Introduction to Jenkins

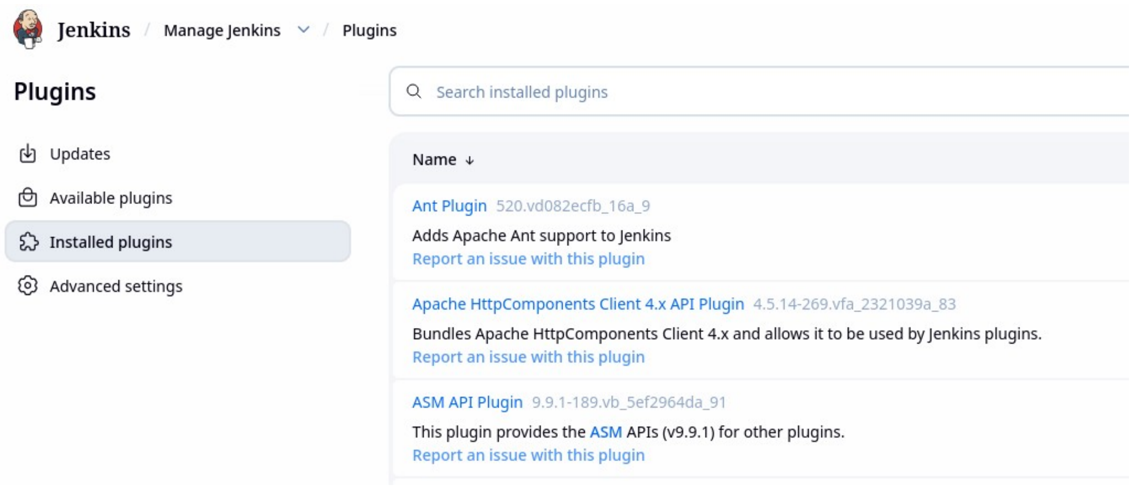
Module 8: Jenkins Plugins

Topics

- Core and recommended plugins
- Code quality (SonarQube)
- Security and dependency scanning
- Cloud, Docker, and Kubernetes integrations

Jenkins Core vs Plugins

- Jenkins is intentionally designed with
  - A small, stable core
  - Most functionality delivered through plugins
- Examples of features provided by plugins
  - SCM integration (Git, GitHub, GitLab)
  - Pipelines and DSLs
  - Credentials handling
  - Cloud and container agents
  - Code quality and security scanning
  - UI extensions



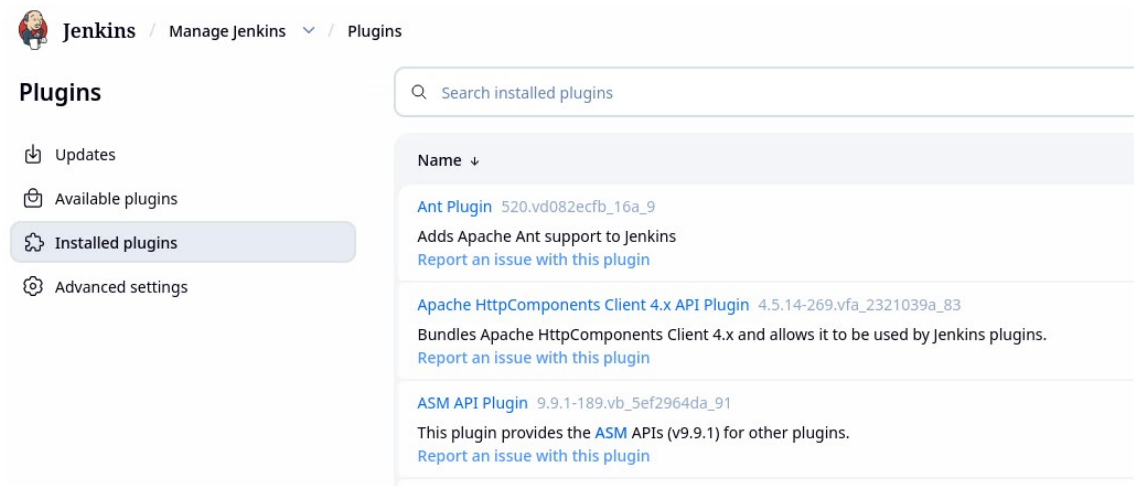
Ant Plugin 520.vd082ecfb_16a_9
Adds Apache Ant support to Jenkins
Report an issue with this plugin

Apache HttpComponents Client 4.x API Plugin 4.5.14-269.vfa_2321039a_83
Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins.
Report an issue with this plugin

ASM API Plugin 9.9.1-189.vb_5ef2964da_91
This plugin provides the ASM APIs (v9.9.1) for other plugins.
Report an issue with this plugin

Plugin Architecture

- Plugins are written in Java
  - Loaded into the Jenkins controller JVM
- Extend Jenkins by providing
  - New pipeline steps
  - New job types
  - New UI screens
  - New APIs
- Runtime behavior
  - Plugins run inside Jenkins
  - Share the same memory and process
  - Have deep access to Jenkins internals



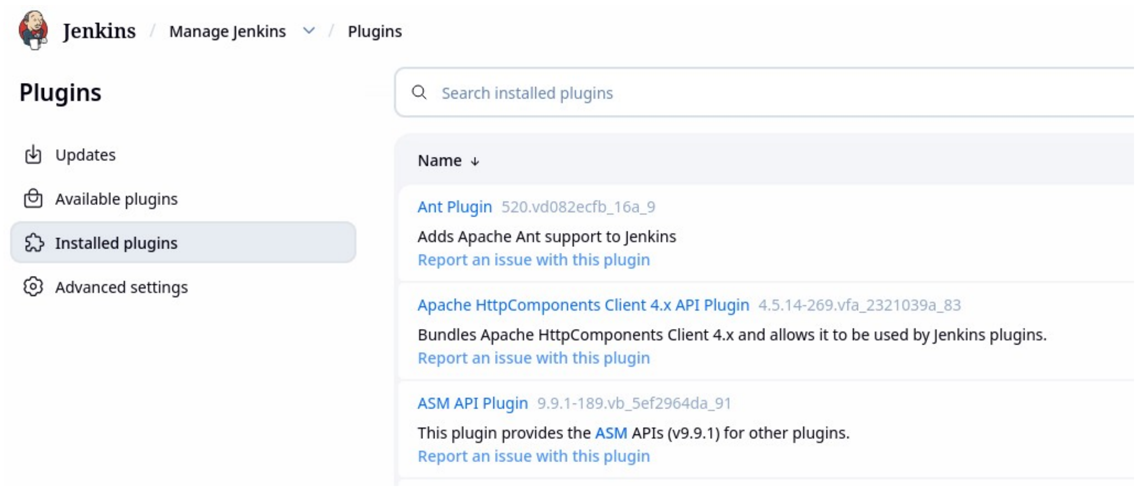
Ant Plugin 520.vd082ecfb_16a_9
Adds Apache Ant support to Jenkins
Report an issue with this plugin

Apache HttpComponents Client 4.x API Plugin 4.5.14-269.vfa_2321039a_83
Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins.
Report an issue with this plugin

ASM API Plugin 9.9.1-189.vb_5ef2964da_91
This plugin provides the ASM APIs (v9.9.1) for other plugins.
Report an issue with this plugin

Plugin Lifecycle

- Plugin lifecycle
  - Add the plugin to the configuration
    - *Only done once*
  - Jenkins loads plugin at startup
  - Plugin adds features
  - Plugin is updated or removed as required
- Some plugins require
  - Jenkins restart
  - Plugin dependency updates



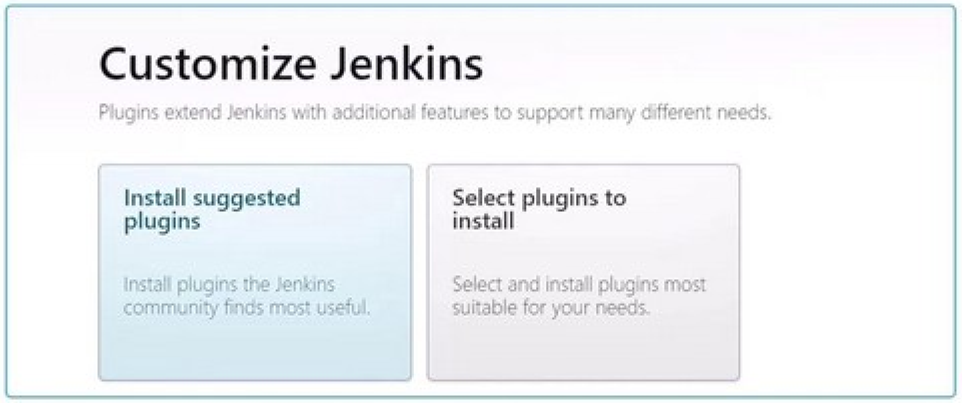
Ant Plugin 520.vd082ecfb_16a_9
Adds Apache Ant support to Jenkins
Report an issue with this plugin

