Fullstack Developer/Devops Engineer Finboa 6 - 10 years 25-35 Lacs P.A. Nagpur Send me jobs like this Posted: 10 days ago Openings: 1 Applicants: 146 Register to apply Login to apply Job description Role & responsibilities We are seeking a talented and motivated Full Stack Developer/DevOps Engineer with strong expertise in both development and cloud infrastructure. The ideal candidate will have a solid understanding of front-end and back-end technologies, hands-on experience with cloud services, and a passion for delivering high-quality solutions in a fast-paced environment. Develop and maintain full-stack applications using C#, .NET Framework, Javascript, and AngularJS. Manage databases with SQL Server and optimize data retrieval and storage processes. Design and implement solutions on Azure Cloud Services (IaaS, PaaS), ensuring scalability, security, and performance. Utilize DevOps practices, including Continuous Integration (CI) and Continuous Delivery (CD) pipelines. Automate workflows and manage infrastructure using PowerShell and container technologies (e.g., Docker and Kubernetes). Conduct thorough code reviews to maintain code quality, ensure best practices, and mentor junior developers. Collaborate with cross-functi onal teams, including product managers, designers, and other developers, to deliver seamless and user-friendly applications. Proficiency in: C# and .NET Framework for backend development. JavaScript and AngularJS for front-end development. SQL Server for database management. Strong experience with: Azure Cloud Services (Infrastructure as a Service and Platform as a Service). DevOps tools and practices, including containers (e.g., Docker/Kubernetes). PowerShell scripting for automation and infrastructure management. Hands-on experience with code reviews and and implementing best practices for high-quality software development. Strong problem-solving skills and ability to work in a dynamic environment. Experience with other cloud platforms (AWS, Google Cloud) is a plus. Knowledge of microservices architecture and serverless computing. Familiarity with other front-end frameworks (React, Vue.js). profile name Fullstack Developer/Devops Engineer tellllme all the possible questions that can be asked for this profile

Which of the following is the correct way to define an asynchronous method in C#?

* A) public async void MyMethod() { /\* method logic \*/ }
* B) public async Task MyMethod() { /\* method logic \*/ }
* C) public async Task MyMethod() => Task.Delay(1000);
* D) public void async MyMethod() { /\* method logic \*/ }

**Answer**: B

In ASP.NET Core, which interface is typically used to define a service that should have a **scoped lifetime**?

* A) ITransient
* B) IServiceCollection
* C) IScoped
* D) None of the above

**Answer**: D  
(Explanation: The correct answer is not to use an interface like IScoped, but to use .AddScoped<TService, TImplementation>() in the ConfigureServices method.)

What will the following code output?

javascript

Copy code

const promise = new Promise((resolve, reject) => {

reject('Error');

});

promise.then(() => console.log('Success')).catch(() => console.log('Failed'));

* A) Success
* B) Failed
* C) Error
* D) Nothing

**Answer**: B

Which of the following LINQ methods is used to select distinct elements from a collection?

* A) Select
* B) Distinct
* C) GroupBy
* D) OrderBy

**Answer**: B

In AngularJS, which directive is used to bind the **value of an HTML element** to application data?

* A) ng-model
* B) ng-bind
* C) ng-controller
* D) ng-init

**Answer**: A

What is the primary use of **delegates** in C#?

* A) To call unmanaged code
* B) To define a reference type that can encapsulate a method
* C) To allow implicit conversion between types
* D) To store multiple values

**Answer**: B

Which of the following statements is **true** about middleware in ASP.NET Core?

* A) Middleware components in ASP.NET Core are executed in a random order.
* B) Middleware components are executed in the order in which they are registered in the pipeline.
* C) Middleware components are only executed when a user is authenticated.
* D) Middleware components must always return an HTML response.

**Answer**: B

What will the following JavaScript code output?

javascript

Copy code

function makeCounter() {

let count = 0;

return function() {

return count++;

}

}

let counter = makeCounter();

console.log(counter());

console.log(counter());

* A) 0 1
* B) 1 1
* C) undefined undefined
* D) 0 0

**Answer**: A

In Entity Framework Core, which method is used to execute **raw SQL queries**?

* A) FromSql
* B) ExecuteSqlCommand
* C) SqlQuery
* D) UseSql

**Answer**: A

Which of the following is **true** about services in AngularJS?

* A) Services in AngularJS are instantiated per controller.
* B) Services in AngularJS are singletons.
* C) Services can only be created using the $provide service.
* D) Services cannot have dependencies.

**Answer**: B

Which of the following is **true** about **Clustered Indexes** in SQL Server?

