1. What are your daily responsibilities?

2. Have you confegured any CI/CD tool from scratch? What if the the agent is in offline mode, how will you make it up?

3. Can we deploy the pipeline through command line in jenkins

4. If there are more than two branches in github then how would you automate the deployment through jenkins?

5. What access you will give to developers to trigger the build & where you will configured that access??

6. What monitoring tool you have used for your system?

7. What kind of work you have done in Git/Github?

8. What is git rebase?

9. What is Cherry-pick merge?

10. For installation of CICD tool what kind of issues you have faced & how you trobleshoot those??

11. Have you worked on the documentation like user guide, troubleshooting procedures?

12. What are the scripting languages you know? What is the command the to check syntax in yaml file?

13. Have you worked on any artifactory & have you download or upload any file there??

14. For a project where there is no devops, so as a devops support engineer what will be your approach & how will implement the devops?

15. have you worked on pre-commit & post commit activity to avoid any commit from unknown user, or if something is missing it should not allow you to commit the code? Have your written any books or scripts for that??

16. Have you done the ansible configuration? can you elaborate more?

17. Apart from jenkins have you worked on any other CICD tool?

18. Have you any automation of process?

19. What is the command to kill docker container?

============================================================================================================================================================================================================

 **What are your daily responsibilities?**

* Daily responsibilities in a DevOps role typically include monitoring and maintaining infrastructure, troubleshooting issues, deploying updates, configuring and managing CI/CD pipelines, collaborating with development teams, managing cloud services (AWS, Azure, GCP), ensuring security and compliance, and automating processes to improve efficiency.

 **Have you configured any CI/CD tool from scratch? What if the agent is in offline mode, how will you make it up?**

* Yes, configuring CI/CD tools from scratch is a common task. If a CI/CD agent is in offline mode, I would start by checking the network connectivity and ensuring that the agent is properly configured to communicate with the master server. Restarting the agent service and reviewing logs for any errors are also crucial steps.

 **Can we deploy the pipeline through the command line in Jenkins?**

* Yes, Jenkins pipelines can be deployed through the command line using the Jenkins CLI or Jenkinsfile. The jenkins-cli.jar can be used with commands like build, cancel-quiet-down, and get-job to manage Jenkins jobs.

 **If there are more than two branches in GitHub, how would you automate the deployment through Jenkins?**

* To automate deployment for multiple branches, I would set up a multibranch pipeline in Jenkins. This pipeline would automatically detect branches in the GitHub repository and create jobs for each branch. Configuration can be done in the Jenkinsfile to define the deployment steps for each branch.

 **What access will you give to developers to trigger the build & where will you configure that access?**

* Developers would typically be given "build" access, allowing them to trigger builds but not modify the pipeline configuration. This can be configured in Jenkins under "Manage Jenkins" -> "Configure Global Security" -> "Matrix-based security" or "Role-based strategy".

 **What monitoring tool have you used for your system?**

* I have used various monitoring tools such as Prometheus, Grafana, Nagios, and AWS CloudWatch to monitor system performance, track metrics, and set up alerts for any issues.

 **What kind of work have you done in Git/GitHub?**

* My work in Git/GitHub includes branching, merging, creating pull requests, code reviews, managing repository permissions, setting up webhooks for CI/CD integrations, and automating workflows using GitHub Actions.

 **What is git rebase?**

* git rebase is a command that allows you to move or combine a sequence of commits to a new base commit. It’s often used to keep a linear project history, especially when working with feature branches.

 **What is Cherry-pick merge?**

* Cherry-picking in Git allows you to apply a specific commit from one branch to another. It’s useful when you want to incorporate a particular change without merging entire branches.

 **For the installation of a CI/CD tool, what kind of issues have you faced & how did you troubleshoot those?**

* Issues during CI/CD tool installation can include dependency conflicts, incorrect configurations, network issues, and permissions errors. Troubleshooting involves reviewing installation logs, verifying system requirements, testing network connectivity, and ensuring all necessary permissions are granted.

 **Have you worked on documentation like user guides or troubleshooting procedures?**

* Yes, I have written and maintained documentation such as user guides, installation manuals, and troubleshooting procedures to assist teams in understanding and managing DevOps tools and processes.

 **What are the scripting languages you know? What is the command to check syntax in a YAML file?**

* I am proficient in scripting languages like Bash, Python, and Groovy. To check the syntax of a YAML file, you can use the yamllint command: yamllint <file.yaml>.

 **Have you worked on any artifactory & have you downloaded or uploaded any files there?**

* Yes, I have worked with JFrog Artifactory, where I have managed artifacts by uploading and downloading files such as build artifacts, libraries, and dependencies.

 **For a project where there is no DevOps, as a DevOps support engineer, what will be your approach & how will you implement DevOps?**

* My approach would be to first assess the current processes, identify bottlenecks, and propose a plan to introduce automation, version control, and CI/CD practices. I would implement tools like Jenkins, Git, Docker, and Ansible, and gradually introduce monitoring, infrastructure as code, and continuous feedback loops.

 **Have you worked on pre-commit & post-commit activities to avoid any commits from unknown users, or if something is missing, it should not allow you to commit the code? Have you written any hooks or scripts for that?**

* Yes, I have worked on pre-commit and post-commit hooks to enforce coding standards, check for missing files, or prevent unauthorized commits. I have written custom scripts to automate these checks as part of the Git workflow.

 **Have you done the Ansible configuration? Can you elaborate more?**

* Yes, I have configured Ansible for automated provisioning, configuration management, and application deployment. I’ve written playbooks, managed inventory files, and used Ansible Galaxy roles to manage infrastructure across multiple environments.

 **Apart from Jenkins, have you worked on any other CI/CD tool?**

* Yes, I have experience with other CI/CD tools like GitLab CI, CircleCI, and Travis CI, where I have configured pipelines, integrated with various services, and managed deployments.

 **Have you done any automation of processes?**

* Yes, I have automated various processes such as infrastructure provisioning with Terraform, CI/CD pipelines with Jenkins, and monitoring setups with scripts that deploy and configure Prometheus and Grafana.

 **What is the command to kill a Docker container?**

* To kill a Docker container, you can use the command: docker kill <container\_id>. If you want to stop it gracefully, use docker stop <container\_id>.