

**TRAIN WITH  
SHUBHAM**

# JIRA

## Project Management Tool Integration With Github



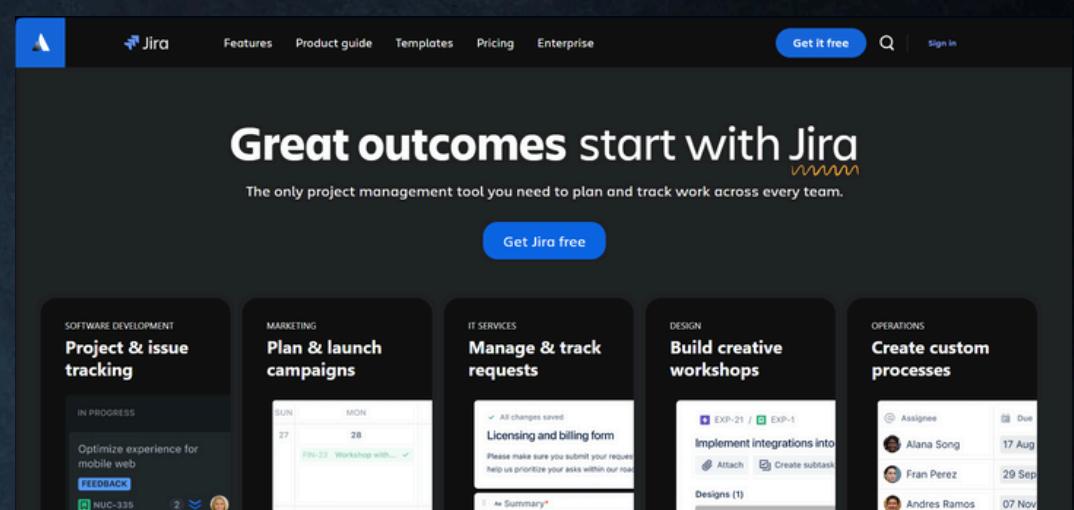
## Introduction:

### **What is Jira?**

Jira is like the central command center for your projects. It keeps everything organized and helps your team communicate effectively. Whether you're a small team working on a simple app or a large company managing multiple projects, Jira can scale to meet your needs.

Basically, it's a software developed by a company called Atlassian. It's a powerful tool used mainly for three things:

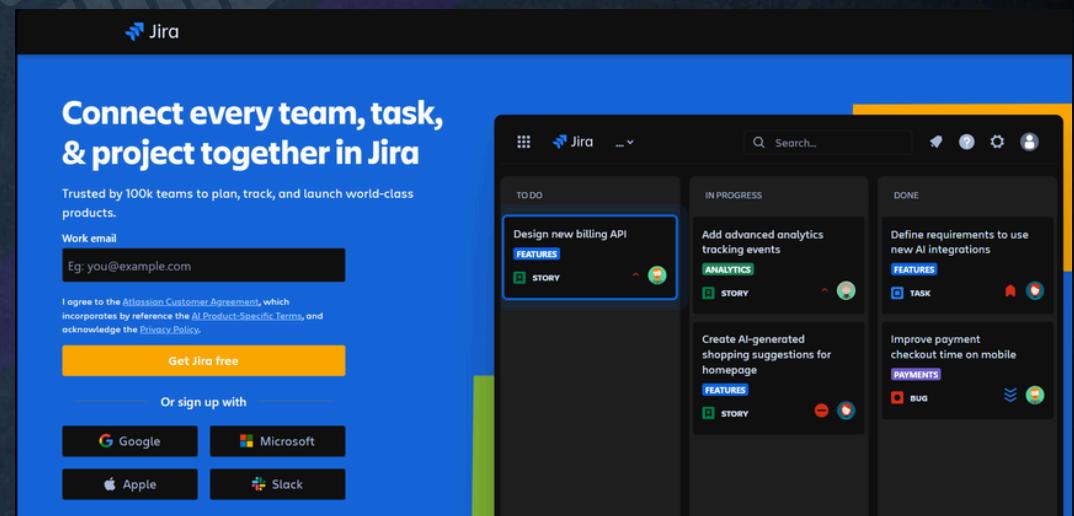
1. Project Management
2. Issue Tracking
3. Agile Software Development



### **Key Features of Jira**

**Customizable Workflows:** Workflows in Jira define how tasks move from start to finish. You can customize these workflows to fit your team's needs. For example, a workflow for fixing a bug might look like this:

- Story
- Story Point
- Epic
- To Do
- In Progress
- In Review
- Done
- Sprint
- Velocity Chart



**Detailed Reporting:** Jira provides various reports to help you understand how your project is progressing. For example, you can see how many tasks are completed, how many are still in progress, and how many are yet to be started. This helps in planning and improving future projects.