* A) You can have multiple clustered indexes on a table.
* B) Clustered indexes store data rows in the order of the index.
* C) Clustered indexes only apply to columns with unique values.
* D) Clustered indexes are automatically created on foreign key columns.

**Answer**: B

Which of the following is **true** about **3rd Normal Form (3NF)** in database design?

* A) It ensures that the table contains no transitive dependencies.
* B) It eliminates all duplicate columns from the same table.
* C) It requires that each column in a table must depend on the primary key.
* D) It ensures that no multi-valued attributes exist.

**Answer**: A

Which of the following **isolation levels** in SQL Server allows **dirty reads**?

* A) READ UNCOMMITTED
* B) READ COMMITTED
* C) REPEATABLE READ
* D) SERIALIZABLE

**Answer**: A

Which of the following scenarios is most likely to result in a **deadlock**?

* A) Two sessions concurrently updating two tables in reverse order.
* B) A query performing an INSERT followed by an UPDATE.
* C) A single transaction holding a lock on a table for a long time.
* D) Queries using READ COMMITTED isolation level.

**Answer**: A

Which of the following is **not** a valid **data type** in SQL Server?

* A) VARCHAR(MAX)
* B) UNIQUEIDENTIFIER
* C) BLOB
* D) XML

**Answer**: C

**6. SQL Server - Backup Types**

Which type of backup **backs up only the data that has changed** since the last **full backup**?

* A) Differential Backup
* B) Transaction Log Backup
* C) Incremental Backup
* D) Full Backup
* **Answer**: A

Which of the following **index types** is best suited for **improving performance** of queries that involve searching **textual data**?

* A) Clustered Index
* B) Non-Clustered Index
* C) Full-Text Index
* D) XML Index

**Answer**: C

Which of the following is **true** about **Views** in SQL Server?

* A) A view can contain a WHERE clause but not a JOIN.
* B) A view can include columns from multiple tables.
* C) You can update the underlying tables directly through a view without restrictions.
* D) Views are automatically updated when the underlying data changes.

**Answer**: B

What is the main benefit of using a **Stored Procedure** over an **ad-hoc query**?

* A) It ensures faster query execution in all cases.
* B) It prevents SQL injection attacks.
* C) It allows query results to be cached.
* D) It provides reusable, precompiled SQL logic.

**Answer**: D

Which of the following **commands** is used to **undo** a transaction in SQL Server?

* A) ROLLBACK
* B) COMMIT
* C) UNDO
* D) SAVE

**Answer**: A

Which of the following is **true** about **row-level locking** in SQL Server?

* A) Row-level locking occurs by default on all SQL Server operations.
* B) Row-level locks reduce concurrency but improve performance.
* C) Row-level locks are automatically escalated to page locks in some situations.
* D) Row-level locking prevents all forms of deadlock.

**Answer**: C

Which of the following is **true** about **table partitioning** in SQL Server?

* A) Table partitioning improves query performance by reducing the size of the index.
* B) Partitioned tables store rows based on a partition function, not a partition scheme.
* C) You can have a different index type on each partition of a partitioned table.
* D) Partitioning allows horizontal division of data, and each partition can reside on a different filegroup.

**Answer**: D

In SQL Server, which tool is primarily used to evaluate the **query execution plan**?

* A) SQL Profiler
* B) Database Engine Tuning Advisor
* C) Query Performance Monitor
* D) SQL Server Management Studio's "Include Actual Execution Plan" option

**Answer**: D

Which of the following is **true** about the **TempDB** database in SQL Server?

* A) TempDB is used for storing permanent user data.
* B) TempDB is cleared and recreated every time the SQL Server instance restarts.
* C) TempDB can store both system and user-defined temporary tables but not table variables.
* D) TempDB does not support transactions.

**Answer**: B

Which of the following **is true** about **Database Mirroring** in SQL Server?

* A) It requires three SQL Server instances for the high-safety mode with automatic failover.
* B) It does not require a witness server for automatic failover.
* C) The principal and mirror databases can be read from simultaneously.
* D) Database mirroring cannot be combined with transaction log shipping.

**Answer**: A

Which of the following **SQL Server isolation levels** allows **non-repeatable reads** but **prevents dirty reads**?

* A) READ UNCOMMITTED
* B) READ COMMITTED
* C) REPEATABLE READ
* D) SERIALIZABLE

**Answer**: B

Which of the following **Dynamic Management Views (DMVs)** provides **information about currently running queries**?

* A) sys.dm\_exec\_requests
* B) sys.dm\_exec\_sessions
* C) sys.dm\_exec\_query\_stats
* D) sys.dm\_db\_index\_operational\_stats

**Answer**: A

Which of the following is **true** regarding **backup compression** in SQL Server?

