Interview questions:

Sure, let's expand the list to 70 questions, ensuring variety and coverage across the specified topics:

### Git & GitHub:

1. What is Git, and how does it work?

2. Differentiate between Git and GitHub.

3. How do you initialize a Git repository?

4. Explain the concept of Git staging area.

5. How do you create a new branch in Git?

6. What is a Git commit, and how do you create one?

7. How do you merge branches in Git?

8. Explain the difference between Git merge and rebase.

9. How do you handle merge conflicts in Git?

10. What is Git cherry-pick, and when would you use it?

11. How do you revert a commit in Git?

12. Explain the purpose of the .gitignore file.

13. What is Git bisect, and how does it work?

14. How do you view the commit history in Git?

15. Explain the difference between Git pull and Git fetch.

### Maven:

16. What is Maven, and how does it help in software development?

17. How do you create a new Maven project?

18. What is a POM file in Maven?

19. Describe the Maven lifecycle phases.

20. How do you manage dependencies in Maven?

21. What is a Maven repository, and how do you use it?

22. Explain the purpose of Maven plugins.

23. How do you run tests using Maven?

24. What is Maven archetype, and when would you use it?

25. How do you deploy artifacts to a Maven repository?

### Jenkins:

26. What is Jenkins, and how does it facilitate continuous integration?

27. How do you install and configure Jenkins?

28. Explain the concept of Jenkins pipeline.

29. What are Jenkins agents, and how do they work?

30. How do you trigger a Jenkins build?

31. What is Jenkinsfile, and how do you define pipelines using it?

32. How do you manage credentials in Jenkins?

33. Explain the difference between Jenkins freestyle and pipeline projects.

34. How do you secure Jenkins?

35. What are Jenkins plugins, and how do you install them?

### Docker:

36. What is Docker, and how does it work?

37. How do you create a Docker container?

38. Explain the difference between an image and a container in Docker.

39. What is a Dockerfile, and how do you use it?

40. How do you manage Docker volumes?

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

42. How do you network Docker containers?

43. What are Docker registries, and how do they work?

44. Explain the concept of Docker Swarm.

45. How do you monitor Docker containers?

### AWS CodeCommit:

46. What is AWS CodeCommit, and how does it compare to other version control systems?

47. How do you set up and configure AWS CodeCommit?

48. How do you manage branches in AWS CodeCommit?

49. What are the key security features of AWS CodeCommit?

50. How do you integrate AWS CodeCommit with CI/CD pipelines?

### AWS CodeDeploy:

51. What is AWS CodeDeploy, and how does it facilitate deployment?

52. How do you deploy an application using AWS CodeDeploy?

53. Explain the deployment strategies available in AWS CodeDeploy.

54. What are the key benefits of AWS CodeDeploy?

55. How do you troubleshoot deployment failures in AWS CodeDeploy?

### AWS CodeBuild:

56. What is AWS CodeBuild, and how does it fit into the CI/CD pipeline?

57. How do you configure a build project in AWS CodeBuild?

58. What are the build environments supported by AWS CodeBuild?

59. How do you integrate AWS CodeBuild with other AWS services?

60. What are the key security features of AWS CodeBuild?

### General DevOps:

61. What are the principles of DevOps, and how do they benefit organizations?

62. Explain the concept of infrastructure as code (IaC).

63. What is continuous integration (CI), and how does it differ from continuous deployment (CD)?

64. Describe blue-green deployment and canary deployment strategies.

65. How do you ensure the scalability and availability of a DevOps pipeline?

66. What are the key metrics for measuring the success of a DevOps implementation?

67. How do you handle sensitive data in a CI/CD pipeline?

68. Explain the concept of immutable infrastructure.

69. How do you ensure compliance and governance in a DevOps environment?

70. What are the best practices for documenting a DevOps pipeline?

These questions provide a comprehensive coverage of DevOps tools and practices commonly encountered at mid-level positions.