**Robust Integration Options:** Jira can connect with many other tools your team might be using. For example, it integrates with GitHub (a code hosting platform) so that code changes can be linked to tasks in Jira. It also integrates with Slack (a communication tool) to get updates about your project directly in your chat.

## Story vs Epic

In Jira, both *Stories* and *Epics* are used to organize work but they serve different purposes and scopes.

A *Story* represents a single piece of work that needs to be completed within a sprint. It's usually a smaller task or requirement that a team can accomplish in a short time frame.

*Stories* often follow a user-focused format like "As a user, I want to be able to log in so that I can access my account."

An *Epic*, on the other hand, is a larger body of work that can be broken down into multiple stories. *Epics* often span multiple sprints and are used to manage more complex projects or features that require substantial effort and time to complete. For example, an epic might be "Implement user authentication system," which would then be broken down into stories like "Create login page," "Set up database for user credentials," and "Implement password reset feature."

## Sprint

- **Definition:** A sprint is a time-boxed period, typically 1-4 weeks, during which a specific set of work must be completed and made ready for review.
- **Purpose:** Allows teams to break down their work into manageable chunks and deliver increments of value regularly.
- **Process:**
  - *Sprint Planning:* Define what can be delivered in the sprint and how that work will be achieved.
  - *Daily Standups:* Short meetings to discuss progress and obstacles.
  - *Sprint Review:* At the end of the sprint, review what was accomplished.
  - *Sprint Retrospective:* Reflect on the sprint to identify improvements.

## Velocity Chart

- **Definition:** A velocity chart tracks the amount of work a team completes during each sprint.
- **Purpose:** Helps predict how much work a team can realistically achieve in future sprints.
- **Components:**
  - *Velocity:* The average amount of story points completed in previous sprints.
  - *Displayed:* As a bar chart showing story points completed per sprint.
- **Usage:** Teams use the velocity to plan sprints by understanding their capacity.

## **Story Points**

- **Definition:** Story points are units of measure for expressing the overall effort required to fully implement a product backlog item or any other piece of work.
- **Purpose:** Helps estimate the complexity, effort, and time required to complete a story.
- **Assignment:**
  - **Relative Estimation:** Compare stories to each other to assign points.
  - **Common Scale:** Fibonacci sequence (1, 2, 3, 5, 8, 13, etc.) is often used.
- **Example:** A simple task might be 1 point, a more complex task 5 points, and a very complex task 13 points.

## **Dashboards and Gadgets**

Jira's Dashboards provide a customizable interface to display key project information using various gadgets. Dashboards help in visualizing data and tracking progress at a glance. Gadgets include:

- **Issue Statistics:** Show statistics on issue types, priorities, or custom fields.
- **Sprint Burndown:** Visualize the progress of a sprint.
- **Pie Charts:** Display issue distribution by status, assignee, or other fields.

## **Permissions and Security**

Jira provides robust permissions and security settings to control access and ensure data integrity. Features include:

- **Project Roles:** Define roles and assign permissions based on team members' responsibilities.
- **Issue Security:** Set security levels to restrict access to specific issues.
- **Field-Level Security:** Control visibility and editability of custom fields based on user roles.

By leveraging these additional features and concepts, you can maximize the potential of Jira for your project management and agile development needs, making it a comprehensive tool for tracking, planning, and collaborating effectively.

## Account Creation:

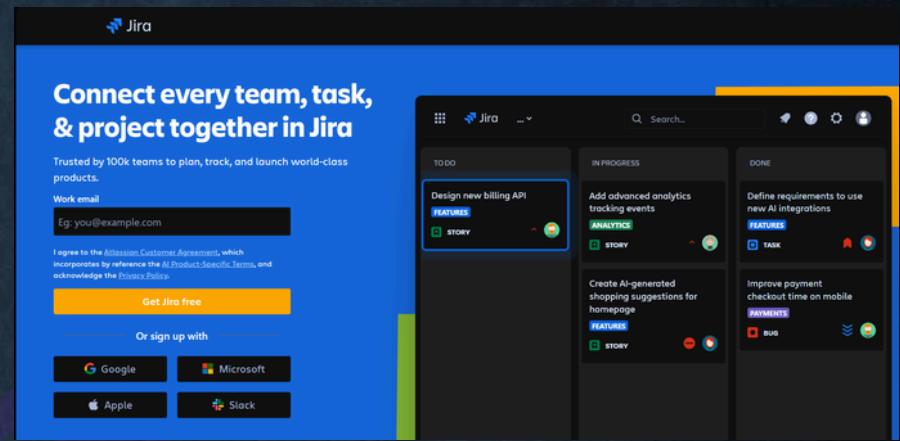
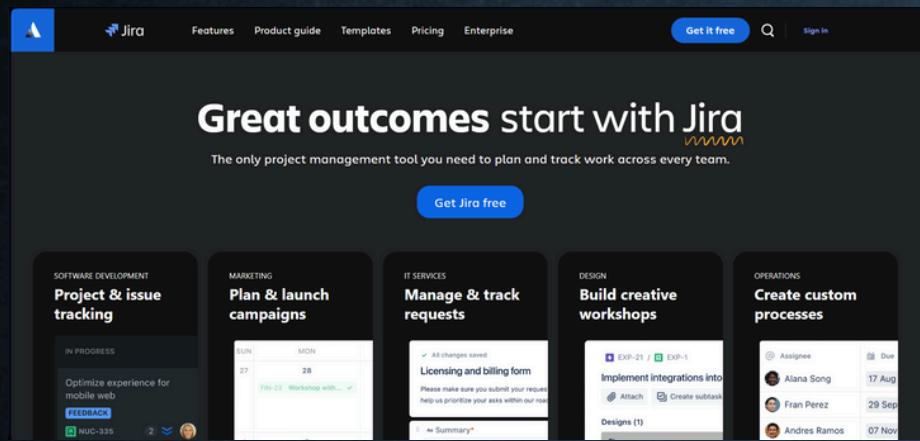
Creating a Jira account is a straightforward process. Here's a step-by-step guide to help you get started:

### 1. Sign Up for Jira

Visit the Jira Website

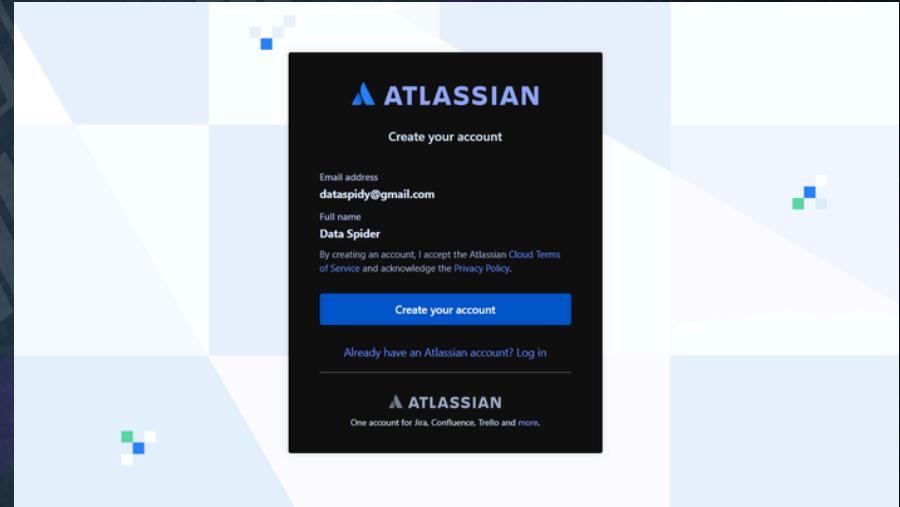
- Step: Open your web browser and go to the Jira website:

<https://www.atlassian.com/software/jira>



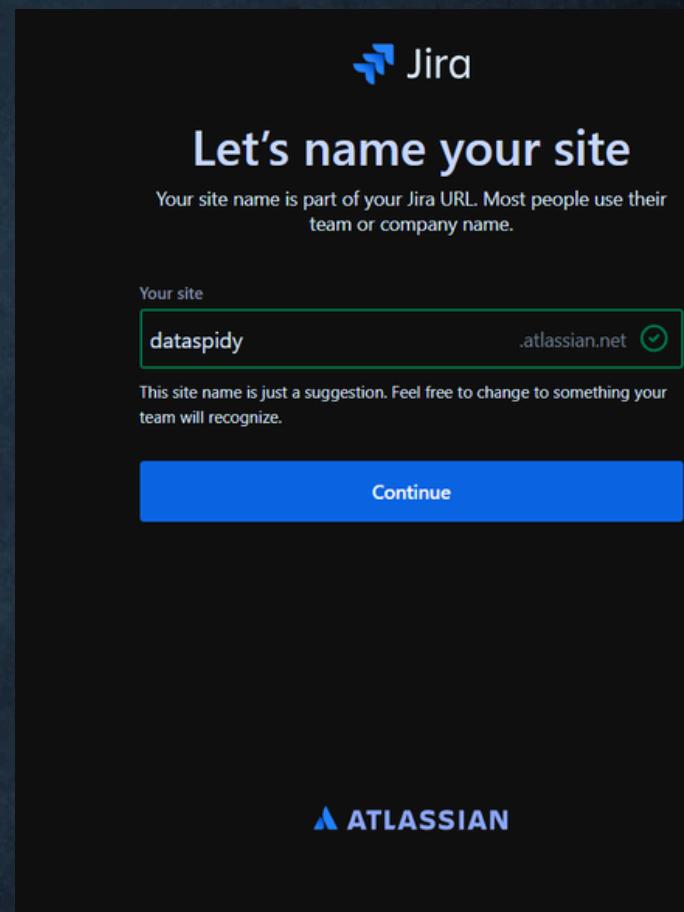
### 2. Create an Atlassian Account

- Step: If you don't already have an Atlassian account, you'll need to create one. Enter your email address and click on Sign up. Follow the prompts to fill in your details and create your account.