* A) Backup compression is enabled by default for all backups in SQL Server.
* B) Compressed backups consume more CPU but reduce I/O operations.
* C) Backup compression is only available in SQL Server Enterprise Edition.
* D) Compressed backups are not restorable to non-compressed versions of SQL Server.

**Answer**: B

Which **query hint** in SQL Server forces the query optimizer to ignore the parallel execution plan?

* A) MAXDOP
* B) OPTION(LOOP JOIN)
* C) OPTION(FAST 1)
* D) NOLOCK

**Answer**: A

Which of the following is **true** about **AlwaysOn Availability Groups** in SQL Server?

* A) It requires SQL Server Standard Edition.
* B) All secondary replicas must be read-only.
* C) You can configure automatic failover between synchronous replicas.
* D) You cannot perform backups on secondary replicas.

**Answer**: C

**Q: Can you walk us through a full-stack application you have developed using C# and .NET?**  
A: I developed an e-commerce web application using C#, ASP.NET MVC for the backend, and AngularJS for the frontend. The application allowed users to browse products, manage their shopping carts, and process payments securely. On the backend, I used Entity Framework to interact with a SQL Server database, ensuring efficient data management. The application also had a RESTful API layer for integration with a mobile app.

**Q: How would you implement a RESTful API in .NET?**  
A: I would use ASP.NET Web API to create RESTful services. I’d define the necessary HTTP methods like GET, POST, PUT, and DELETE. Each endpoint would be associated with an entity or resource. I’d handle data serialization using JSON format, apply model validation, and implement token-based authentication (e.g., JWT) for securing the API.

**Explain how you manage state in AngularJS applications.**  
A: In AngularJS, state management can be handled using services, which act as singletons and allow data to be shared across controllers. For persistent data, I would use localStorage or sessionStorage. Additionally, the $rootScope or shared services can be used to manage the state globally.

**How do you optimize performance in large-scale AngularJS apps?**  
A: Performance optimization can be achieved by minimizing the number of watchers, using one-time bindings (e.g., ::), lazy loading of modules, and reducing DOM manipulation. Caching HTTP requests using $cacheFactory and breaking the application into small, reusable components also help improve performance.

**Q: What is the role of dependency injection in .NET?**  
A: Dependency injection (DI) in .NET is used to achieve loose coupling between classes. It allows us to inject the dependencies an object needs at runtime, which makes the code more maintainable, testable, and modular. In ASP.NET Core, the built-in DI container is used to manage the lifetime of services.

**How do you manage authentication and authorization in a full-stack application?**  
A: I use token-based authentication (JWT) for stateless authentication. For authorization, I implement role-based or policy-based checks in the backend. In AngularJS, I use route guards to restrict access to certain views based on the user's role or permissions.

2. **SQL Server (Database Management)**

**Q: How do you optimize queries in SQL Server?**  
A: I optimize queries by using appropriate indexing strategies, avoiding SELECT \*, and ensuring joins are written efficiently. I also make use of query execution plans to identify performance bottlenecks, and where necessary, use stored procedures to batch multiple operations.

**Q: What strategies do you use for database indexing and optimization?**

A: I analyze query patterns and frequently accessed columns to determine the need for indexes. I use clustered indexes for primary keys and non-clustered indexes for search columns. Periodically, I also defragment or rebuild indexes to maintain performance.

**Q: How do you handle database migrations in production systems?**  
A: I use migration tools like Entity Framework’s Code First Migrations or Redgate’s SQL Change Automation to handle database schema changes. I ensure the migrations are thoroughly tested in staging before deploying them to production. I also make use of database backups as a safety measure.

**Can you explain a challenging SQL problem you've solved recently?**  
A: In a recent project, I was tasked with optimizing a complex report generation query that was taking too long to execute. After analyzing the query execution plan, I realized that a full table scan was happening on a large table. By adding a composite index and rewriting the joins more efficiently, I was able to reduce the query time significantly.

**How do you ensure data consistency and integrity in your applications?**  
A: I ensure data consistency using transactions, foreign key constraints, and proper normalization of database schema. For integrity, I enforce business rules at both the application and database levels, using validation logic and triggers where necessary.

**Q: How do you decide between using IaaS and PaaS in Azure?**  
A: The choice depends on the level of control and management required. IaaS gives more control over the infrastructure (e.g., VMs), whereas PaaS abstracts the infrastructure, allowing the focus to be on application development (e.g., Azure App Services). For scalability and reduced management overhead, I typically prefer PaaS unless there is a need for custom configurations.

**Q: Can you describe a project where you leveraged Azure’s scalability features?**  
A: I worked on an application hosted in Azure App Services that needed to handle unpredictable traffic spikes. By configuring auto-scaling rules based on CPU and memory usage, we ensured the application could dynamically scale during peak hours and scale back during off-peak times, optimizing both performance and costs.

