108 Docker interview questions to hire top engineers

Questions

1. What is Docker and why do people use it? Imagine explaining it to a friend who doesn't know anything about computers.

2. Can you describe the difference between a Docker image and a Docker container?

3. What are some common Docker commands you might use every day?

4. How can you list all the Docker containers that are currently running?

5. How do you stop a running Docker container?

6. What is a Dockerfile, and what is its purpose?

7. Can you explain how to create a simple Dockerfile?

8. What does the 'docker build' command do?

9. What is a Docker Hub, and what is it used for?

10. How do you pull an image from Docker Hub?

11. How do you run a Docker image as a container?

12. How can you see the logs of a Docker container?

13. What are Docker volumes, and why are they important?

14. Explain the difference between a bind mount and a Docker volume.

15. What are environment variables in Docker, and how can you use them?

16. How can you set environment variables when running a Docker container?

17. What is Docker Compose, and when would you use it?

18. Can you give a simple example of a Docker Compose file?

19. How would you remove a Docker image from your local machine?

20. What are some advantages of using Docker over virtual machines?

21. If a container isn't working how would you troubleshoot it?

22. What is Docker, in simple terms, and why do developers use it?

23. Can you explain what a Docker image is, as if you were explaining it to someone who knows nothing about computers?

24. What's the difference between a Docker image and a Docker container?

25. How do you start a Docker container?

26. What is a Dockerfile, and what is it used for?

27. Can you name a few basic Docker commands you've used?

28. How would you check if a Docker container is running?

29. What is Docker Hub, and what can you find there?

30. Have you ever pulled an image from Docker Hub? If so, which one?

31. What does it mean to 'build' a Docker image?

32. Why might you want to use Docker in your projects?

33. If you made changes to a file inside a running container, would those changes be saved automatically?

34. What is the purpose of exposing ports when running a Docker container?

35. How can you see the logs of a Docker container?

36. What is a Docker volume, and why is it useful?

37. Can you explain a simple use case for Docker Compose?

38. What are environment variables in the context of Docker, and why are they helpful?

39. How do you stop a running Docker container?

40. What does the docker ps command do?

41. What is the purpose of the .dockerignore file?

42. Have you ever encountered a problem using Docker? What was it, and how did you solve it?

43. What is the difference between CMD and ENTRYPOINT in a Dockerfile?

44. How can you copy files from your local machine into a Docker container?

45. What's the difference between using COPY and ADD in a Dockerfile?

46. How do you clean up unused Docker images and containers to free up space?

47. What is a multi-stage Docker build, and why might you use one?

48. How do you specify the base image in a Dockerfile?

49. What is a Docker network, and why might you need to create one?

50. Can you describe a scenario where you would need to use Docker networking?

51. What are some best practices for writing Dockerfiles?

52. How would you implement zero-downtime deployments with Docker and a container orchestration tool?

53. Can you explain the difference between the COPY and ADD instructions in a Dockerfile, and when would you use each?

54. Describe a scenario where you would use multi-stage builds in Docker, and explain the benefits.

55. How would you secure sensitive information, such as API keys or passwords, in your Docker containers?

56. Explain how you can monitor the health and performance of your Docker containers in a production environment.

57. What are Docker volumes, and how do they differ from bind mounts? When would you choose one over the other?

58. How can you optimize Docker image size to reduce build times and improve deployment speed?

59. Describe your experience with Docker Compose. How does it simplify the process of managing multi-container applications?

60. How can you use Docker to create a development environment that is consistent across different machines?

61. Explain how Docker namespaces and cgroups contribute to container isolation and resource management.

62. What is a Docker registry, and how does it facilitate the sharing and distribution of Docker images?

63. How would you go about debugging a Docker container that is experiencing issues in a production environment?

64. Explain the concept of Docker networking. How can containers communicate with each other and with the outside world?

65. What are the best practices for writing Dockerfiles to ensure reproducibility and maintainability?

66. How can you use Docker to implement continuous integration and continuous delivery (CI/CD) pipelines?

67. Describe a time when you had to troubleshoot a complex Docker-related problem. What steps did you take to resolve it?