Apache HttpComponents Client 4.x API Plugin 4.5.14-269.vfa_2321039a_83
Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins.
Report an issue with this plugin

ASM API Plugin 9.9.1-189.vb_5ef2964da_91
This plugin provides the ASM APIs (v9.9.1) for other plugins.
Report an issue with this plugin

Plugin Update Management

- If using Jenkins LTS as a production system
  - Update plugins regularly, not randomly
  - Test plugin updates in non-production Jenkins instances
  - Update plugins in controlled batches
  - Avoid auto-updating in production



Customize Jenkins

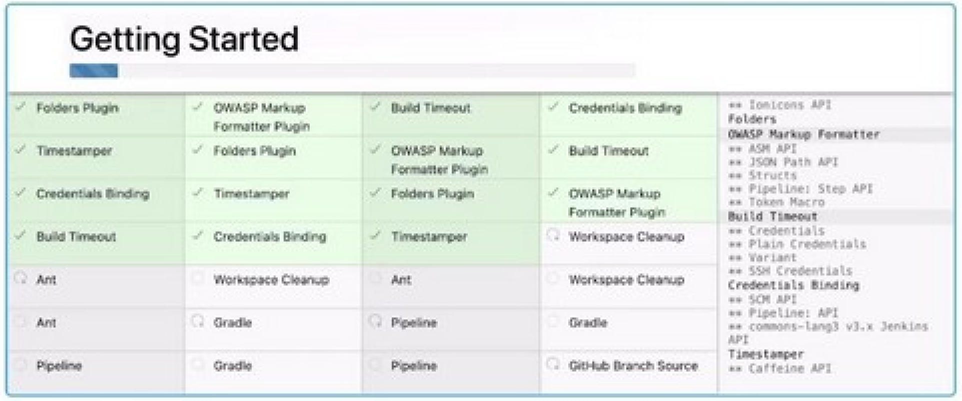
Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.



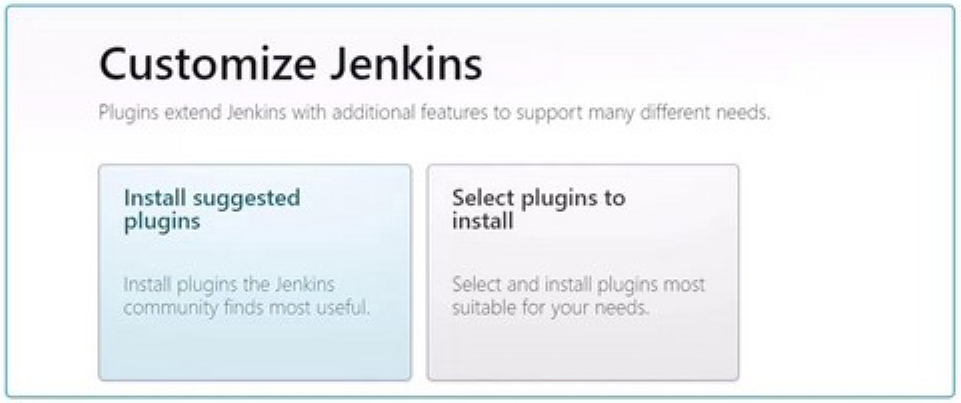
Getting Started

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout
✓ Credentials Binding	✓ Timestamper	✓ Folders Plugin	✓ OWASP Markup Formatter Plugin
✓ Build Timeout	✓ Credentials Binding	✓ Timestamper	Workspace Cleanup
Ant	Workspace Cleanup	Ant	Workspace Cleanup
Ant	Gradle	Pipeline	Gradle
Pipeline	Gradle	Pipeline	GitHub Branch Source

```
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** Pipeline: Step API
** Token Macro
Build Timeout
** Credentials
** Plain Credentials
** Variant
** SSH Credentials
Credentials Binding
** SCM API
** Pipeline: API
** commons-lang3 v3.x Jenkins API
Timestamper
** Caffeine API
```

Core and Recommended Plugins

- Jenkins core provides basic functionality
- Includes
  - Job scheduling
  - UI framework
  - Security foundation
- Core is intentionally minimal.



Customize Jenkins

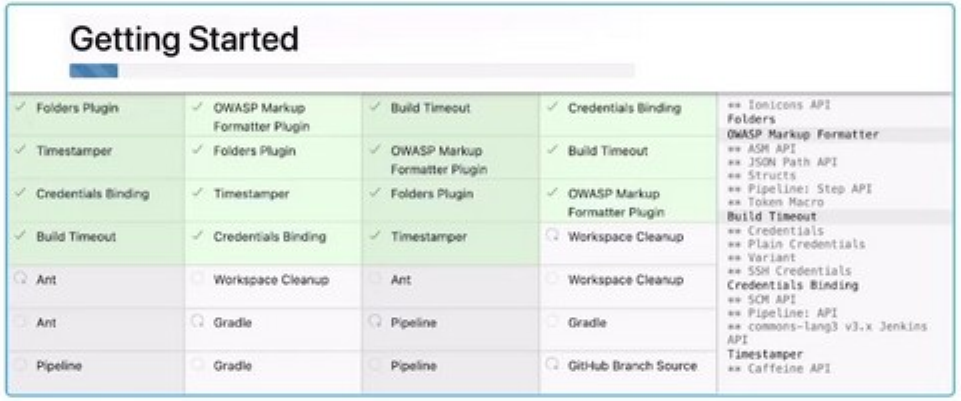
Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.



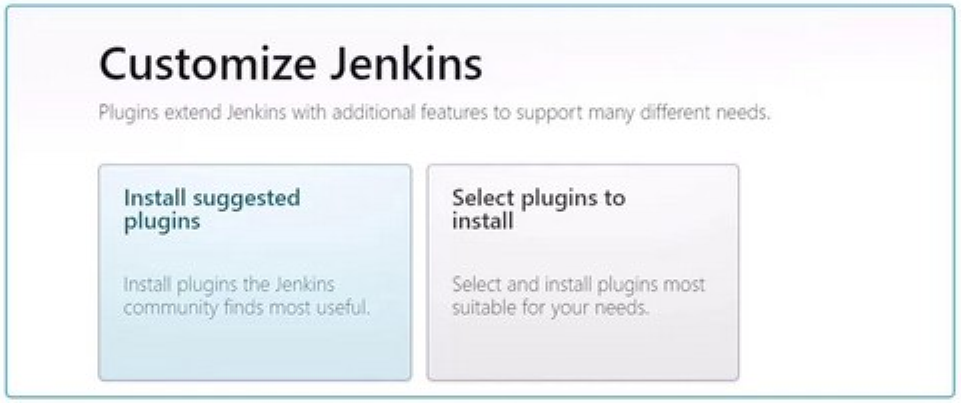
Getting Started

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout
✓ Credentials Binding	✓ Timestamper	✓ Folders Plugin	✓ OWASP Markup Formatter Plugin
✓ Build Timeout	✓ Credentials Binding	✓ Timestamper	Workspace Cleanup
Ant	Workspace Cleanup	Ant	Workspace Cleanup
Ant	Gradle	Pipeline	Gradle
Pipeline	Gradle	Pipeline	GitHub Branch Source

```
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** Pipeline: Step API
** Token Macro
Build Timeout
** Credentials
** Plain Credentials
** Variant
** SSH Credentials
Credentials Binding
** SCM API
** Pipeline: API
** commons-lang3 v3.x Jenkins
API
Timestamper
** Caffeine API
```

Core and Recommended Plugins

- During initial setup, Jenkins offers "Suggested plugins"
  - These typically include
    - *Pipeline*
    - *Git*
    - *Credentials*
    - *Matrix authorization*
    - *Workspace cleanup*
- Plugin best practices
  - Install only what you need
  - Remove unused plugins
  - Prefer well-maintained plugins
  - Check update activity and issue history



Customize Jenkins

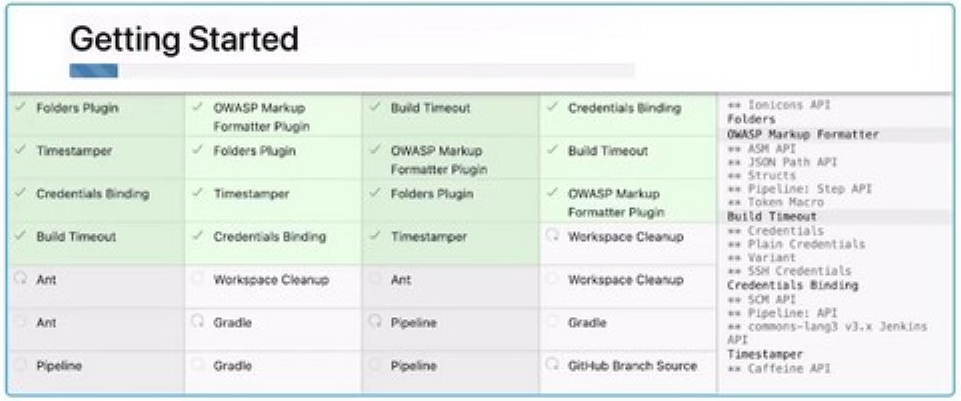
Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.