### 3. Set Up Your Jira Site

- Step: After creating your Atlassian account, you'll be asked to choose a name for your Jira site. This will be part of your site URL (e.g., <https://yoursitename.atlassian.net>). Enter a unique name and click on Continue.



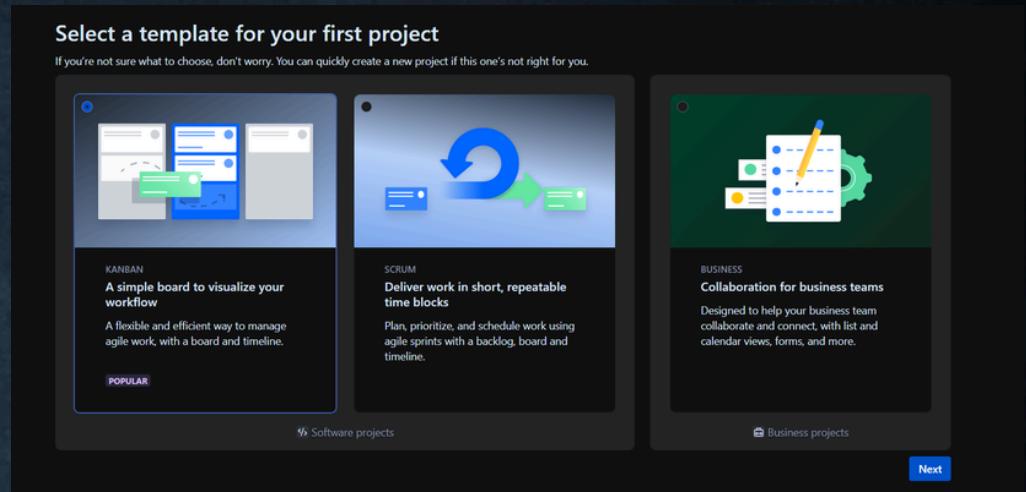
## Initial Configuration

### 1. Choose a Template

- Step: Jira will prompt you to choose a template based on your project type.

For software development, you can choose between Scrum or Kanban.

Select your preferred template and click on Use template.



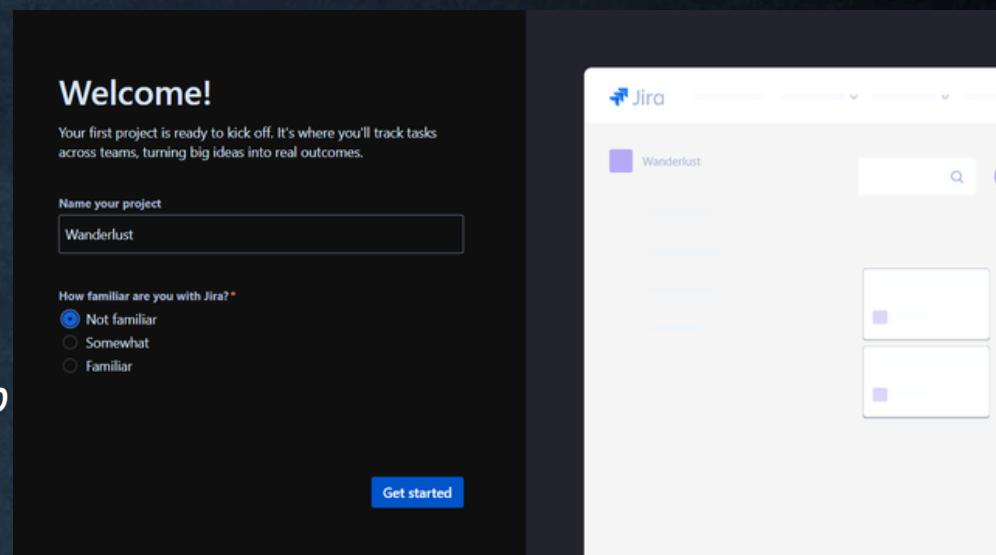
## Example: Using Jira in a Project

1. Assign Tasks: Assign these tasks to team members. For example, the designer gets the "Design Login Screen" task, and the developer gets the "Develop Login Functionality" task.
2. Track Progress: As team members work on their tasks, they update the status in Jira. You can see which tasks are "To Do," "In Progress," "In Review," and "Done."
3. Use Reports: Generate reports to see how the project is progressing. Are most tasks on track? Are there any blockers?
4. Integrate with Other Tools: Connect Jira with GitHub so that when the developer commits code for the login functionality, it automatically updates the task in Jira. Or integrate with Slack to get notifications about task updates.

