Deployment Guide

Contents

[Development and Deployment Workflow 2](#_Toc22581672)

[Process Diagram 3](#_Toc22581673)

[Incident Management: 5](#_Toc22581674)

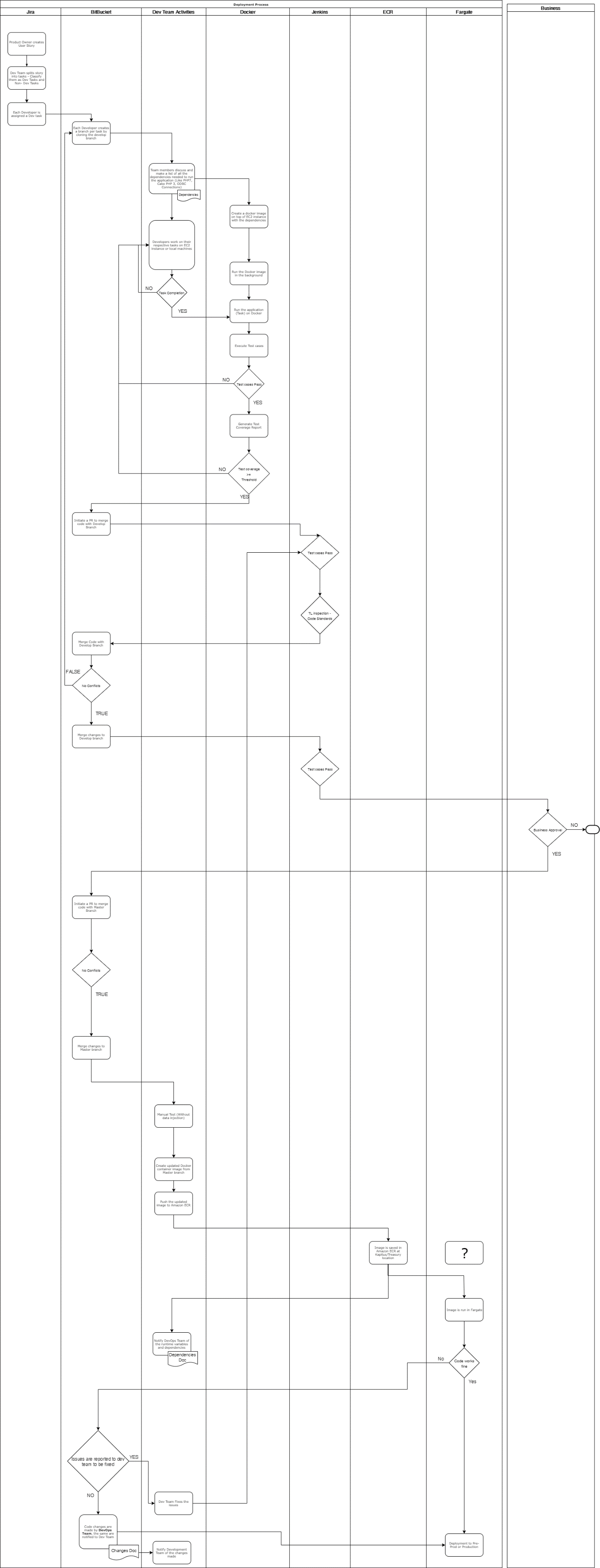
[Workflow 5](#_Toc22581675)

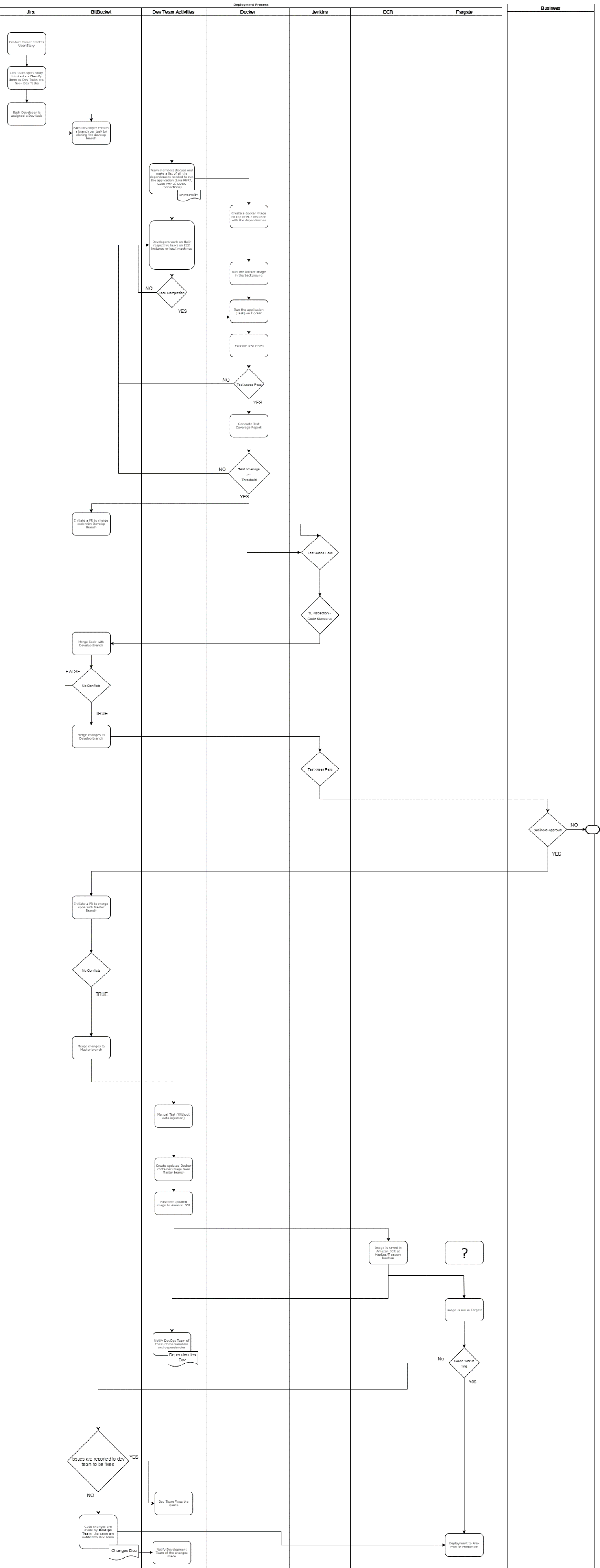
[RACI 6](#_Toc22581676)

