

Using Matrix Strategy in Jenkins

Introduction

The Matrix Strategy in Jenkins allows running the same pipeline with different parameters or configurations in parallel. This is particularly useful for testing across multiple environments, operating systems, or tool versions.

Benefits of Matrix Strategy

- **Parallel Execution:** Reduces build time by executing multiple configurations simultaneously.
- **Scalability:** Supports various test environments without duplicating pipeline definitions.
- **Better Test Coverage:** Ensures that software runs smoothly across different configurations.

Example: Matrix Strategy in a Declarative Pipeline

```
pipeline {  
    agent any  
    matrix {  
        axes {  
            axis {  
                name 'OS'  
                values 'ubuntu', 'windows', 'macos'  
            }  
            axis {  
                name 'JDK'  
                values '11', '17'  
            }  
        }  
    }  
}
```

```
stages {  
  stage('Build & Test') {  
    steps {  
      echo "Building on ${OS} with JDK ${JDK}"  
      sh 'echo Running tests'  
    }  
  }  
}
```

Best Practices for Matrix Strategy

- **Limit Combinations:** Avoid excessive parallel executions to prevent performance issues.
- **Use Labels for Agents:** Assign labels to agents for better resource allocation.
- **Optimize Step Execution:** Only run necessary steps for each configuration.

Conclusion

Using the Matrix Strategy in Jenkins improves efficiency and test coverage by running parallel builds for different configurations. It ensures that applications work across multiple environments, enhancing software reliability.

Reference links

<https://www.jenkins.io/blog/2019/11/22/welcome-to-the-matrix/>