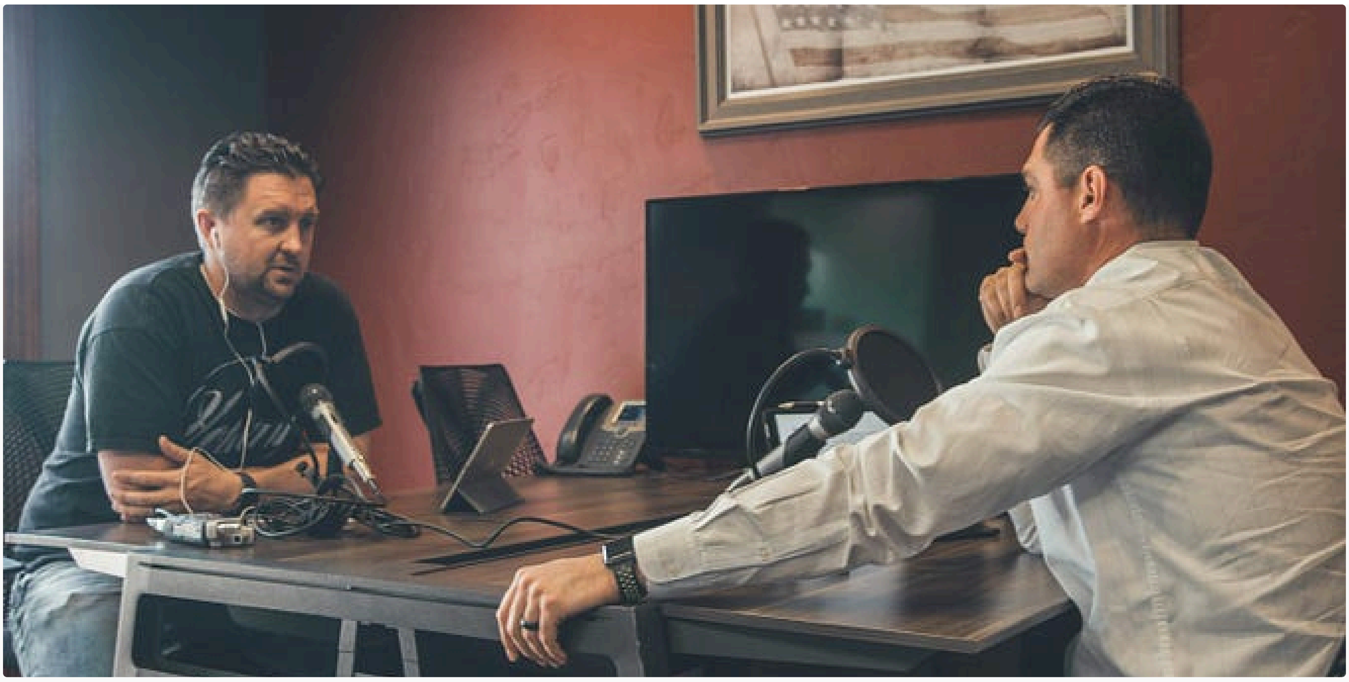
The Toughest Kubernetes Interview Questions I Faced — You Won't Believe

From managing clusters to scaling nightmares, here are the trickiest Kubernetes questions that left me stumped during my interviews.

