How do you troubleshoot performance issues in a Linux environment?

A.

Sample answer

As a Back End Developer, troubleshooting performance issues in a Linux environment is a critical skill. I utilize various tools and techniques to identify and resolve performance bottlenecks. Firstly, I analyze system resource usage using 'top', 'htop', or 'glances' to identify processes consuming excessive CPU, memory, or disk I/O. Additionally, I use 'vmstat' and 'iostat' to monitor system statistics and identify any potential resource saturation. If network-related issues are suspected, 'iftop' or 'nethogs' can provide insights into network bandwidth usage. I also leverage performance profiling tools like 'perf' to analyze specific application performance and identify areas for optimization. By systematically analyzing and diagnosing performance issues, I can take appropriate actions such as optimizing code, scaling resources, or tuning system parameters to improve overall performance.

ersonalised questions

Common questions

Commonly asked questions for your role

cross-icon

Q.

How do you troubleshoot performance issues in a Linux environment?

A.

Sample answer

As a Back End Developer, troubleshooting performance issues in a Linux environment is a critical skill. I utilize various tools and techniques to identify and resolve performance bottlenecks. Firstly, I analyze system resource usage using 'top', 'htop', or 'glances' to identify processes consuming excessive CPU, memory, or disk I/O. Additionally, I use 'vmstat' and 'iostat' to monitor system statistics and identify any potential resource saturation. If network-related issues are suspected, 'iftop' or 'nethogs' can provide insights into network bandwidth usage. I also leverage performance profiling tools like 'perf' to analyze specific application performance and identify areas for optimization. By systematically analyzing and diagnosing performance issues, I can take appropriate actions such as optimizing code, scaling resources, or tuning system parameters to improve overall performance.

footer-thumbs-upfooter-thumbs-down

Viewed

Q.

How can you implement authentication and authorization in a Java backend application using Spring Security?

A.

Sample answer

Spring Security is a powerful framework that provides authentication and authorization capabilities for Java backend applications. It allows you to secure your application at the method, class, or URL level, ensuring that only authorized users can access protected resources. To implement authentication, you can leverage Spring Security's built-in authentication providers, such as in-memory authentication, user-details service, or LDAP. These providers authenticate the user's credentials against the configured authentication source, such as a database or an LDAP server. Once the user is authenticated, Spring Security generates a security context that contains the user's details and authorities. Authorization can be implemented using various mechanisms, including role-based access control (RBAC), permissions, or custom access control expressions. You can define authorization rules using annotations, XML configurations, or custom AccessDecisionVoters. Spring Security also provides features like session management, CSRF protection, password encoding, and integration with external identity providers, such as OAuth or SAML. Overall, Spring Security simplifies the process of implementing secure authentication and authorization in Java backend applications, enabling developers to focus on business logic rather than security concerns.

How can you implement authentication and authorization in a Java backend application using Spring Security?

How does the Model layer in MVC architecture handle data storage and retrieval? Provide an example of using Spring and Hibernate for data access.

Q.

What is the role of Maven in Java development?

Q.

What is the use of environment variables in Linux?

Q.

How would you optimize a Java application for performance?

Q.

How do you search for a specific text within files in Linux?

Q.

How do you automate software deployments using Linux tools?

Q.

How do you monitor and analyze system logs in a Linux environment?

Q.

What is the role of Docker in software development?

Q.

How does Spring Boot simplify the process of building and deploying backend applications in Java?

Q.

How do you compress and decompress files in Linux?

Q.

How do you configure and manage service daemons in a Linux environment?

Q.

What is the role of jQuery in web development?

Q.

How would you design a microservices architecture in a Java application?

Q.

How do you configure and secure web servers in a Linux environment?

Q.

How do you schedule recurring jobs in Linux?

Q.

Explain the role of a version control system like Git in backend development

Q.

Explain the role of Ajax in frontend development and how it can be used with a Java backend.