# Development and Deployment Workflow

1. Product Owner creates User Story in Jira
2. Dev Team splits story into tasks - Classifies them as Dev Tasks and Non-Dev Tasks
3. Each Developer is assigned a Dev task
4. Each Developer creates a branch per dev task by cloning the develop branch
5. Team members discuss and make a list of all the dependencies needed to run the application (Like PHP7, Cake PHP 3, ODBC Connections)
6. Developers work on their respective tasks on EC2 instance or local machines
7. In the meanwhile, docker image is to be created on top of EC2 instance with the dependencies. Run the Docker image in the background before running the application
8. Run the application (Task) on Docker
9. Developers Execute Test cases on Docker application and generate a test coverage report. For the build to qualify, Test coverage >= Threshold.
10. TL inspects Code Standards
11. If the build passes the test coverage and TL's inspection, initiate a PR to merge code with Develop Branch
12. When PR is initiated, Jenkins executes the test cases. If there are No Conflicts in Bitbucket, the code will be merged to develop, and the test cases are executed. If any case fails, Jenkins notifies of it and the developer has to work on it.
13. If the business approves a feature to be moved to Master, initiate a PR to merge code with Master Branch
14. If there are No Conflicts in Bitbucket, the code will be merged to develop branch.
15. Manual Test (Without data injection)
16. Create updated Docker container image from Master branch and push it to Amazon ECR in Kapitus/Treasury location.
17. //Need more inputs of what happens next (AWS Fargate, etc)

## Process Diagram



  
Ref: <https://strategicfunding.atlassian.net/wiki/x/MIBGH>

# Incident Management:

In the event of occurrence of issue(s) in production, the identifier should raise a Helpdesk ticket immediately and alert the IT support team. The issue can be functional or Non-Functional – and can be alerted by Application monitoring tools like Cloudwatch/PRTG.

The support team will validate the same and create a ticket in Jira and tag concerned personnel. The Product owner should be immediately notified of the issue if its severity is high.

Kapitus IT team (DevOps) team will perform the preliminary troubleshooting to locate the fault (Server issue/code issue).

The analysis and the findings are to be clearly communicated to the product owner and the concerned project team, who will make work on identifying the root cause.

Temporary countermeasures are devised as soon as possible in consulting with the Business and Product Owner. The impact of the issue is analyzed and communicated to the product owner and the business and an action plan is prepared to resolve the issue jointly by the product owner and the project team.

Workflow**:**

Concerned teams scan the logs (PRTG or CloudWatch) to identify the root cause

Ticket is raised in Jira with severity (Concerned team members can be tagged)

Helpdesk ticket is created by Business

Severity Profile

Low

High

Product Owner is alerted immediately

Impact Analysis

Incorporate temporary countermeasures (If possible)

Prepare an Action Plan with Product Owner

CloudWatch/PRTG Sends Alert message

Business users note an issue

## RACI

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Category** | **Activities** | **Contact/Tool** | **Helpdesk Team** | **Kapitus IT** | **Ameex** | **Product Owner** | **Business** |
| Production Issues | Raise ticket in Helpdesk | Helpdesk | I | I |  | I | R |
| Categorize issue and post to JIRA | JIRA |  | R |  |  | R |
| Alert Kapitus IT Team and Ameex Team | JIRA/Phone/Email |  |  |  | R |  |
| Preliminary troubleshooting / Impact Analysis | NA |  | R | R | I/C |  |
| Temporary Countermeasures | NA |  | R | R | C |  |
| Ensure Pre-requisites for issue-fix (Log monitoring tools, access to production, etc.,) | NA | NA | R | C | R | NA |
| Action Plan Preparation | NA | NA | R | R | A | C |
| Issue Fix | NA |  | R | R | C |  |