Getting Started

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout
✓ Credentials Binding	✓ Timestamper	✓ Folders Plugin	✓ OWASP Markup Formatter Plugin
✓ Build Timeout	✓ Credentials Binding	✓ Timestamper	Workspace Cleanup
Ant	Workspace Cleanup	Ant	Workspace Cleanup
Ant	Gradle	Pipeline	Gradle
Pipeline	Gradle	Pipeline	GitHub Branch Source

```
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** Pipeline: Step API
** Token Macro
Build Timeout
** Credentials
** Plain Credentials
** Variant
** SSH Credentials
Credentials Binding
** SCM API
** Pipeline: API
** commons-lang3 v3.x Jenkins
API
Timestamper
** Caffeine API
```

Code Quality Plugins (SonarQube)

- The SonarQube plugin enables
  - Static code analysis
  - Code smell detection
  - Coverage reporting
  - Quality gates
- Pipeline flow
  - Build code
  - Run tests
  - Run Sonar analysis
  - Evaluate quality gate

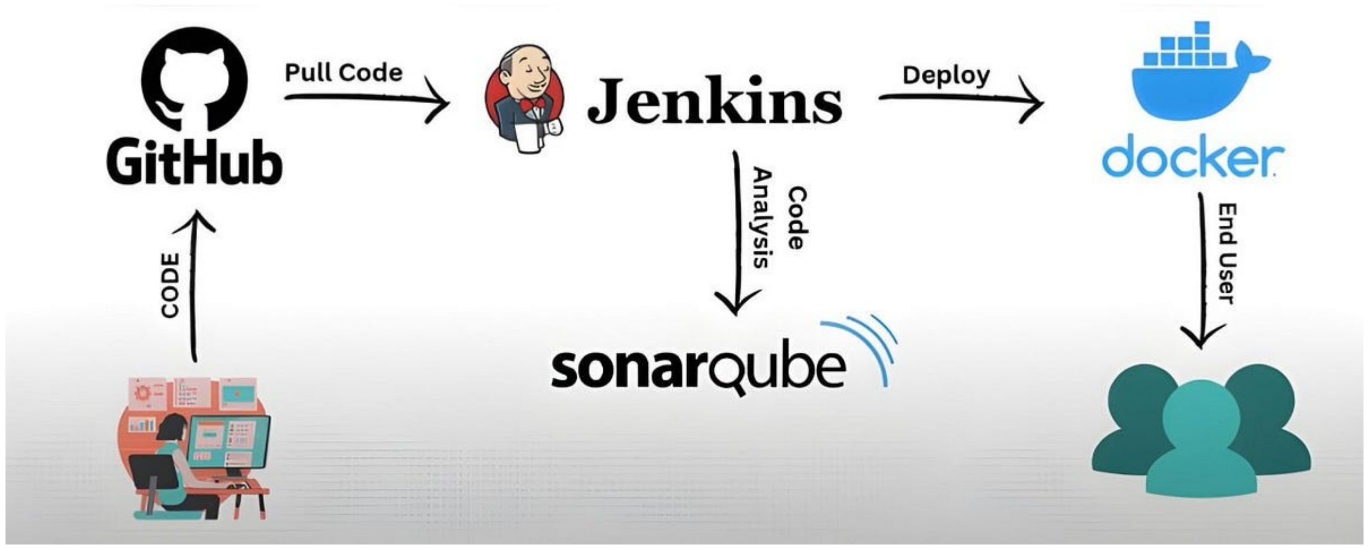
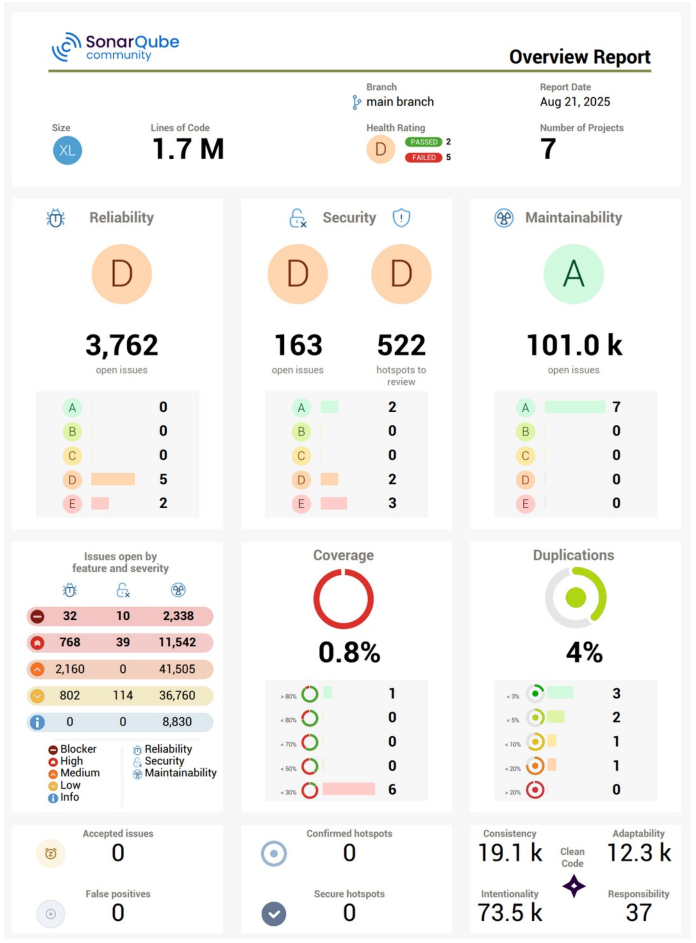


Image Credit: https://medium.com/@abubakr.sadiq/integrating-sonarqube-with-jenkins-pipeline-5bbfe3b46655

Code Quality Plugins (SonarQube)

- Why SonarQube is popular
  - Language-agnostic
  - Enterprise-friendly
  - Strong Jenkins integration
  - Supports visual dashboards
- Regular use of code scans support
  - Maintaining code quality
  - Reducing technical debt incurred by poorly structured code
  - Enforcing standards and best practices



SonarQube community

Overview Report

Size	Lines of Code	Branch / Health Rating	Report Date / Number of Projects
XL	1.7 M	main branch / D (PASSED 2, FAILED 5)	Aug 21, 2025 / 7

	Reliability	Security	Security (hotspots)	Maintainability
Rating	D	D	D	A
	3,762 open issues	163 open issues	522 hotspots to review	101.0 k open issues

	Reliability	Security	Maintainability
A	0	2	7
B	0	0	0
C	0	0	0
D	5	2	0
E	2	3	0

Issues open by feature and severity

	Reliability	Security	Maintainability
Blocker	32	10	2,338
High	768	39	11,542
Medium	2,160	0	41,505
Low	802	114	36,760
Info	0	0	8,830

