CI/CD

Question: What are the key components of Continuous Integration?

CI/CD

Continuous Integration(CI) is a critical practice in DevOps that promotes collaboration and early detection of integration issues

CI/CD

1. Version Control System

CI/CD

2. Build Automation

CI/CD

2. Build Automation

Benefits

1. Efficiency
2. Consistency
3. Early Detection
4. Faster Feedback
5. Integration with pipelines
6. Scalability
7. Traceability

CI/CD

3. Automated Testing

CI/CD

3. Automated Testing

Benefits

1. Early Bug Detection
2. Rapid Feedback
3. Regression Testing
4. Test Coverage
5. Continuous Integration and Deployment
6. Faster Time-To-Market
7. Continuous Improvement

CI/CD

4. Artifact Repository

CI/CD

5. Notifications and Reporting

CI/CD

Question: Explain any 5 Jenkins Plugins that you normally use?

CI/CD

1. Git Plugin

- Clone, Fetch, Checkout source code
- Tagging & Branching
- Build triggers based on Git code changes

CI/CD

2. Maven Integration Plugin

- Building Java projects
- Auto Build Triggers
- Publish Maven build artifacts

CI/CD

3. Docker Plugin

- Building Docker Containers
- Integrates with Docker Hub
- Docker-Compose support

CI/CD

4. Email Extension Plugin

- Email Notifications
- Custom Email Content
- Attach Build Artifacts

CI/CD

5. JUnit Plugin

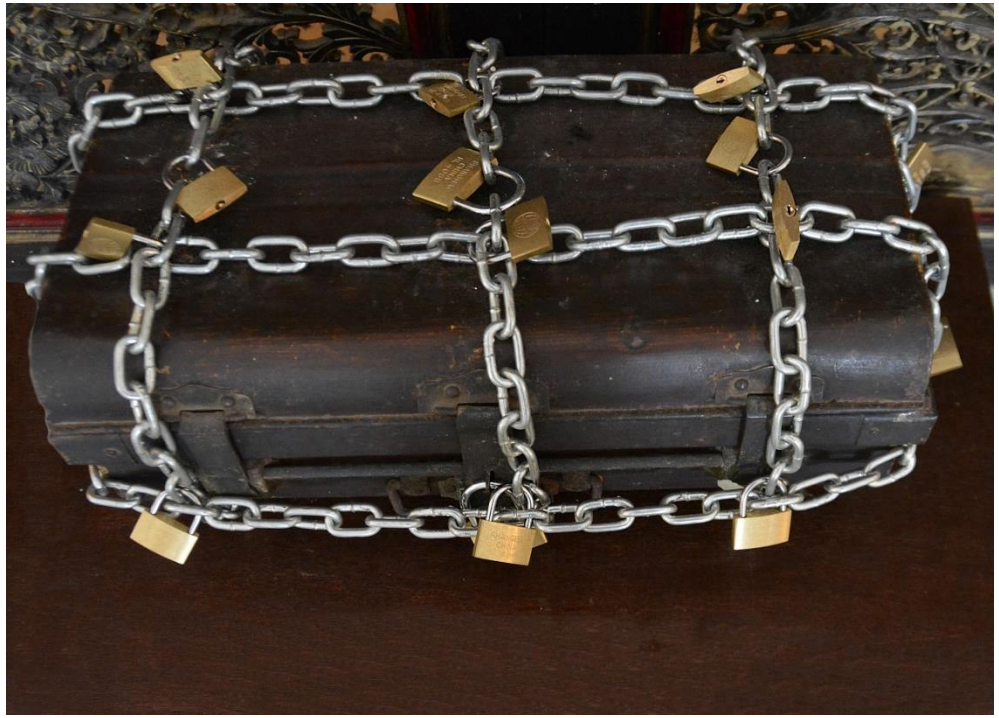
- Parses Test Results
- Test Trends and Statistics
- Build Status Thresholds

CI/CD

Question: Explain the concept of ‘Build as an immutable artifact’?