68. What is Docker Swarm, and how does it compare to Kubernetes for container orchestration?

69. How can you manage and persist data generated by Docker containers?

70. Explain the difference between docker run and docker start.

71. How would you implement a rolling update strategy using Docker and a container orchestrator?

72. What are Docker secrets, and how do they provide a secure way to manage sensitive data in Docker Swarm?

73. Explain how you can use Docker Healthcheck to monitor the health of your applications and automatically restart unhealthy containers.

74. Describe how you would configure logging for your Docker containers to collect and analyze application logs.

75. How can you limit the resources (CPU, memory) that a Docker container can consume?

76. Explain the purpose of a .dockerignore file and how it can improve Docker build performance.

77. How would you use Docker to package and deploy a machine learning model?

78. Explain the concept of Docker image layering and its impact on image size and build times.

79. Let's say you have a Docker container experiencing high CPU usage. How would you diagnose the root cause?

80. How would you approach optimizing a Dockerfile for smaller image size and faster build times, and what tools or techniques would you employ?

81. Describe a situation where you had to troubleshoot a complex networking issue between Docker containers. What steps did you take to diagnose and resolve the problem?

82. Explain your experience with Docker Swarm or Kubernetes for orchestrating Docker containers. What are the pros and cons of each, and when would you choose one over the other?

83. How do you handle persistent data in Docker containers, and what are the different options available for managing volumes?

84. What are some security best practices you follow when building and deploying Docker containers, and how do you mitigate potential security risks?

85. Describe your experience with implementing CI/CD pipelines for Dockerized applications. What tools and processes did you use to automate the build, test, and deployment process?

86. How would you monitor the health and performance of Docker containers in a production environment, and what metrics would you track?

87. Explain your understanding of Docker's underlying architecture and how it utilizes Linux kernel features like namespaces and cgroups.

88. How do you handle versioning and rolling back Docker images in a production environment?

89. Describe a time when you had to debug a performance bottleneck in a Dockerized application. What tools and techniques did you use to identify and resolve the issue?

90. Explain your experience with multi-stage builds in Dockerfiles. What are the benefits of using multi-stage builds, and how do they improve image size and security?

91. How would you design a Docker-based solution for a high-availability application, ensuring minimal downtime and automatic failover?

92. Describe your experience with using Docker Compose for defining and managing multi-container applications. What are the advantages of using Docker Compose over other orchestration tools?

93. How do you manage secrets and sensitive information in Docker containers, and what tools or techniques do you use to prevent them from being exposed?

94. Explain your understanding of Docker's networking model and how containers communicate with each other and with the outside world.

95. How do you handle logging in Docker containers, and what are the best practices for collecting, storing, and analyzing container logs?

96. Describe a time when you had to troubleshoot a failed Docker build. What steps did you take to diagnose and resolve the problem?

97. Explain your experience with using Docker in different environments, such as development, testing, and production. How do you adapt your Docker configurations for each environment?

98. How do you ensure consistency and reproducibility of Docker builds across different environments?

99. Describe your experience with using Docker for microservices architecture. What are the benefits and challenges of using Docker in a microservices environment?

100. How would you implement a blue-green deployment strategy using Docker containers?

101. Explain your understanding of Docker's storage drivers and how they affect container performance and storage utilization.

102. How do you handle resource constraints in Docker containers, such as CPU and memory limits, and how do you ensure that containers do not consume excessive resources?

103. Describe your experience with using Docker for legacy applications. What are the challenges of Dockerizing legacy applications, and how do you overcome them?

104. How would you automate the process of building and deploying Docker images to a container registry?

105. Explain your understanding of Docker's security scanning tools and how you use them to identify and remediate vulnerabilities in Docker images.

106. How do you handle dependencies and conflicts between different Docker images in a multi-container application?