Coverage: 0.8%

> 80%	1
< 80%	0
< 70%	0
< 50%	0
< 30%	6

Duplications: 4%

< 3%	3
< 5%	2
< 10%	1
< 20%	1
> 20%	0

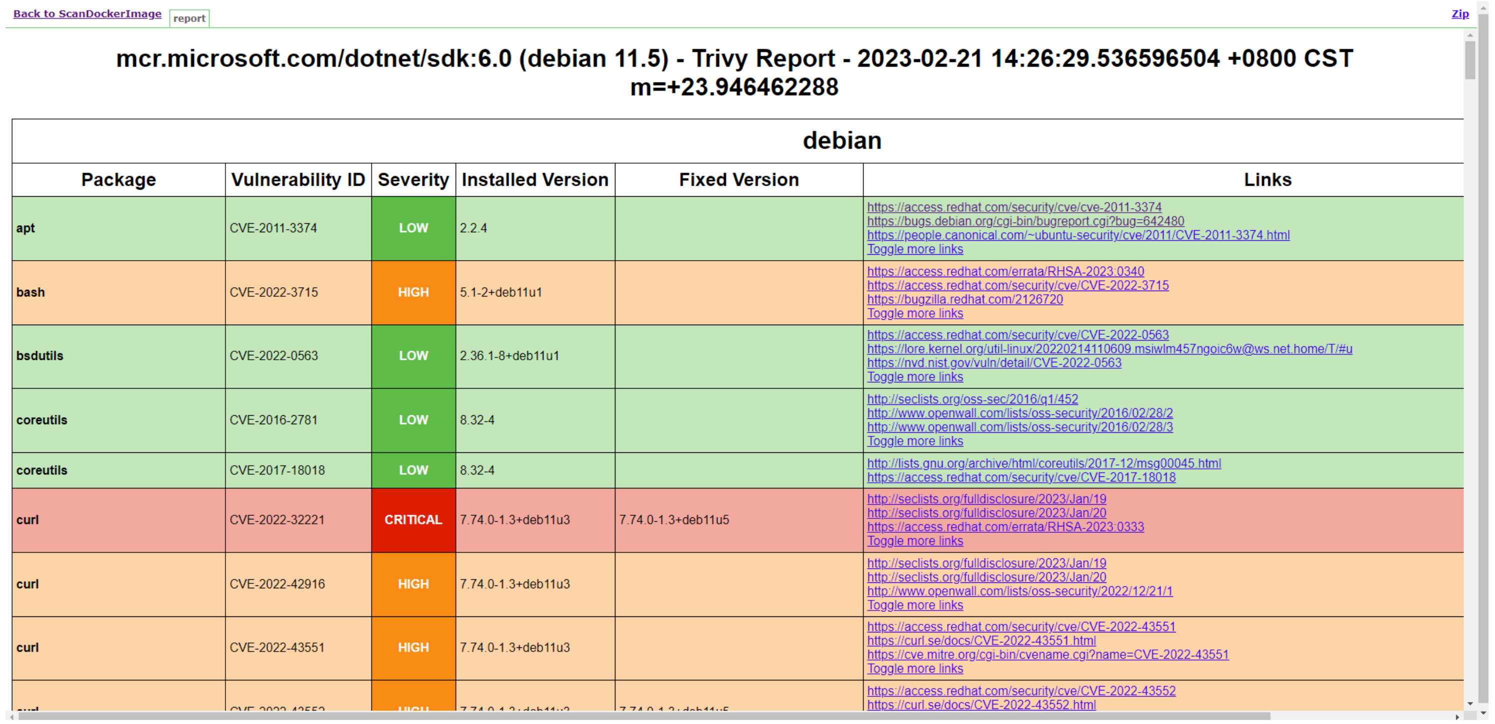
Accepted issues: 0
False positives: 0
Confirmed hotspots: 0
Secure hotspots: 0

Clean Code
Consistency: 19.1 k
Adaptability: 12.3 k
Intentionality: 73.5 k
Responsibility: 37

Image Credit: https://www.bitegarden.com/how-to-create-sonarqube-overview-mqr

Security and Dependency Scanning

- Modern CI pipelines:
  - Shift security left
  - Catch vulnerabilities early
  - Automate compliance checks
- Common security plugin types
  - Dependency vulnerability scanning
  - Container image scanning
  - Secret detection
  - License compliance checks
- Examples:
  - OWASP Dependency-Check
  - Trivy
  - Snyk
  - Anchore



Back to ScanDockerImage report Zip

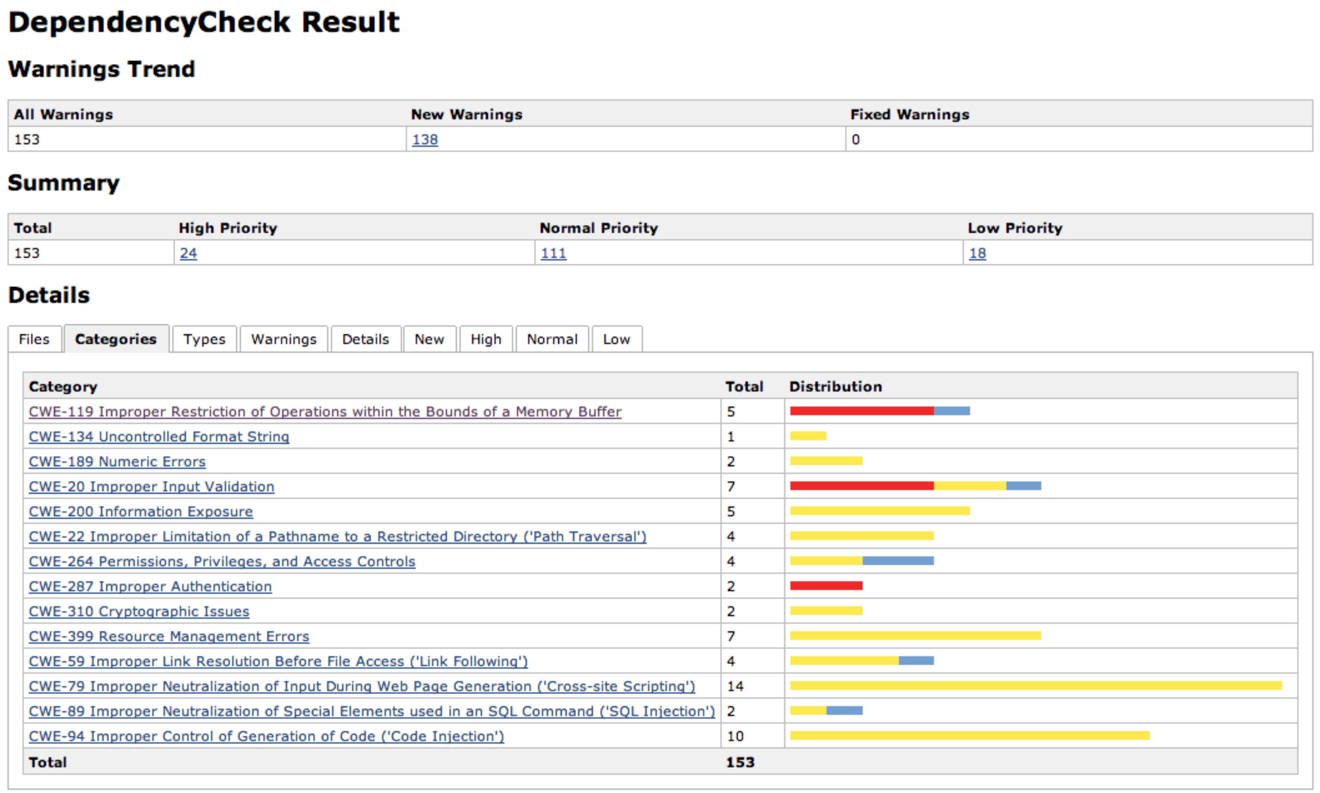
mcr.microsoft.com/dotnet/sdk:6.0 (debian 11.5) - Trivy Report - 2023-02-21 14:26:29.536596504 +0800 CST m=+23.946462288

debian					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
apt	CVE-2011-3374	LOW	2.2.4		https://access.redhat.com/security/cve/cve-2011-3374 https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=642480 https://people.canonical.com/~ubuntu-security/cve/2011/CVE-2011-3374.html Toggle more links
bash	CVE-2022-3715	HIGH	5.1-2+deb11u1		https://access.redhat.com/errata/RHSA-2023:0340 https://access.redhat.com/security/cve/CVE-2022-3715 https://bugzilla.redhat.com/2126720 Toggle more links
bsdutils	CVE-2022-0563	LOW	2.36.1-8+deb11u1		https://access.redhat.com/security/cve/CVE-2022-0563 https://lore.kernel.org/util-linux/20220214110609.msiwlm457ngoic6w@ws.net.home/T/#u https://nvd.nist.gov/vuln/detail/CVE-2022-0563 Toggle more links
coreutils	CVE-2016-2781	LOW	8.32-4		http://seclists.org/oss-sec/2016/q1/452 http://www.openwall.com/lists/oss-security/2016/02/28/2 http://www.openwall.com/lists/oss-security/2016/02/28/3 Toggle more links
coreutils	CVE-2017-18018	LOW	8.32-4		http://lists.gnu.org/archive/html/coreutils/2017-12/msg00045.html https://access.redhat.com/security/cve/CVE-2017-18018
curl	CVE-2022-32221	CRITICAL	7.74.0-1.3+deb11u3	7.74.0-1.3+deb11u5	http://seclists.org/fulldisclosure/2023/Jan/19 http://seclists.org/fulldisclosure/2023/Jan/20 https://access.redhat.com/errata/RHSA-2023:0333 Toggle more links
curl	CVE-2022-42916	HIGH	7.74.0-1.3+deb11u3		http://seclists.org/fulldisclosure/2023/Jan/19 http://seclists.org/fulldisclosure/2023/Jan/20 http://www.openwall.com/lists/oss-security/2022/12/21/1 Toggle more links
curl	CVE-2022-43551	HIGH	7.74.0-1.3+deb11u3		https://access.redhat.com/security/cve/CVE-2022-43551 https://curl.se/docs/CVE-2022-43551.html https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-43551 Toggle more links
curl	CVE-2022-43552	HIGH	7.74.0-1.3+deb11u3	7.74.0-1.3+deb11u5	https://access.redhat.com/security/cve/CVE-2022-43552 https://curl.se/docs/CVE-2022-43552.html

Image Credit: https://github.com/aquasecurity/trivy/issues/3660

Security and Dependency Scanning

- Security plugins
  - Run as pipeline steps
  - Produce reports
  - Can fail builds automatically



