You are the DevOps lead for a software project. Your team uses Git for version control and an issue tracking system like Jira or GitHub Issues. The project has a critical bug that needs fixing.

Question:

How would you allocate the task of fixing the critical bug to a team member using Git and the issue tracking system?

Task1: Describe the process of creating a new issue or bug report in the issue tracking system.

* **Access the Issue Tracking System**
* **Create a New Issue**
* **Label and Categorize**
* **Set Priority**
* **Add Attachments**
* **Submit the Issue**

**Steps:**

* Log in to your issue tracking system account and navigate to the project where you want to report the bug
* Jira: Click on the "+ Create" button at the top of the project board, then select "Issue" or "Bug." Fill in the required information, including the issue summary, description, and any relevant details such as steps to reproduce and expected behavior.
* GitHub Issues: Click on the "New Issue" button in the repository's "Issues" tab. Provide a clear title and detailed description of the bug.
* Assign relevant labels, components, or categories to the issue to help categorize and prioritize it. This makes it easier for team members to find and work on issues.
* Specify the priority level of the issue (e.g., critical, high, medium, low) to communicate its urgency.
* If applicable, attach screenshots, logs, or any other supporting files that can help the team understand and reproduce the issue.
* Finally, click the "Create" or "Submit" button to create the issue. It will now be visible to the team.

Task 2: Explain how to assign the issue to a specific team member.

* **Access the Issue**
* **Assignee Selection**
* **Save Changes**

**Steps:**

* Open the issue that needs to be assigned within the issue tracking system.
* Jira: Look for the "Assignee" field on the issue details page. Click on it and select the team member from the dropdown list or start typing their name to search for them.
* GitHub Issues: On the issue details page, click the "Assignees" section on the right side. Start typing the team member's GitHub username or name, and select them from the suggestions.
* After selecting the team member, save the changes to assign the issue to them. They will now receive notifications about the assignment.

Task 3:Describe how to use Git branches and pull requests to work on the issue, ensuring that changes are tracked and reviewed

* **Create a Git Branch**
* **Make Changes**
* **Push Changes to Remote**
* **Create a Pull Request (PR)**
* **Review and Collaboration**
* **Merge the PR**
* **Close the issue**

**Steps:**

* Using the command line, run git checkout -b issue-branch where "issue-branch" is a descriptive name for your branch.
* Ensure that your local repository is up-to-date with the latest changes from the main or development branch by running git pull from the main branch before creating the new branch.
* Make the necessary code changes to fix the critical bug within your newly created branch. Commit your changes with clear and concise commit messages.
* Push your branch to the remote repository with git push origin issue-branch
* Navigate to the repository on GitHub.
* Click on the "Pull Requests" tab and then the "New Pull Request" button.
* Select your issue-branch as the base branch (usually "main" or "develop") and the target branch.
* Add a clear title and description for the PR, referencing the issue number
* Click "Create Pull Request."
* Team members can review the changes, provide feedback, and discuss the code within the PR's comments.
* Address any feedback and make additional commits to the same branch.
* Once the PR is approved and all checks pass, it can be merged into the target branch.
* After merging, the issue can be automatically closed by referencing it in the PR's description Many issue tracking systems will detect this and close the issue for you.