## 2. Create Your First Project

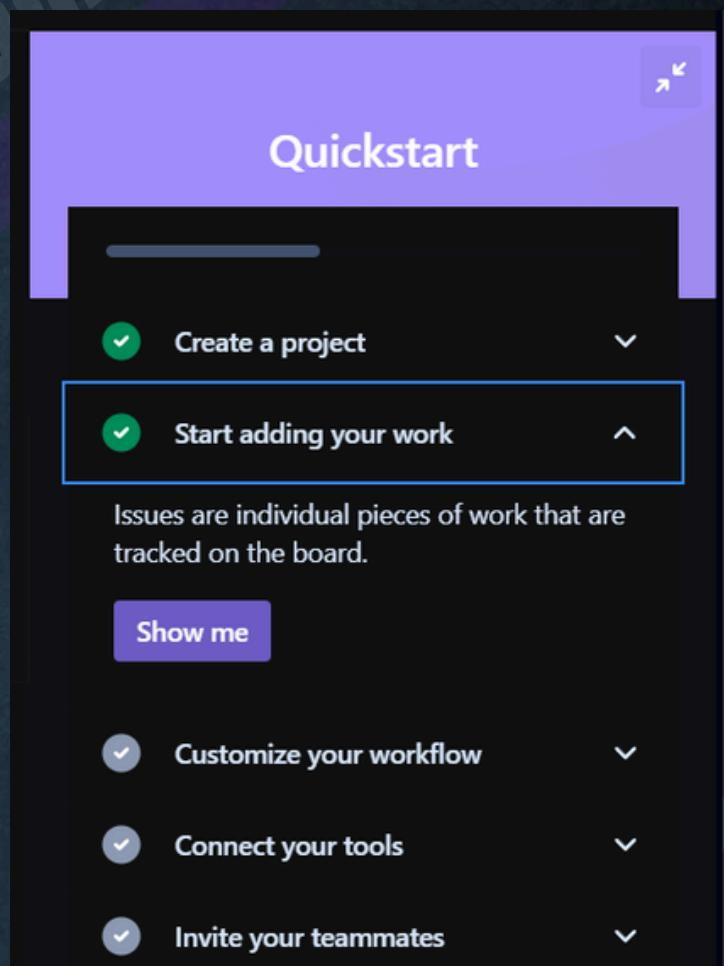
- Step: Enter a name for your project and optionally a key (a unique identifier for your project). Click on Create.

Let's say you're developing a new webapp *Wanderlust*. You create a project in Jira called "**Wanderlust Deployment**".



## 3. Explore Jira

- Step: After setting up your project, take some time to explore the Jira interface. Familiarize yourself with the dashboard, backlog, boards, and other features. Also use QuickStart Panel for better step by step guidance.



#### 4. Edit or Add Columns:

Break down the project into smaller columns, like "STORY", "READY", "TO DO", "IN PROGRESS", "REVIEW" & "DONE".

- **STORY:** High-level tasks or epics.
- **READY:** Tasks that are ready to be picked up.
- **TO DO:** Tasks that need to be started.
- **IN PROGRESS:** Tasks that are currently being worked on.
- **REVIEW:** Tasks that are completed and need review.
- **DONE:** Tasks that are completed and verified.

The screenshot shows the Jira software interface. At the top, there's a navigation bar with links for 'Jira', 'Your work', 'Projects', 'Filters', 'Dashboards', 'Teams', 'Plans', 'Apps', 'Create', 'Upgrade', 'Search', and various icons. On the left, a sidebar lists 'PLANNING' (Timeline, Board, Calendar, List, Issues), 'DEVELOPMENT' (Code, Project pages, Add shortcut), and a note 'You're in a team-managed project'. The main area is titled 'WD board' and shows a Kanban board with three columns: 'STORY', 'READY', and 'TO DO'. A purple callout box above the board says, 'Your issue can be a project task, support ticket, request form, or any other unit of work relevant to your project.' Below the board are buttons for 'Search', 'DS', 'GROUP BY', 'None', 'Import work', 'Insights', 'View settings', and '+ Create issue'. To the right, a 'Quickstart' panel is open, providing steps to 'Create a project', 'Start adding your work' (with a note that 'Issues are individual pieces of work that are tracked on the board'), 'Customize your workflow', 'Connect your tools', 'Invite your teammates', and 'Give feedback'. There are also 'Show me' and 'Dismiss Quickstart' buttons. A large purple arrow graphic is overlaid on the top half of the screen, pointing from the top-left towards the bottom-right.