DependencyCheck Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
153	138	0

Summary

Total	High Priority	Normal Priority	Low Priority
153	24	111	18

Details

Files | **Categories** | Types | Warnings | Details | New | High | Normal | Low

Category	Total	Distribution
CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer	5	🟥🟥🟥🟥🟦
CWE-134 Uncontrolled Format String	1	🟨
CWE-189 Numeric Errors	2	🟨🟨
CWE-20 Improper Input Validation	7	🟥🟥🟥🟥🟨🟨🟦
CWE-200 Information Exposure	5	🟨🟨🟨🟨🟨
CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	4	🟨🟨🟨🟨
CWE-264 Permissions, Privileges, and Access Controls	4	🟨🟨🟦🟦
CWE-287 Improper Authentication	2	🟥🟥
CWE-310 Cryptographic Issues	2	🟨🟨
CWE-399 Resource Management Errors	7	🟨🟨🟨🟨🟨🟨🟨
CWE-59 Improper Link Resolution Before File Access ('Link Following')	4	🟨🟨🟨🟦
CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	14	🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨
CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	2	🟨🟦
CWE-94 Improper Control of Generation of Code ('Code Injection')	10	🟨🟨🟨🟨🟨🟨🟨🟨🟨🟨
Total	**153**	

Image Credit: https://wiki.jenkins.io/JENKINS/OWASP-Dependency-Check-Plugin.html

Cloud, Docker, and Kubernetes Integration

- Early Jenkins setups
  - Ran builds on the controller
  - Used long-lived static agents
  - Were hard to scale
  - Had inconsistent environments
- Modern Jenkins needs
  - Isolation between builds
  - Elastic scaling
  - Clean environments per job
  - Cost-efficient resource usage
    - *Jenkins controls builds, but agents execute them.*
    - *Cloud and container integrations are ways to create agents dynamically.*

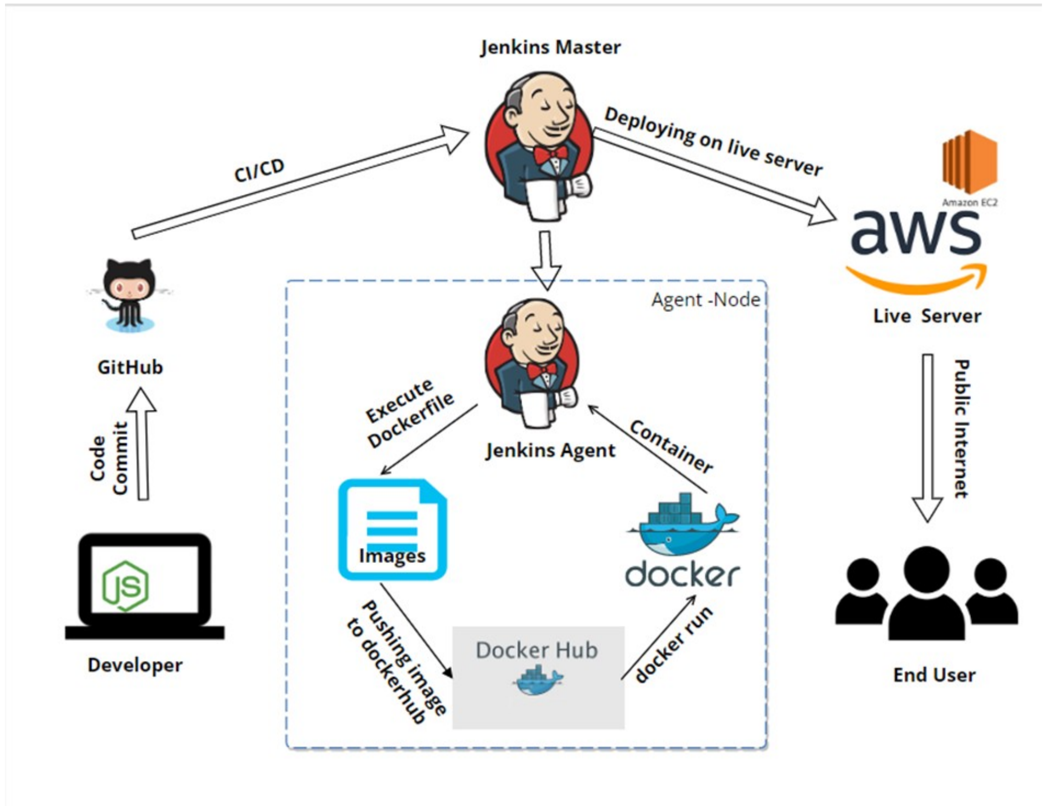


Image Credit: https://asktheman.xyz/

Docker Integration

- Docker integration means two things
  - Docker as a build environment
    - *Jenkins agent runs inside a Docker container*
    - *Each build gets a clean container*
    - *Tools are preinstalled or injected*
  - Docker as a build target
    - *Pipeline builds Docker images*
    - *Pushes them to registries*
    - *Deploys containers*

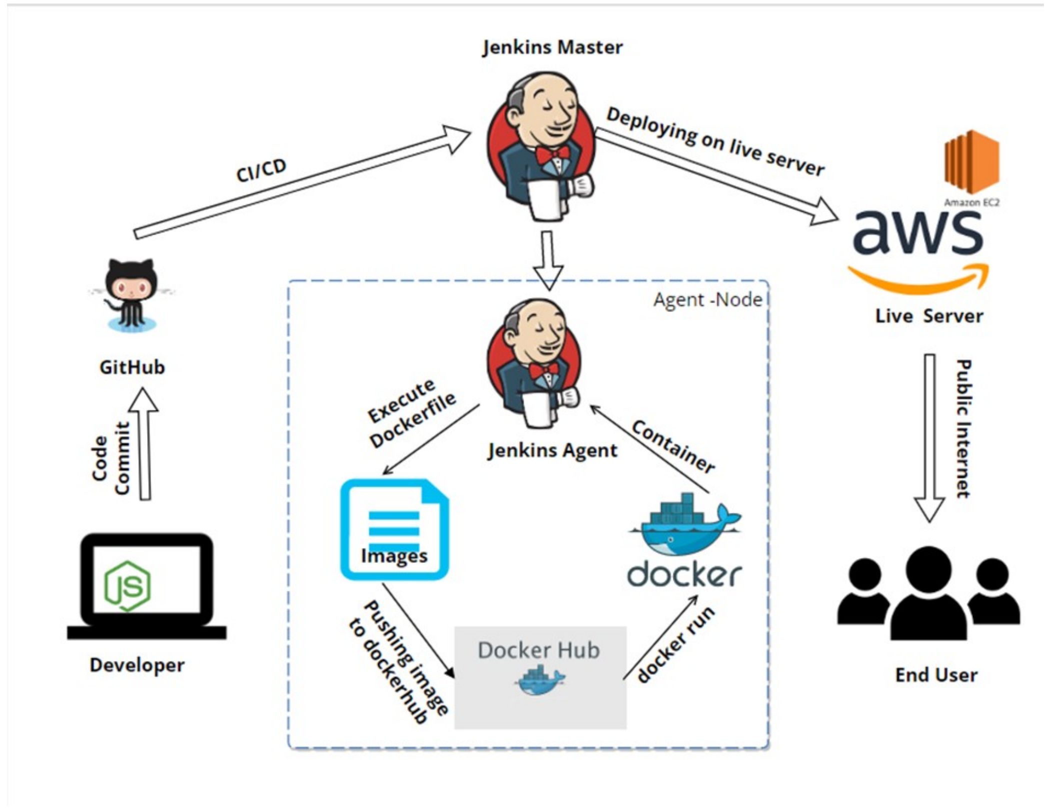


Image Credit: https://asktheman.xyz/

Common Docker Plugins and Capabilities

- Docker-related plugins enable
  - Docker-based agents
  - Running steps inside containers
  - Managing Docker credentials
  - Interacting with Docker registries
- Use cases
  - Language-specific builds
  - Legacy tool isolation
  - Reproducible CI environments