Sep 4 22 1

See all from Deep

Recommended from Medium

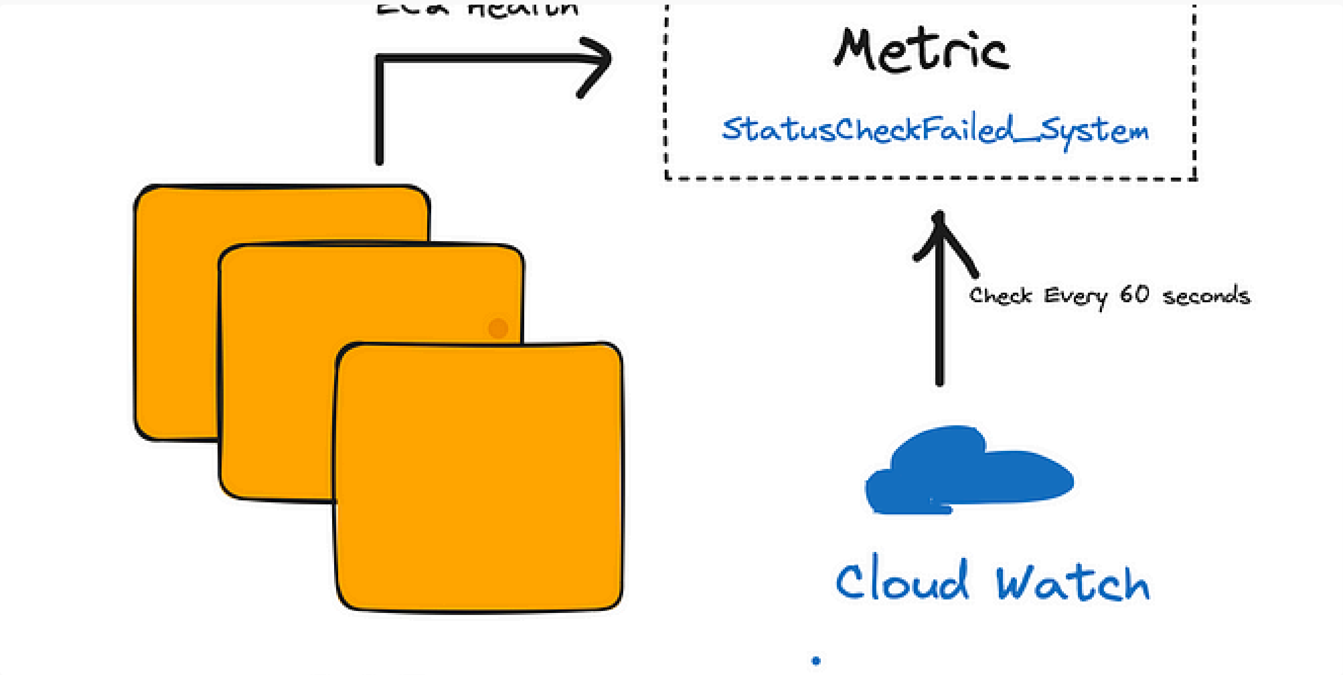


Nurunnubi Talukder in Cloud, DevOps, Security & AI Career Talk

Solution Architect AWS interview questions and answers[25Q-Part1]!!

Aspiring to become a Solution Architect on AWS? This guide covers essential interview questions and answers to help you prepare. Whether...

✦ Jun 30 👏 43



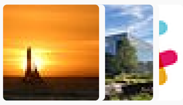
Vikas Taank

Orchestrating Cloud Watch Alarm for EC2

AWS does recover failed virtual machines under only some circumstances. For example, AWS will not recover an EC2 instance if a whole rack...