## Add Tasks in Jira

### STORY

#### 1. Setup and Verify GitHub Repository

- Task: Review existing code and Dockerfile
- Task: Validate Kubernetes YAML files
- Task: Set up GitHub Actions workflow

#### 2. Integrate Jira with GitHub

- Task: Install GitHub for Jira app
- Task: Connect Jira to GitHub repository

The screenshot shows a Jira board titled 'WD board' under the project 'Wanderlust Deployment'. It contains two issues: 'Setup and Verify GitHub Repository' and 'Integrate Jira with GitHub'. Both issues are assigned to 'WD-1'. The 'Setup and Verify GitHub Repository' issue has a checkmark next to it.

The screenshot shows the detailed view of the 'Setup and Verify GitHub Repository' issue. The status is set to 'STORY'. The 'Details' panel shows the following fields:

- Assignee: Unassigned
- Labels: None
- Parent: NEW
- Reporter: Data Spider

Timestamps indicate it was created 8 minutes ago and updated 8 minutes ago. A 'Configure' button is also present.

The screenshot shows the detailed view of the 'Integrate Jira with GitHub' issue. The status is set to 'STORY'. The 'Details' panel shows the following fields:

- Assignee: Unassigned
- Labels: None
- Parent: NEW
- Reporter: Data Spider

Timestamps indicate it was created 6 minutes ago and updated 6 minutes ago. A 'Configure' button is also present.

### READY

Move tasks from STORY to READY once they are broken down and ready to be picked up by the team.

# Task Breakdown and Workflow

## TO DO

### 1. Review Existing Code and Dockerfile

- **Description:** Ensure the Dockerfile and application code are properly set up.
- **Assignee:** [Team Member]

The screenshot shows a Jira issue page for task WD-10. The title is "Review Existing Code and Dockerfile". The "Description" section contains two items: "Description: Ensure the Dockerfile and application code are properly set up." and "Assignee: DataSpider". The "Activity" section shows tabs for "All", "Comments", and "History", with "Comments" selected. A comment input field says "Add a comment...". The "Details" panel on the right shows the assignee as "Data Spider", labels as "None", and a parent issue labeled "NEW". It also shows the creation and update times as "Created 1 minute ago" and "Updated 1 minute ago".

### 2. Validate Kubernetes YAML Files

- **Description:** Review deployment.yaml and service.yaml for correctness.
- **Assignee:** [Team Member]

The screenshot shows a Jira issue page for task WD-11. The title is "Validate Kubernetes YAML Files". The "Description" section contains two items: "Description: Review deployment.yaml and service.yaml for correctness." and "Assignee: DataSpider". The "Activity" section shows tabs for "All", "Comments", and "History", with "Comments" selected. A comment input field says "Add a comment...". The "Details" panel on the right shows the assignee as "Data Spider", labels as "None", and a parent issue labeled "NEW". It also shows the creation and update times as "Created 36 seconds ago" and "Updated 36 seconds ago".

### 3. Set Up GitHub Actions Workflow

- **Description:** Create or verify the GitHub Actions workflow file for deployment.
- **Assignee:** [Team Member]

The screenshot shows a Jira issue page for task WD-12. The title is "Set Up GitHub Actions Workflow". The "Description" section contains the instruction "Description: Create or verify the GitHub Actions workflow file for deployment." and "Assignee: DataSpider". The "Activity" section shows tabs for "All", "Comments", and "History", with "Comments" selected. A comment input field says "Add a comment...". The "Details" panel on the right shows the assignee as "Data Spider", labels as "None", and a parent issue labeled "NEW". It also shows the creation and update times as "Created 17 seconds ago" and "Updated 17 seconds ago".

## 4. Install GitHub for Jira App

- Description: Install and configure the GitHub for Jira app.
- Assignee: [Team Member]

The screenshot shows a Jira issue page with the title 'Install GitHub for Jira App'. The issue key is WD-13. The description field contains the text 'Description: Install and configure the GitHub for Jira app.' and 'Assignee: DataSpider'. The activity section shows a comment from 'DS' with the text 'Add a comment...'. The details panel on the right shows the assignee as 'Data Spider', labels as 'None', parent as 'NEW', and reporter as 'Data Spider'. The issue was created 16 seconds ago and updated 16 seconds ago. A 'Configure' button is visible.

## 5. Connect Jira to GitHub Repository

- Description: Install and configure the GitHub for Jira app.
- Assignee: [Team Member]

The screenshot shows a Jira issue page with the title '5. Connect Jira to GitHub Repository'. The issue key is WD-14. The description field contains the text 'Description: Install and configure the GitHub for Jira app.' and 'Assignee: DataSpider'. The activity section shows a comment from 'DS' with the text 'Add a comment...'. The details panel on the right shows the assignee as 'Data Spider', labels as 'None', parent as 'NEW', and reporter as 'Data Spider'. The issue was created 16 seconds ago and updated 16 seconds ago. A 'Configure' button is visible.