CI/CD

Immutable artifact



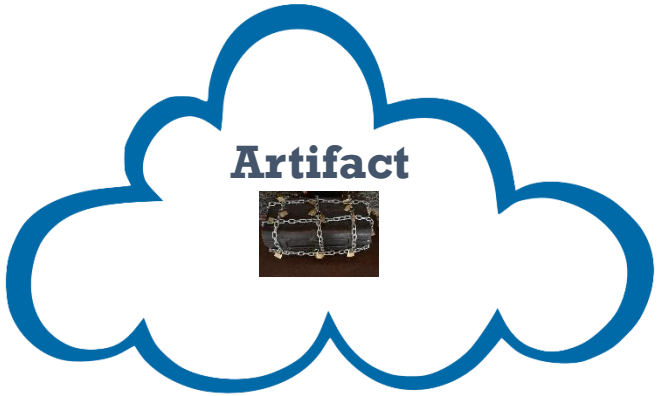
CI/CD

Why Immutable artifacts

- Reproducibility
- Portability
- Versioning
- Rollback and Rollforward

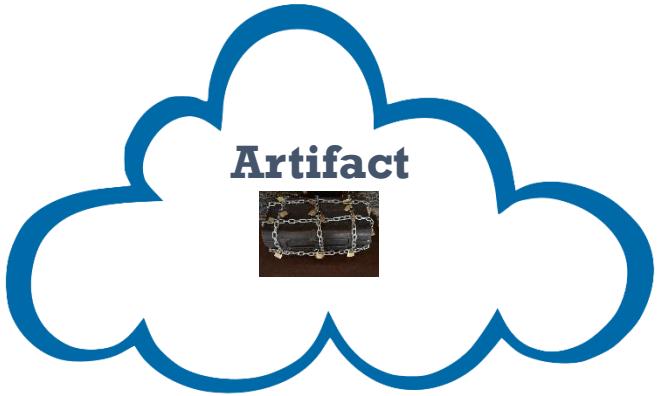
CI/CD

Development Environment



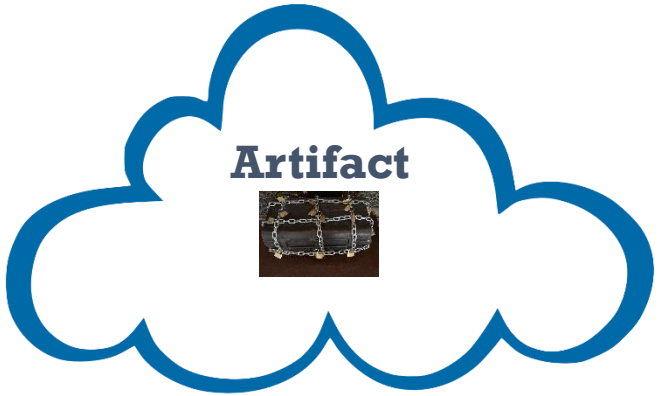
CI/CD

Testing Environment



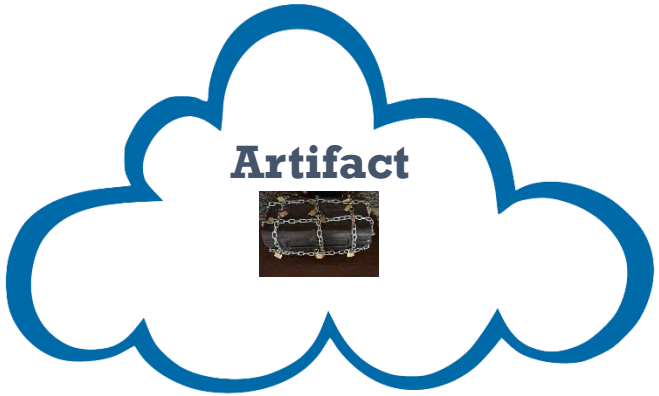
CI/CD

Staging Environment

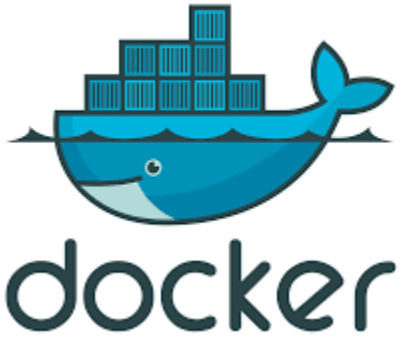


CI/CD

Production Environment



CI/CD



Artifact

CI/CD

Jenkins

Artifact

CI/CD

Artifact

CI/CD

Artifact

CI/CD

Question:

Define a Jenkins job and explain its components, including build steps, post-build actions, and triggers.

CI/CD

Build Steps

- Source code checkout
- Build and Package
- Static Code Analysis
- Containerization

CI/CD

Post Build Actions

- Artifact Archiving
- Email Notification
- Deploy to Staging

CI/CD

Triggers

- SCM Polling
- Nightly Build Schedule
- Manual Trigger

CI/CD

Question: Continuous Integration Vs Continuous Delivery Vs Continuous Deployment

CI/CD

Continuous Integration

- Integration of code changes
- Regular code commits
- Prevent integration issues
- Catch integration issues
- Code quality

CI/CD

Continuous Delivery

- Automate software release
- Build, Test and Deploy
- Consistent releases
- Prep the environment
- Production Ready

CI/CD

Continuous Deployment

- Automate software release to Production
- Eliminate manual intervention
- Robust Testing
- Frequent releases

CI/CD

Key Differences

	Continuous Integration	Continuous Delivery	Continuous Deployment
Scope	Code Integration	Release Preparation	Entire Release Process
Manual Intervention	Limited	Prod – Manual Non Prod - Automated	Automated Release to Production
Frequency	Frequent Code Commits	Frequent Release to Non Production	Frequent Release to Production
Risk and Control	Early Detection of Issues	Control over Release Process	Robust Automated Testing
Business Readiness	Code Integration only	Production Ready Deployments	Automated Production Release

CI/CD

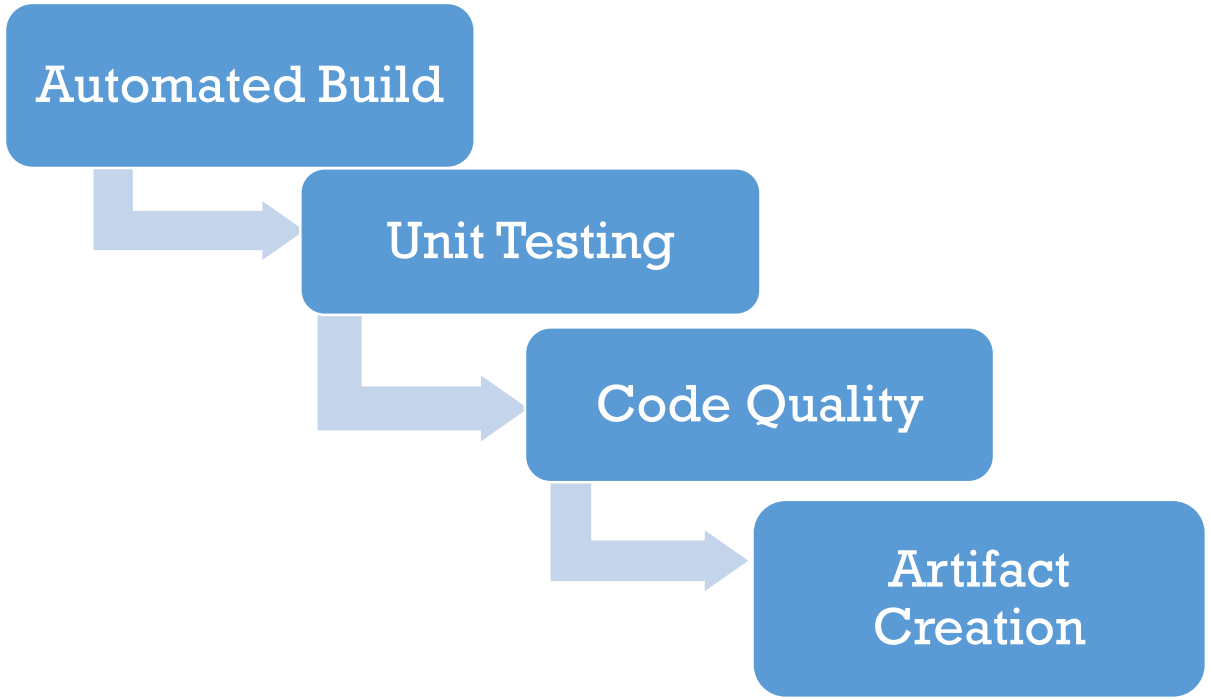
Question: Explain the CI/CD pipeline?