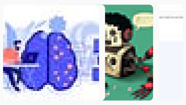
Jun 21 1

Lists



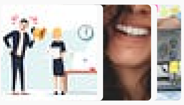
Stories to Help You Level-Up at Work

19 stories · 793 saves



The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 456 saves

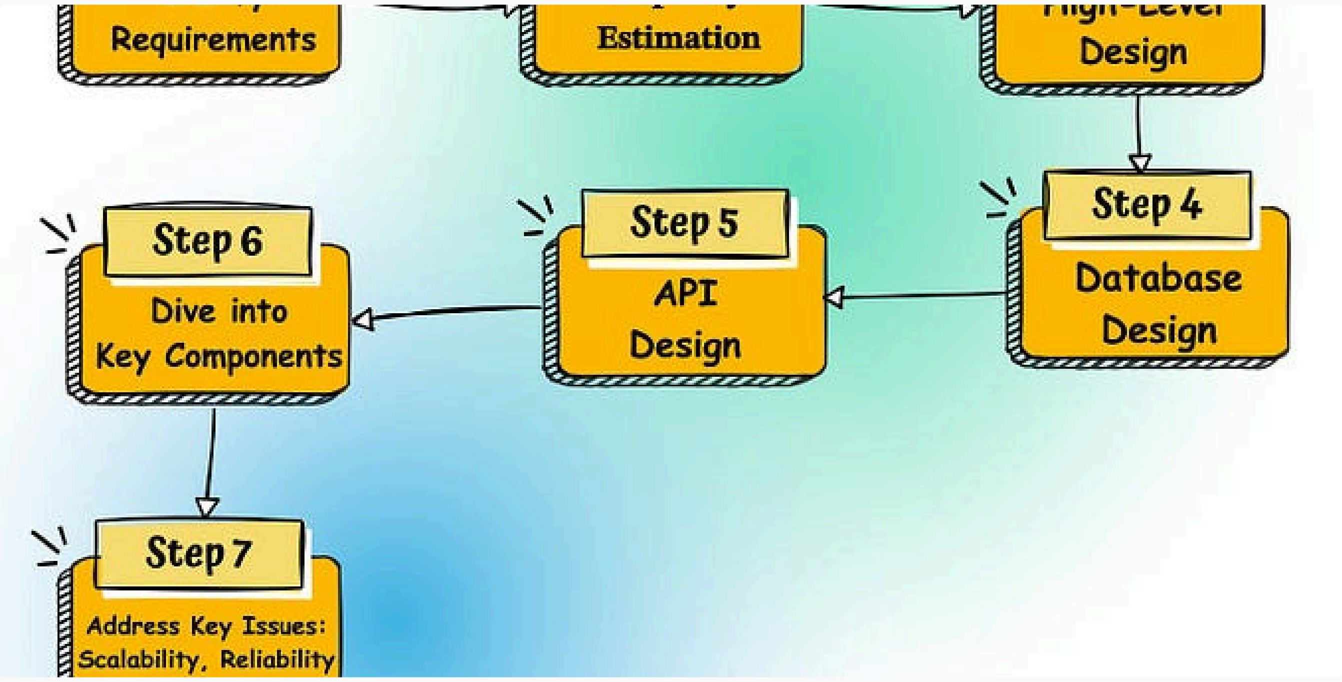


Work 101

26 stories · 167 saves

Stories to Help You Grow as a Software Developer

19 stories · 1349 saves



Ashish Pratap Singh in AlgoMaster.io

A Step-by-Step Guide to System Design Interviews

System design interviews can be one of the most challenging but with a structured approach, you can navigate these interviews more...

Aug 25 123 1

Ashwin

AWS EC2 Instance Storage [Interview Series]

Are you familiar with EC2 instance storage, EBS volumes, and Amazon EFS? If not, don't worry!

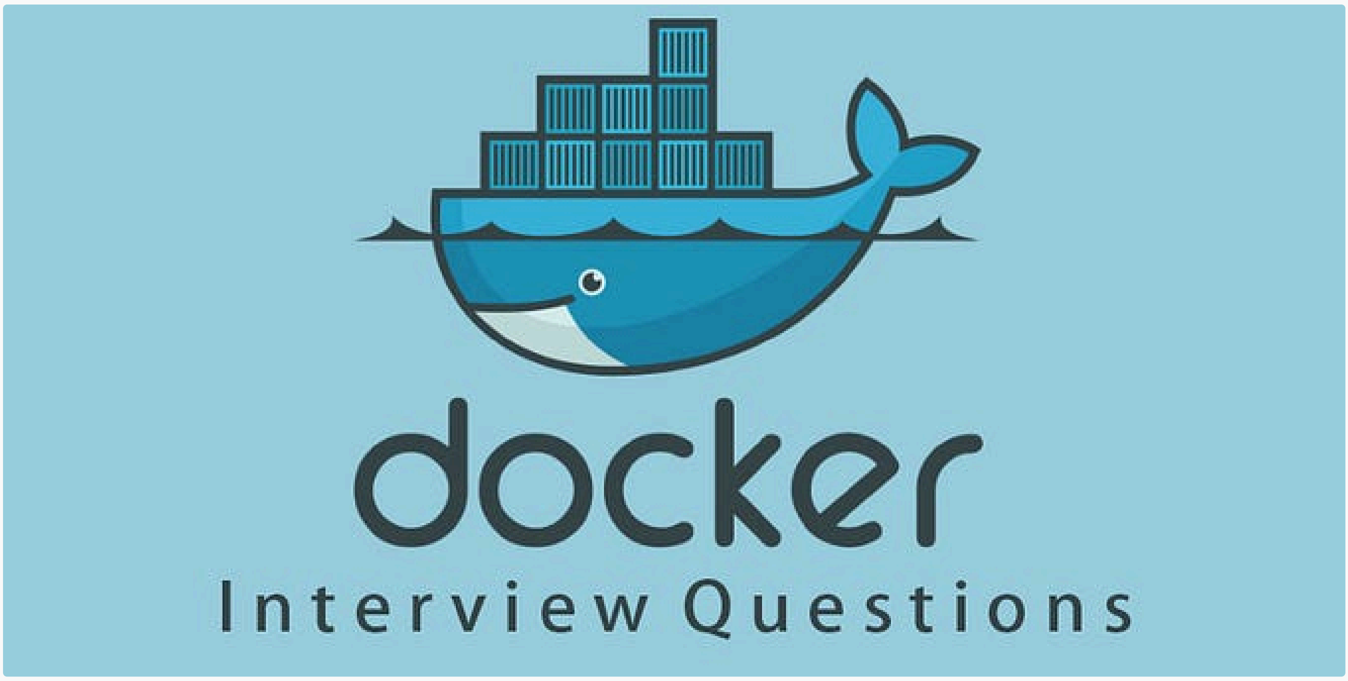
Apr 29 👏 1

Liu Zuo Lin in Level Up Coding

AWS Data Engineer Associate Cheatsheet (Consolidated) + Self-Test

If you're studying for the AWS Data Engineer Associate Certification DEA-C01, hopefully this is helpful in some way. I'm also assuming that…

✦ Apr 2 👏 720 💬 1



Aditya Pathak

Docker Interview Questions Basic to Advanced

Summary about docker.

Mar 14 👏 1

See more recommendations