## IN PROGRESS

Move tasks to IN PROGRESS as team members start working on them.

1. Review Existing Code and Dockerfile (IN PROGRESS)

2. Validate Kubernetes YAML Files (IN PROGRESS)

3. Set Up GitHub Actions Workflow (IN PROGRESS)

4. Install GitHub for Jira App (IN PROGRESS)

5. Connect Jira to GitHub Repository (IN PROGRESS)

The screenshot shows a Jira board titled 'WD board' with four columns: STORY 2, READY 1, TO DO 1, and IN PROGRESS 3. The 'IN PROGRESS' column contains three tasks: 'Review Existing Code and Dockerfile' (issue key WD-10), 'Validate Kubernetes YAML Files' (issue key WD-11), and 'Install GitHub for Jira App' (issue key WD-13). The 'Install GitHub for Jira App' task is currently selected. The board also includes a search bar, user filters, and settings buttons at the top.

## REVIEW

Move tasks to REVIEW once they are completed and need to be reviewed.

1. Review Existing Code and Dockerfile (REVIEW)
2. Validate Kubernetes YAML Files (REVIEW)
3. Set Up GitHub Actions Workflow (REVIEW)
4. Install GitHub for Jira App (REVIEW)
5. Connect Jira to GitHub Repository (REVIEW)

## DONE

Move tasks to DONE once they are reviewed and verified.

1. Review Existing Code and Dockerfile (DONE)
2. Validate Kubernetes YAML Files (DONE)
3. Set Up GitHub Actions Workflow (DONE)
4. Install GitHub for Jira App (DONE)
5. Connect Jira to GitHub Repository (DONE)

## Detailed Steps for Each Task

### Task: Review Existing Code and Dockerfile

1. Review the code in the GitHub repository to ensure it is correctly set up for deployment.
2. Verify the Dockerfile to ensure it builds the image correctly.
3. Move the task to REVIEW once completed.

### Task: Validate Kubernetes YAML Files

1. Check the deployment.yaml and service.yaml files in the GitHub repository for correctness.
2. Ensure the configurations match the expected deployment settings.
3. Move the task to REVIEW once completed.

## Task: Set Up GitHub Actions Workflow

### 1. Verify the existing GitHub Actions workflow file or create a new one if not present.

- Example of a workflow file (.github/workflows/deploy.yaml):

```
name: Deploy Wanderlust Web App

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v2

      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v1

      - name: Login to Docker Hub
        uses: docker/login-action@v1
        with:
          username: ${{ secrets.DOCKER_USERNAME }}
          password: ${{ secrets.DOCKER_PASSWORD }}

      - name: Build and push Docker image
        run:
          docker build -t your-dockerhub-username/wanderlust-web-app:v1 .
          docker push your-dockerhub-username/wanderlust-web-app:v1

      - name: Set up Kubernetes
        uses: azure/setup-kubectl@v1
        with:
          version: v1.19.3

      - name: Deploy to Kubernetes
        run:
          kubectl apply -f k8s/deployment.yaml
          kubectl apply -f k8s/service.yaml
    env:
      KUBE_CONFIG_DATA: ${{ secrets.KUBE_CONFIG_DATA }}
```

Projects / Wanderlust Deployment

WD board

The screenshot shows a Jira board titled 'WD board' with the following columns:

- STORY 2**: Contains tasks 'Setup and Verify GitHub Repository' (status: WD-1) and 'Integrate Jira with GitHub' (status: WD-7).
- READY 1**: Contains task '5. Connect Jira to GitHub Repository' (status: WD-14).
- TO DO 1**: Contains task 'Validate Kubernetes YAML ...' (status: WD-11). This card is highlighted with a blue border and has a 'Create issue' button below it.
- IN PROGRESS 2**: Contains tasks 'Review Existing Code and Dockerfile' (status: WD-10) and 'Install GitHub for Jira App' (status: WD-13).
- REVIEW 1**: Contains task 'Set Up GitHub Actions Workflow' (status: WD-12).
- DONE**: An empty column.

At the top, there are navigation icons for search, user, and settings, along with buttons for 'Import work', 'Insights', and 'View settings'. The 'GROUP BY' dropdown is set to 'None'.

