Short Questions

1. Explain the concept of forking in Git.
2. What is a repository in Git?
3. How does Git handle conflicts during merges?
4. What is the difference between a Git commit and a Git push?
5. Describe the role of Git hooks in source code management.
6. What is the purpose of a Git submodule?
7. How does Git handle file versioning?
8. What is the significance of the .gitignore file in Git?
9. Explain the concept of rebasing in Git.
10. What is the difference between Git and Mercurial?
11. How does Git ensure data integrity?
12. What is the purpose of Git's staging area?
13. Describe the concept of distributed version control.
14. What is the purpose of a Git tag?
15. How does Git handle file permissions?
16. What are the advantages of using Git over centralized version control systems?
17. Describe the role of Git branches in collaborative development.
18. What is the purpose of Git's reflog?
19. How does Git handle large binary files?
20. What is the purpose of Git bisect?
21. Describe the concept of cherry-picking in Git.
22. What is Git LFS, and what problem does it solve?
23. How does Git handle file renaming?
24. What is the purpose of Git's blame command?
25. Explain the concept of Git's three-stage workflow (working directory, staging area, and repository).
26. What is the purpose of integrating the system in software development?
27. Define build systems.
28. Explain the role of Jenkins build server in continuous integration.
29. What are build dependencies, and why are they important?
30. Name two popular build dependency management tools.
31. How do Jenkins plugins enhance its functionality?
32. Describe the file system layout of Jenkins.
33. What is the significance of the host server in Jenkins?
34. Define build slaves in the context of Jenkins.

35. How does Jenkins manage software on the host server?
36. What are triggers in Jenkins, and how do they work?
37. Explain the concept of job chaining in Jenkins.
38. What are build pipelines, and how do they improve the software development process?
39. Define infrastructure as code (IaC) in the context of build servers.
40. How does Jenkins handle building by dependency order?
41. Describe the different build phases in Jenkins.
42. Name two alternative build servers to Jenkins.
43. What measures are collated for assessing software quality in Jenkins?
44. Explain the purpose of artifact repositories in Jenkins.
45. What is the role of version control systems in Jenkins?
46. How does Jenkins handle parallel builds?
47. Describe the concept of distributed builds in Jenkins.
48. What is the purpose of Jenkins agents?
49. How does Jenkins manage security for build jobs?
50. Explain the concept of build pipelines in Jenkins.
51. What is the significance of continuous integration in Jenkins?
52. How does Jenkins handle test automation?
53. Describe the process of setting up a new build job in Jenkins.
54. What is Jenkinsfile, and how is it used?
55. How does Jenkins handle build failures?
56. What is the purpose of build artifacts in Jenkins?
57. Describe the role of environment variables in Jenkins.
58. How does Jenkins support code review processes?
59. What is the role of Jenkins in deployment automation?
60. Explain the concept of parameterized builds in Jenkins.
61. How does Jenkins handle build notifications?
62. Describe the role of build scripts in Jenkins.
63. What is the purpose of Jenkins pipelines?
64. How does Jenkins integrate with containerization technologies like Docker?
65. Explain the concept of build triggers in Jenkins.
66. What is the purpose of Jenkins plugins in enhancing functionality?
67. Describe the role of Jenkins agents in distributed builds.
68. How does Jenkins handle versioning of build artifacts?
69. Explain the concept of Jenkins master-slave architecture.
70. What is the purpose of the Jenkins dashboard?

71. Describe the role of Jenkins in continuous delivery.
72. How does Jenkins handle build scheduling?
73. What is the purpose of Jenkins global tool configuration?
74. Explain the concept of Jenkins build retention policy.
75. How does Jenkins integrate with continuous deployment processes?
76. What are the different types of testing commonly used in software development?
77. Describe the pros and cons of automating testing.
78. What is Selenium, and what is its primary purpose?
79. Name two key features of Selenium.
80. Explain the concept of JavaScript testing.
81. How do you test backend integration points in software development?
82. Define test-driven development (TDD).
83. What is REPL-driven development, and how does it differ from TDD?
84. Describe the role of deployment systems in software development.
85. Name two popular virtualization stacks used for deployment.
86. How is code executed at the client in deployment processes?
87. What is Puppet, and what role does it play in deployment?
88. Explain the concept of Puppet master and agents.
89. What is Ansible, and how does it differ from Puppet?
90. Name two deployment tools similar to Ansible.
91. What is the purpose of Chef in deployment processes?
92. Describe the role of SaltStack in deployment automation.
93. How does Docker facilitate deployment?
94. Define continuous integration (CI) testing.
95. Explain the concept of end-to-end (E2E) testing.
96. What is unit testing, and why is it important?
97. Describe the process of regression testing.
98. How does smoke testing differ from other types of testing?
99. Define acceptance testing in software development.
100. What is load testing, and when is it typically performed?
101. Explain the concept of black-box testing.
102. Describe the role of gray-box testing in software quality assurance.
103. What is boundary testing, and why is it relevant?
104. Define exploratory testing, and when is it commonly used?
105. How does integration testing differ from unit testing?
106. Explain the concept of fuzz testing.
107. Describe the purpose of security testing in software development.

108. What is usability testing, and how is it conducted?
109. Define performance testing, and give an example.
110. How does automated testing improve software development processes?
111. What are the limitations of automated testing?
112. How does Selenium automate web browser testing?
113. Describe the role of Selenium WebDriver in test automation.
114. What is the Selenium IDE, and how is it used?
115. Explain the concept of headless browser testing.
116. How do JavaScript testing frameworks like Jest facilitate testing?
117. Describe the process of testing backend integration points using frameworks like Postman.
118. What are the primary benefits of test-driven development (TDD)?
119. Explain the concept of red-green-refactor cycle in TDD.
120. How does REPL-driven development improve developer productivity?
121. Describe the role of deployment scripts in automated deployment processes.
122. What are the advantages of using virtualization stacks for deployment?
123. How does Docker containerization simplify deployment?
124. Explain the concept of infrastructure as code (IaC) in deployment automation.
125. What are the key considerations when selecting deployment tools for a project?