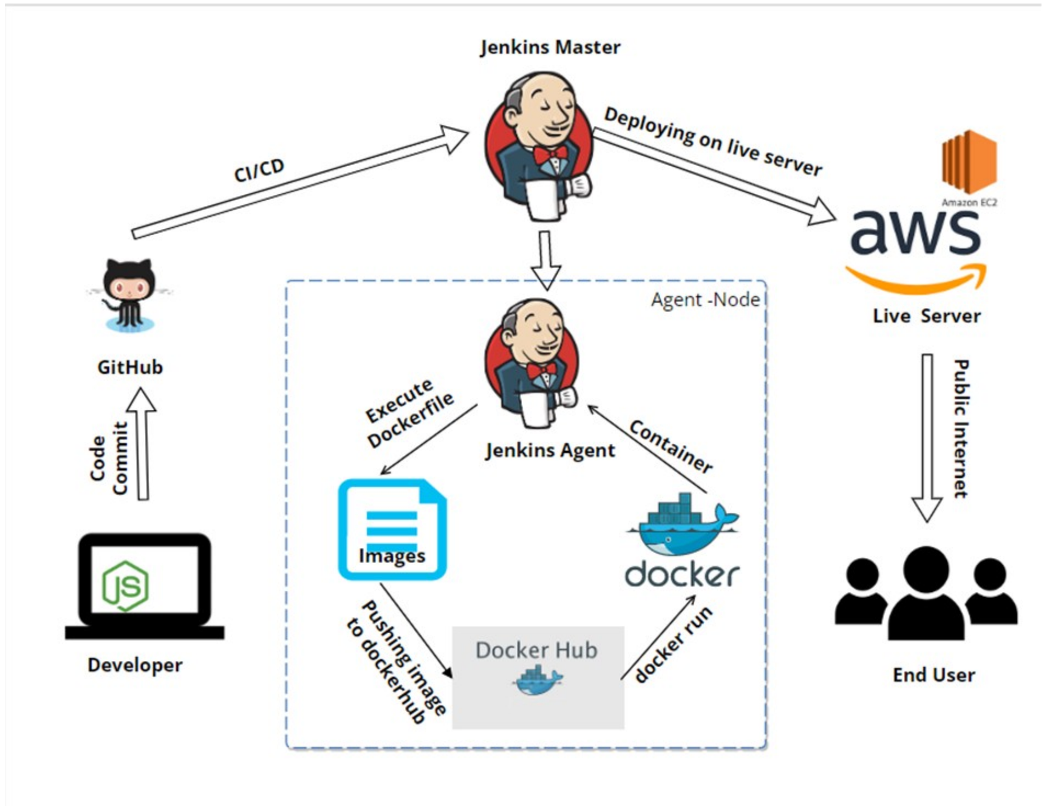


Image Credit: https://asktheman.xyz/

Kubernetes Integration in Jenkins

- Kubernetes solves problems Docker alone can't
  - Multi-node scheduling
  - Auto-scaling
  - Resource management
  - High availability
- Kubernetes can act as a dynamic agent factory
  - Can scale up the number of Docker containers required in a pipeline
  - Allows better load balancing across agents
  - Allows agents to scale as needed

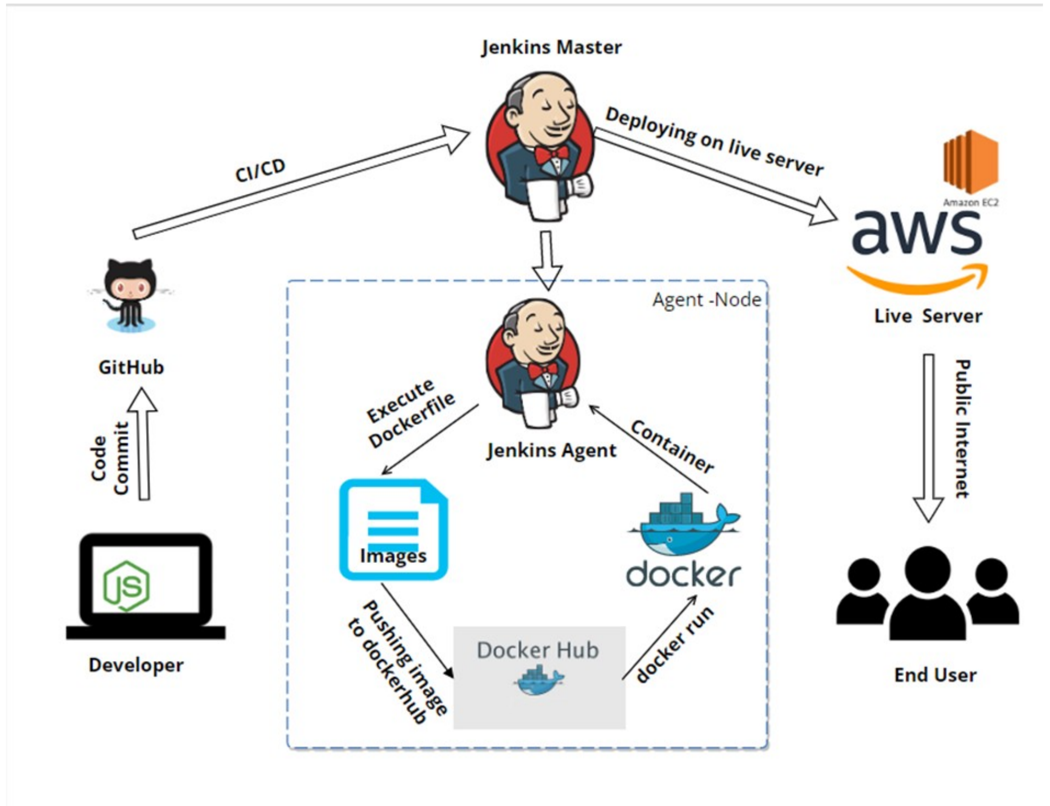


Image Credit: https://asktheman.xyz/

Kubernetes Integration in Jenkins

- The Kubernetes plugin allows Jenkins to
  - Define agent templates as pods
  - Request pods on demand
  - Run pipelines inside pods
  - Destroy pods after completion
- Pipeline flow
  - Jenkins needs an agent
  - Kubernetes creates a deployment to create pods that contain agents
  - Pipeline runs
  - Pod is deleted

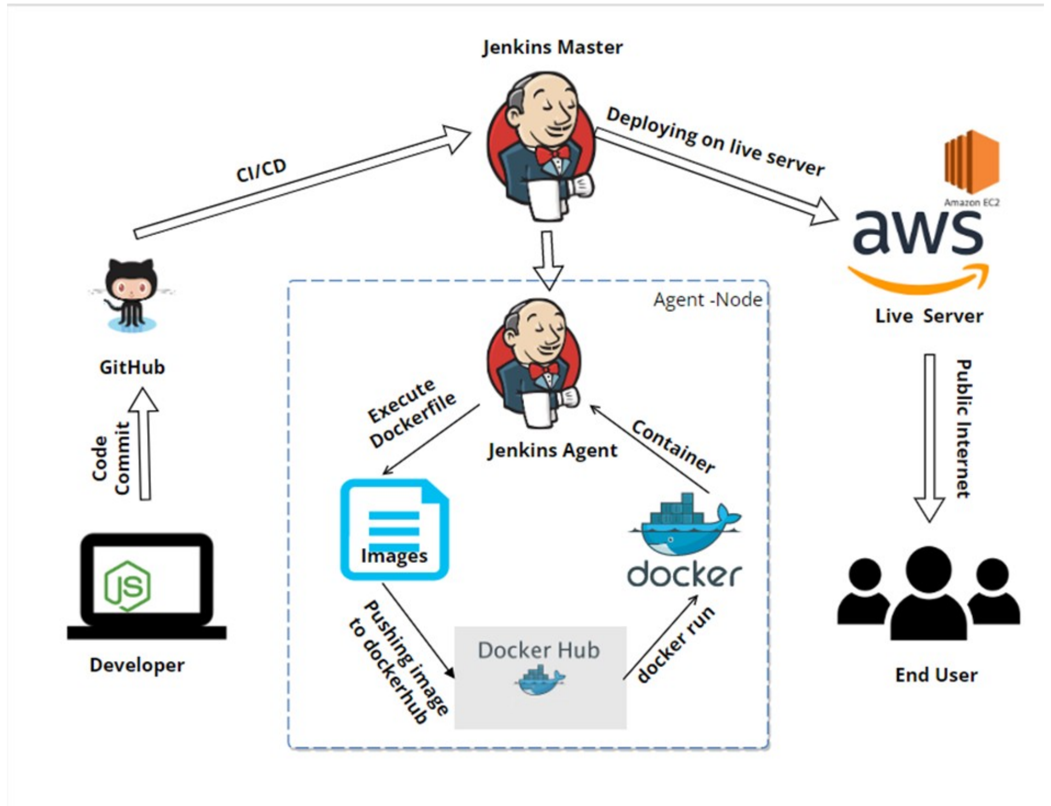


Image Credit: https://asktheman.xyz/

Kubernetes Integration in Jenkins

- Ephemeral pods
  - Reduce attack surface
  - Prevent state leakage
  - Ensure clean environments
  - Simplify scaling