# Task: Install GitHub for Jira App

1. Go to Jira settings.

2. Select Apps > Find new apps.

3. Search for GitHub for Jira and click Get it now.

4. Follow the prompts to install the app.

5. Move the task to REVIEW once completed.

Add to Jira



GitHub for Jira

by Atlassian

2.7/4 ★★★★☆ 420

149,441 installs

CLOUD SECURITY PARTICIPANT FREE

GitHub for Jira will perform the following actions:

- Delete data from the host application
- Write data to the host application
- Read data from the host application

App Info

Get it now Cancel

Jira / Marketplace apps

Discover apps and integrations for Jira

Sort ▾ Free for all teams More Filters

51 results

**GitHub for Jira**  
Integrate GitHub with Jira for smooth development workflow by tracking branches, PRs, commits, builds & deployments within... Deployments, Integrations, Source code, Workflow  
2.7/4 ★★★★☆ 420 149.4k installs CLOUD SECURITY PARTICIPANT

**Github Links for Jira**  
Jira GitHub integration simplifies development, reduces context switching. Start FREE with GitHub for Jira Integrations, Repository connectors, Source code, W...  
1.9/4 ★★★★☆ 7 988 installs CLOUD FORTIFIED

GitHub for Jira installed successfully

Your app has been added to your instance and is ready for use.

Get started · Manage app

Connect GitHub to Jira

Before you start, you should have:

- A GitHub account
- Owner permission for a GitHub organization

Learn how to check GitHub permissions

Continue →

Select your GitHub product

GitHub Cloud GitHub Enterprise Server

How do I check my GitHub product?

Next

Jira by Atlassian would like permission to:

Verify your GitHub identity (udaybambal)

Know which resources you can access

Act on your behalf

Learn more about Jira

Cancel Authorize Jira

Authorizing will redirect to <https://github.atlassian.com>

Install on your personal account Uday Bambal

for these repositories:

All repositories This applies to all current and future repositories owned by the resource owner. Also includes public repositories (read-only).

Only select repositories Select at least one repository. Also includes public repositories (read-only).

with these permissions:

Read access to Dependabot alerts, actions, metadata, secret scanning alerts, and security events

Read and write access to code, deployments, issues, and pull requests

Install Cancel

Next: you'll be directed to the GitHub App's site to complete setup.

uyaybambal is now connected!

Your team needs to add issue keys in GitHub To import development work into Jira and track it in your issues, add issue keys to branches, pull request titles, and commit messages.

Add another organization How to add issue keys

Exit set up

## Task: Install GitHub for Jira App

1. Go to Jira settings.
2. Select Apps > Find new apps.
3. Search for GitHub for Jira and click Get it now.
4. Follow the prompts to install the app.
5. Move the task to REVIEW once completed.

## Task: Connect Jira to GitHub Repository

1. Go to Apps > Manage your apps.
2. Find GitHub for Jira and click on Configure.
3. Follow the instructions to link your GitHub account and select the repository for the Wanderlust web app.
4. Move the task to REVIEW once completed.

The screenshot shows the Jira interface with the 'Manage apps' section open. The 'GitHub for Jira' app is listed under 'User-installed apps'. The app card includes a 'Get started' button, a screenshot showing 'Streamline GitHub connections', and details like Version 1.143-AC, Vendor Atlassian, and Support Supported by Atlassian. To the right, there are links for 'Watch app', 'Marketplace listing', 'Documentation', 'EULA', 'Data security and privacy', and 'Support and issues'. Below this, the 'GitHub configuration' section is shown, featuring a summary of GitHub integration status and a 'Connect a GitHub organization' button.

Connected organization: udaybambal

Repository access: Only select repos 0 / 1

Backfill status: IN PROGRESS

Settings

Connect a GitHub organization

The screenshot shows the Jira 'GitHub configuration' page. At the top, there's a banner about connecting GitHub to Jira for viewing development activity and sending data. Below it, a table provides details about the connected organization ('udaybambal'), repository access ('Only select repos'), backfill status ('FINISHED Backfilled from: 7/18/24'), and settings. The left sidebar lists various Jira Marketplace apps like 'Find new apps', 'Manage apps', 'App requests', 'Promotions', and 'OAuth credentials'. A 'GITHUB FOR JIRA' section includes a 'Configure' link.

## Monitor and Manage the Project

- *Daily Standups: Regularly update the status of tasks.*
- *Review Sessions: Regularly review tasks moved to the REVIEW column.*
- *Retrospectives: After project completion, review what went well and what could be improved.*

The screenshot shows the Jira 'WD board' for the 'Wanderlust Deployment' project. The board has columns for STORY, READY, TO DO, IN PROGRESS, REVIEW, and DONE. Tasks are listed with their status (e.g., 'Setup and Verify GitHub Repository' is in STORY 2). The left sidebar includes sections for PLANNING (Timeline, Board, Calendar), DEVELOPMENT (Code, Project pages), and general project management (Issues, Add view, Add shortcut, Project settings).

*By following this detailed plan, you ensure a structured approach to managing the deployment of the Wanderlust web app using Kubernetes, integrated with Jira and GitHub for effective project management and continuous deployment.*