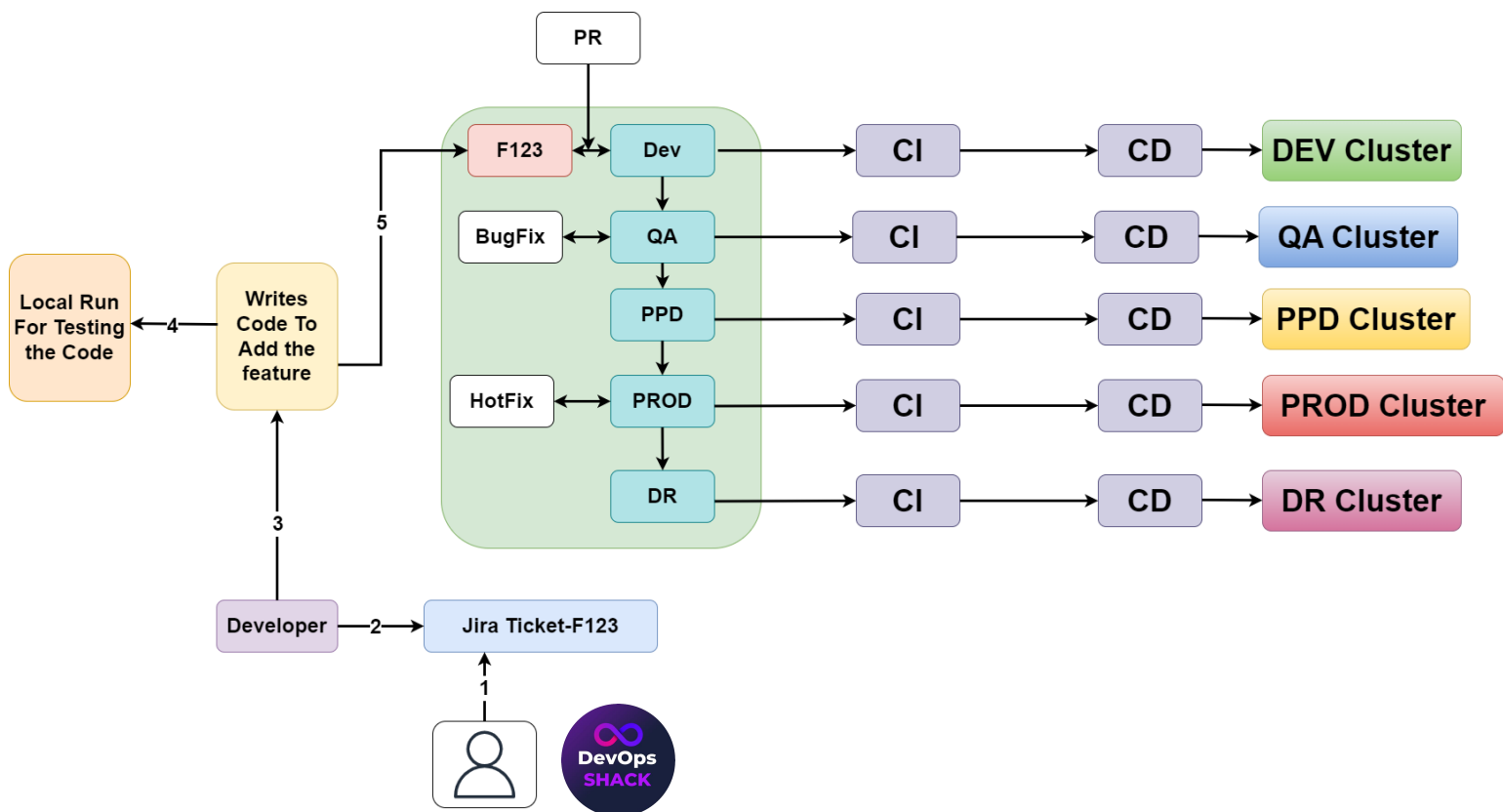# DevOps Shack

## DAY-1 | DevOps Interview Preparation

**Click Here To Enrol To Batch-5 | DevOps & Cloud DevOps**

## DevOps Workflow Documentation

# Step 1: Feature Request

- **Client Action**: A client identifies the need for a new feature in an application and submits a request.
- **Jira Ticket Creation**: The client raises a ticket in Jira to officially request the new feature.

# Step 2: Code Development

- **Developer Assignment**: A developer is assigned to the ticket and starts working on the new code for the requested feature.

# Step 3: Local Testing

- **Initial Testing**: After completing the code, the developer tests it on their local machine to ensure it functions correctly and meets the initial requirements.

# Step 4: Feature Branch Creation

- **Branch Strategy**: Once the local tests pass, the developer creates a feature branch from the main development (Dev) branch. This helps in isolating the new changes from the main codebase.

# Step 5: Pull Request and Review

- **Pull Request**: The developer submits a pull request to merge the feature branch into the Dev branch.
- **Code Review and Approval**: An architect reviews the changes. If approved, the changes are merged into the Dev branch.

# Step 6: CI/CD Pipeline Setup for Dev

- **DevOps Engineer Role**: A DevOps engineer sets up a continuous integration and continuous deployment (CI/CD) pipeline specifically for deploying the new changes to the development environment

# Step 7: Deployment to QA

- **Quality Assurance**: If the deployment in the Dev environment is successful and stable, the code is then merged from the Dev branch to the QA (Quality Assurance) branch.
- **Deployment**: The changes are deployed to the QA environment for further testing.

# Step 8: Staged Deployment to Other Environments

- **Repeat Process**: This deployment process is repeated for the Pre-Production (PPD), Production (PROD), and Disaster Recovery (DR) environments.
- **Sequential Deployment**: Each environment is updated sequentially to ensure stability at each stage.

# Step 9: Handling Bugs in Non-Prod Environments

- **Bug Identification**: If any bugs are found in environments below Production (e.g., QA, PPD), a bugfix branch is created.
- **Bug Fixing**: The issue is fixed in this branch.
- **Merging**: The bugfix branch is then merged back into the original branch from which it was created, ensuring the fixes are propagated through the subsequent environments.

# Step 10: Handling Production Critical Issues

- **Hotfix Process**: For critical issues that arise in the Production environment, a hotfix branch is created.
- **Issue Resolution**: The critical issue is fixed within this branch.
- **Immediate Deployment**: The hotfix branch is then merged directly into the Production branch to quickly rectify the problem without disrupting the production services.

This documentation provides a comprehensive outline of the processes and steps involved in managing and deploying new features and fixes in a typical DevOps workflow. Each step ensures that the code is thoroughly tested and reviewed before making its way to the production environment, thus maintaining the reliability and integrity of the application.