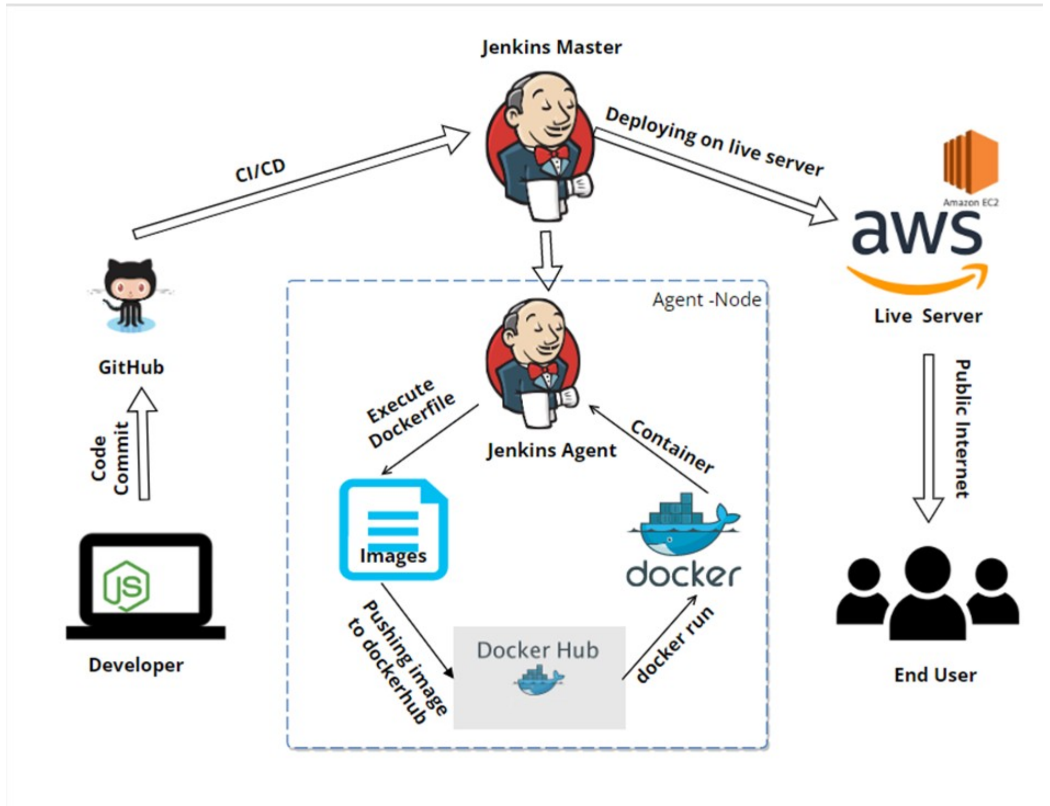


Image Credit: https://asktheman.xyz/

Cloud Provider Integration in Jenkins

- Cloud plugins allow Jenkins to
  - Provision agents as VMs
  - Integrate with cloud identity systems
  - Use cloud-native storage and networking
- Examples
  - EC2 agents
  - Managed Kubernetes clusters
  - Cloud credential injection

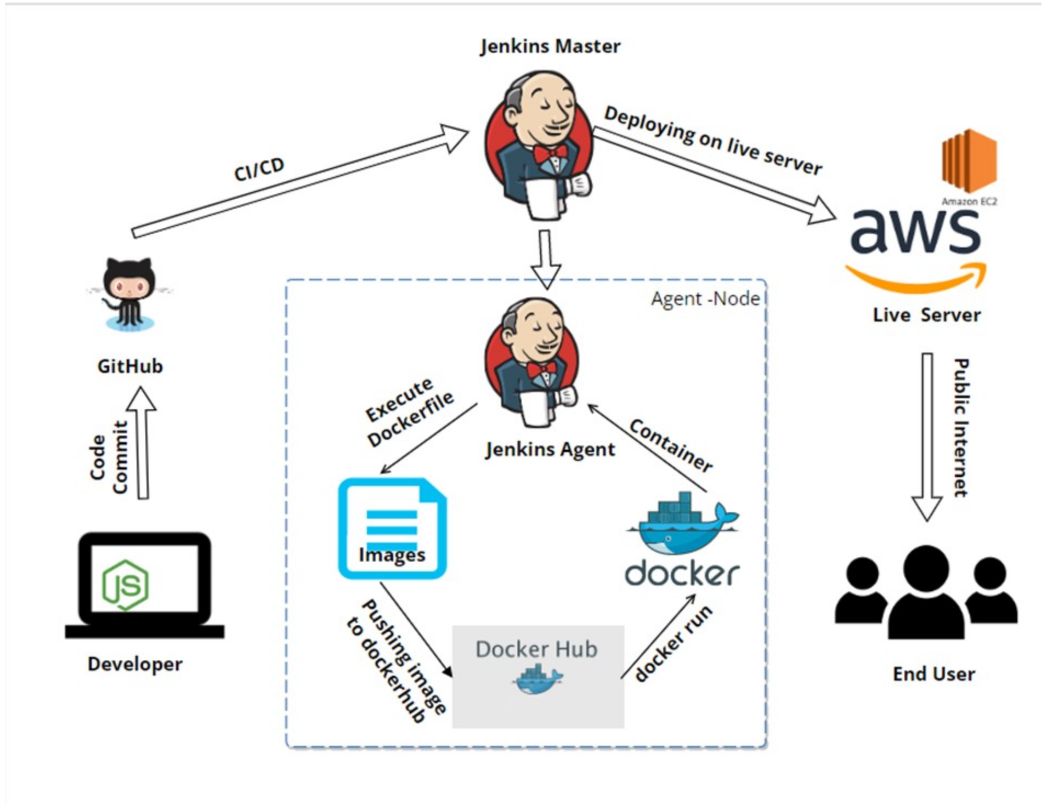


Image Credit: https://asktheman.xyz/

Cloud-Based Jenkins Pattern

- Jenkins controller runs
  - On-prem
  - In a VM
  - In Kubernetes
- Agents run
  - On cloud VMs
  - In Kubernetes pods
  - As containers
- This supports
  - Hybrid environments
  - Gradual cloud adoption

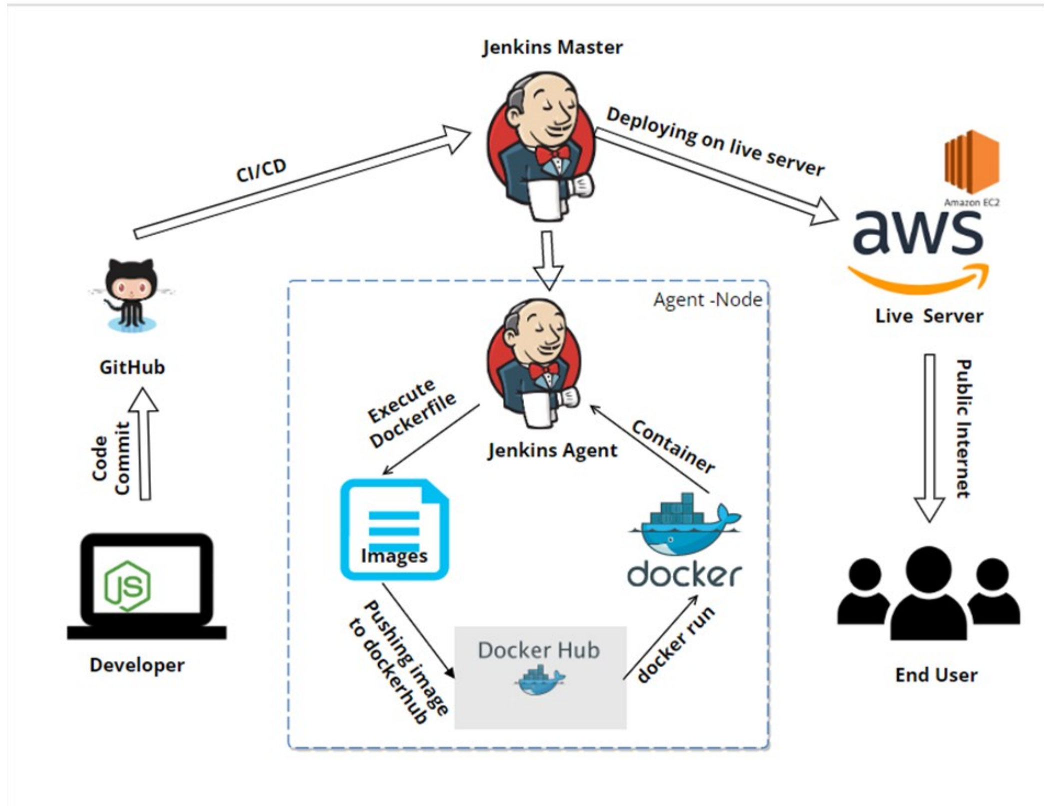


Image Credit: https://asktheman.xyz/

Comparison

Feature	Docker Agents	Kubernetes Agents	Cloud VM Agents
Setup Complexity	Low	Medium	Medium–High
Scalability	Limited	Excellent	Good
Isolation	Good	Excellent	Good
Cost Efficiency	Good	Excellent	Variable
Enterprise Adoption	High	Very High	High

Tool Config Revisited

- With ephemeral agents
  - Nothing is preinstalled
- Tools must be provided in one of the following ways
  - Auto-installation when the agent runs
  - Prebaked into Docker images
  - Provided via additional containers
- This is why
  - Global tool definitions matter
  - Container images are curated
  - Shared libraries become important