CI/CD

1. Code Development

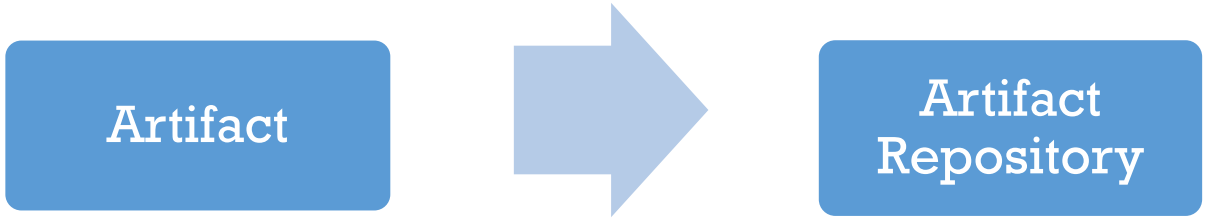
CI/CD

2. Continuous Integration



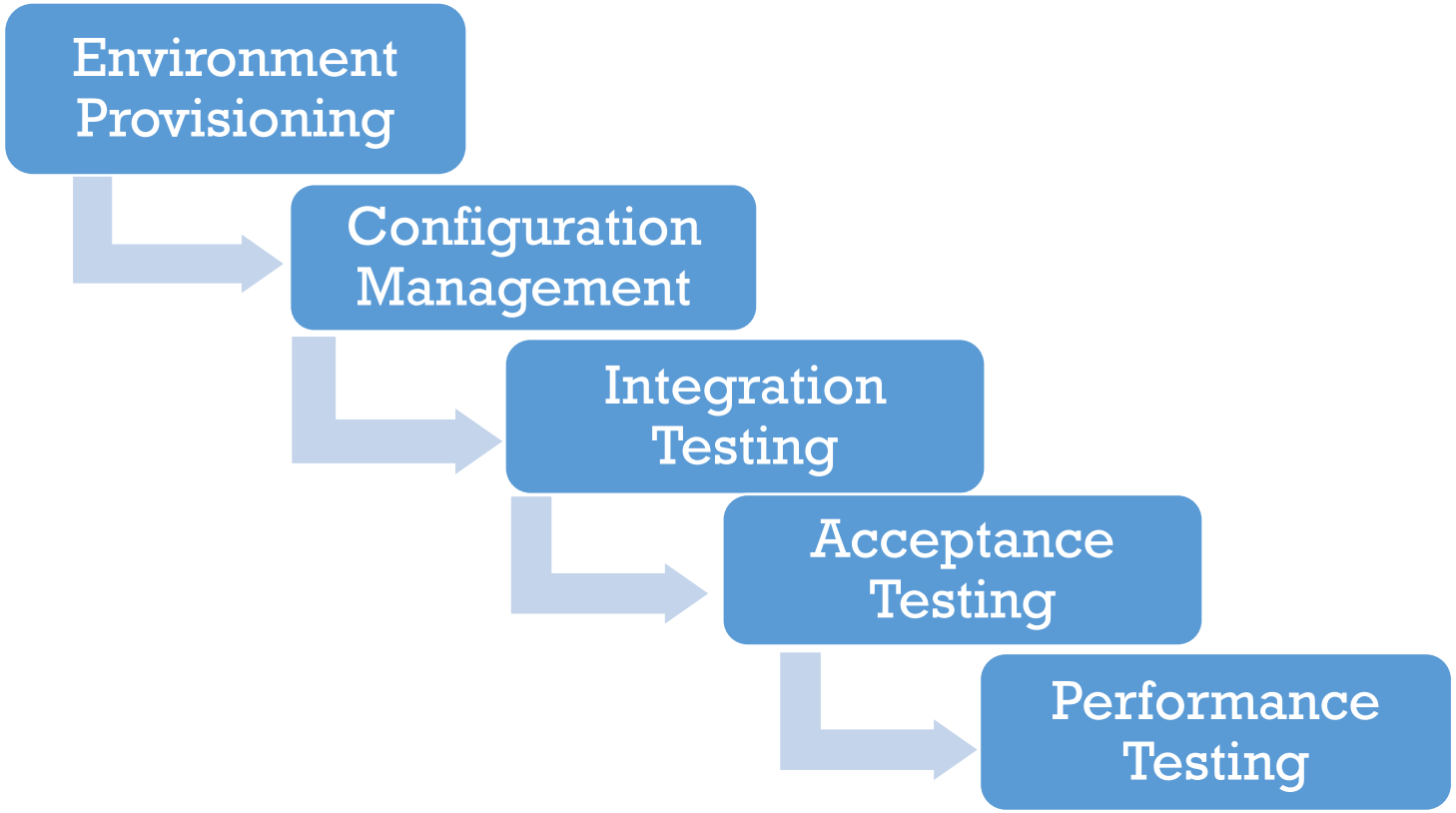
CI/CD

3. Artifact Storage



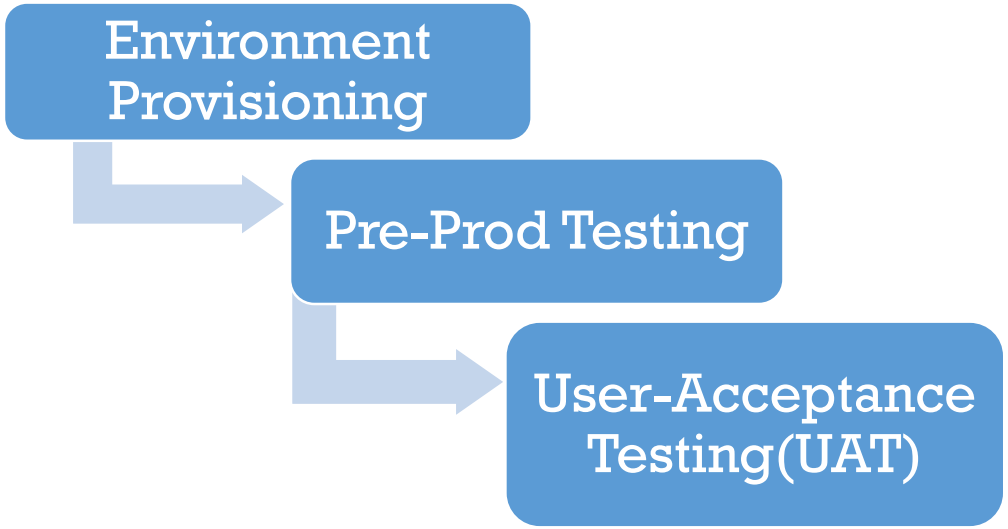
CI/CD

4. Automated Deployment to Testing Environment



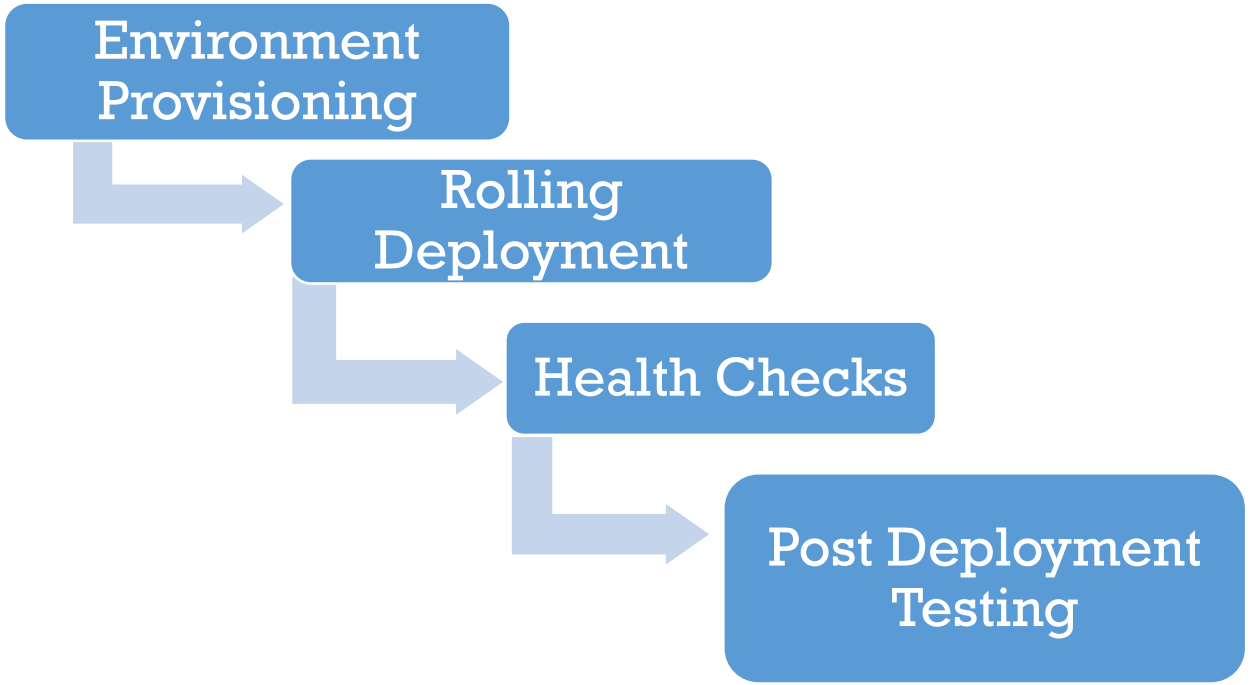
CI/CD

5. Automated Deployment to Staging Environment



CI/CD

6. Deployment to Production



CI/CD

7. Continuous Monitoring

CI/CD

Question: Describe the stages involved in Continuous Delivery?

CI/CD

Continuous Delivery

1. Source Code Management
2. Build and Compile
3. Automated Testing
4. Code Quality and Analysis
5. Artifact Repository
6. Deployment and Env Provisioning
7. Automated Testing
8. Approval and Manual Testing
9. Release to Production
10. Monitoring and Feedback

CI/CD

Question: What are some common challenges and best practices to implement Continuous Integration in your Organization/Team?

CI/CD

Challenges

- Cultural Transformation
- Legacy Systems
- Test Automation
- Infrastructure Management
- Release Coordination

CI/CD

Best Practices

- Start with a Pilot Project
- Cross-Functional Collaboration
- Test Automation
- Infrastructure Management
- Continuous Feedback
- Incremental Deployment and Feature Flags
- Version Control and Release Branching
- Continuous Learning and Improvement

Sooraj Mohammed

CI/CD

Question:

Outline the steps you would take to identify and resolve issues with failed Jenkins build?

CI/CD

- Check Console Output
- Examine Build Log
- Check Build Environment
- Review Source Code Changes
- Check Plugins Versions
- Reproduce Locally
- Isolate Failing Test Cases
- Utilize Jenkins Plugins for Analysis
- Check Jenkins Server Logs
- Collaborate with Dev Teams
- Implement Corrective Measures
- Run Incremental Builds
- Document Findings and Resolutions
- Continuous Monitoring