**Q: How do you implement security best practices in Azure?**  
A: I follow Azure security best practices such as using Azure AD for authentication, encrypting data at rest and in transit, using Network Security Groups (NSGs) for VMs, enabling multi-factor authentication (MFA), and implementing role-based access control (RBAC) to limit access to resources.

**Q: What monitoring tools do you use to track the performance of applications deployed in Azure?**  
A: I use Azure Monitor, Application Insights, and Azure Log Analytics to monitor the performance, uptime, and health of deployed applications. These tools help in identifying bottlenecks and proactively alerting in case of any anomalies.

**Q: Can you explain how Azure Kubernetes Service (AKS) works?**  
A: AKS is a managed Kubernetes service in Azure that simplifies the deployment, management, and operations of Kubernetes clusters. It automates tasks like cluster scaling, upgrades, and patching. With AKS, I can focus more on deploying containerized applications, while Azure handles the underlying infrastructure.

. **DevOps Practices (CI/CD, Containers)**

**Q: How do you implement Continuous Integration and Continuous Delivery (CI/CD) pipelines?**  
A: I typically use tools like Azure DevOps or Jenkins to set up CI/CD pipelines. The pipeline automates the build, test, and deployment process. For CI, every commit triggers a build and automated tests. For CD, successful builds are automatically deployed to staging environments, and with manual approval, to production.

**Q: How do you ensure zero-downtime deployments in a microservices environment?**  
A: I use strategies like blue-green deployments or canary releases, where traffic is gradually shifted from the old version to the new one. This ensures that any issues with the new deployment don’t affect all users, and it allows for a quick rollback if necessary.

**Q: Explain the role of containers in DevOps and their benefits.**  
A: Containers, like Docker, help in packaging applications and their dependencies into isolated units that can run consistently across different environments. This solves the "it works on my machine" problem and makes development, testing, and deployment processes more predictable.

**Q: What challenges have you faced with Docker and how did you resolve them?**  
A: One challenge I faced was managing stateful applications in Docker containers. To resolve this, I used volumes to persist data and also ensured proper orchestration using Kubernetes to manage the containers’ lifecycle and ensure high availability.

**Q: How do you manage scaling in a Kubernetes environment?**  
A: I use Kubernetes’ Horizontal Pod Autoscaler to automatically scale the number of pods based on CPU or memory utilization. I also configure resource requests and limits for each container to ensure optimal scaling and avoid resource contention.

**5. Automation and Scripting (PowerShell, Infrastructure Management)**

**Q: Can you give an example of a task you automated using PowerShell?**  
A: I automated the deployment of virtual machines in Azure using PowerShell scripts. The script created VMs, assigned them to specific subnets, configured network security groups, and installed necessary software.

**Q: How do you manage infrastructure as code (IaC) using tools like Terraform or ARM templates in Azure?**  
A: I use Terraform for infrastructure management due to its cloud-agnostic nature. For Azure-specific resources, I also use ARM templates. Both tools allow me to version control the infrastructure configurations, making it easier to replicate environments and automate resource provisioning.

**Q: Explain how you handle monitoring and alerting in automated environments.**  
A: I set up monitoring tools like Azure Monitor and integrate them with automated alerting systems. Alerts are configured to notify on critical metrics like CPU usage, memory, or request failures. These alerts trigger automated remediation scripts if necessary.

**Q: What kind of PowerShell scripts have you written for cloud infrastructure management?**  
A: I’ve written scripts for tasks like bulk VM creation, automating resource group cleanups, rotating secrets in Key Vault, and automating the scaling of services based on demand.

**Q: What are the key areas you focus on during a code review?**  
A: I focus on code readability, adherence to coding standards, performance implications, security concerns, and the overall design of the solution. I also check for proper test coverage and adherence to SOLID principles.

**Q: How do you enforce coding standards across a team?**  
A: I use code linters, formatters, and pre-commit hooks to ensure consistency. Additionally, I hold regular code reviews where team members provide feedback. Documentation and guidelines on best practices are also maintained to ensure uniformity across the team.

**Q: Can you explain how you ensure high test coverage in your codebase?**  
A: I follow test-driven development (TDD) whenever possible. I write unit tests for individual components and use integration tests to validate the system as a whole. Continuous Integration pipelines are configured to fail builds if the code coverage falls below a certain threshold.

**Q: What are your strategies for refactoring code?**  
A: I refactor code in small, manageable steps, ensuring that the code continues to work as expected by running tests after each change. I focus on improving code readability, reducing complexity, and eliminating technical debt without changing the system’s external behavior.