Lab 7: Import existing scripted pipeline into Blue Ocean and visualize

Step 1 — Install Blue Ocean Plugin

1. **Login to Jenkins** → `Manage Jenkins → Plugins → Available Plugins`.
2. Search for **Blue Ocean**.
3. Select and install:
   - `blueocean` (main plugin)
   - It will auto-install required dependencies (e.g., `pipeline-stage-view`, `pipeline-rest-api`).
4. Restart Jenkins after installation.

✅ Verify installation:

- You should see a **Blue Ocean** link on the left sidebar in Jenkins Dashboard.

Step 2 — Import Scripted Pipeline

We'll use the **Forex Converter pipeline (Lab 6 corrected Jenkinsfile)**.

1. Push your project with `Jenkinsfile` into a Git repository (GitHub, GitLab, Bitbucket).
2. In Jenkins dashboard:
   - Click **Blue Ocean** → `Create Pipeline`.
   - Choose **GitHub (or Git)**.
   - Provide repo URL & credentials.
3. Jenkins will detect the `Jenkinsfile`.
4. Import it → Pipeline shows up in Blue Ocean UI.

Step 3 — Visualize Pipeline Execution

1. Open **Blue Ocean** → select your pipeline.
2. Run the pipeline.
3. Observe:
   - **Stages displayed as visual nodes** (`Checkout`, `Build`, `Unit Test`, `Integration Test`, `Package`, `Deploy`).

- If a stage fails → the node turns **red**, and you can click it to expand logs.
- Parallel test execution (if configured in Maven) will show **branches** in the UI.
- You can re-run failed stages without re-running entire pipeline.

Step 4 — Debugging with Blue Ocean

- Example: If the `Unit Test` stage fails, you'll see a **red node** → expand → logs show which test failed.
- Example: If artifact not archived, you'll see **empty Package stage** → easier to detect missing paths.
- Example: If Deploy fails, Blue Ocean highlights **only Deploy node red** (instead of scrolling raw logs).

Step 5 — Optional Advanced Exercises

1. **Introduce a Failure**
   - Break artifact path in Jenkinsfile (`**/build/*.jar`).
   - Re-run in Blue Ocean → watch how the visualization pinpoints Package stage failure.
2. **Add Parallel Stage**
   Modify Jenkinsfile:

```
stage('Tests') {
    parallel(
        "Unit Tests": { sh 'mvn test' },
        "Integration Tests": { sh 'mvn verify -P integration-tests' }
    )
}

```

   ✅ Observe: Blue Ocean shows **parallel branches**.
3. **Stage Replay**
   - Fail `Integration Test` stage.
   - Replay only that stage in Blue Ocean → confirm fix.
4. **Multi-branch Pipeline View**
   - Create `feature/` branch in repo.
   - Push different Jenkinsfile.
   - Blue Ocean shows **multi-branch visualization**.

Step 6 — Expected Learning

- Teams learn to **move from raw Jenkins console logs → visual CI/CD insights**.
- Debugging becomes faster (stage-by-stage view instead of full log scrubbing).
- Understand **parallelism & multi-branch pipelines** via visualization.
- Build habit of **observing pipeline health over time** using Blue Ocean UI.