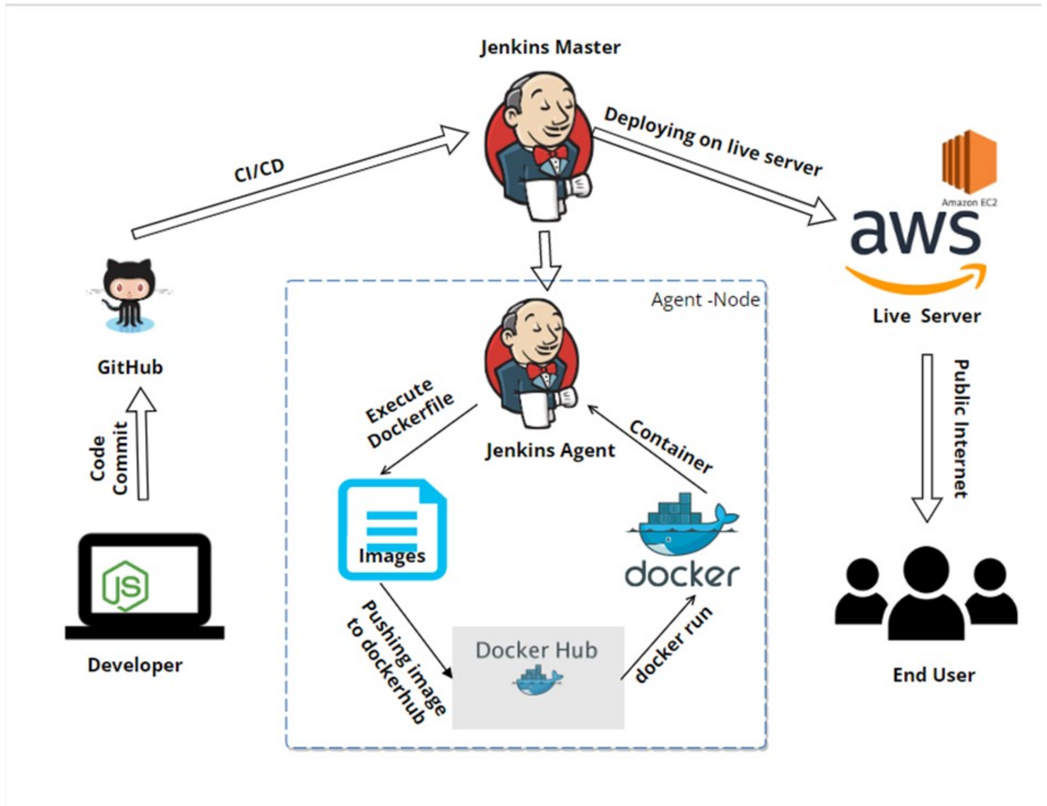


Image Credit: https://asktheman.xyz/

Global Tool Definitions

- When Jenkins uses
  - Docker agents
  - Kubernetes pod agents
  - Cloud-provisioned agents
- Those agents
  - Start empty
  - Are destroyed after the build
  - Do not remember previous builds
  - So Jenkins cannot assume tools already exist

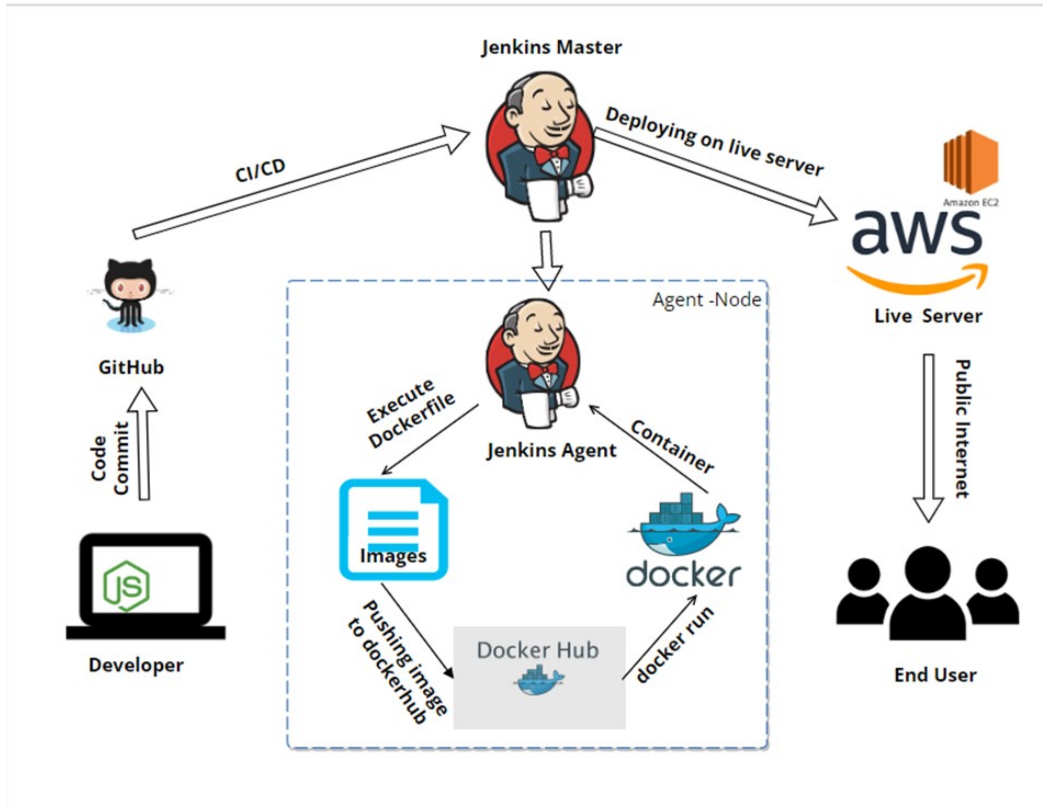


Image Credit: https://asktheman.xyz/

Global Tool Definitions

- Global tools
  - Define what tools a pipeline needs
  - Provide consistent versions
- Allow Jenkins agents to
  - Install tools automatically
  - Expose tools via PATH
  - Make pipelines portable

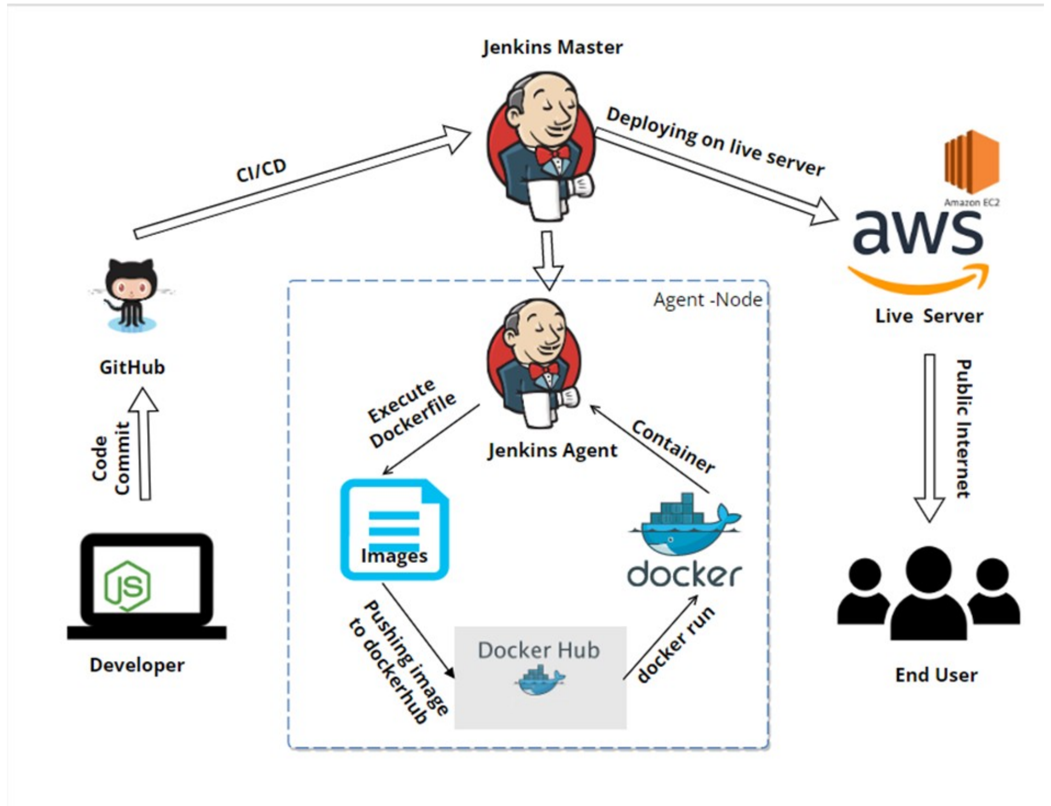


Image Credit: https://asktheman.xyz/

Container Images Are Curated

- Auto-installing tools on every build
  - Slows pipelines
  - Depends on external downloads
  - Introduces variability from repeated installations, version drift for example
  - Creates failure points
- Curated solution
  - Pre-build container images
  - With:
    - *JDK*
    - *Build tools*
    - *OS packages*
  - These images become standardized build environments

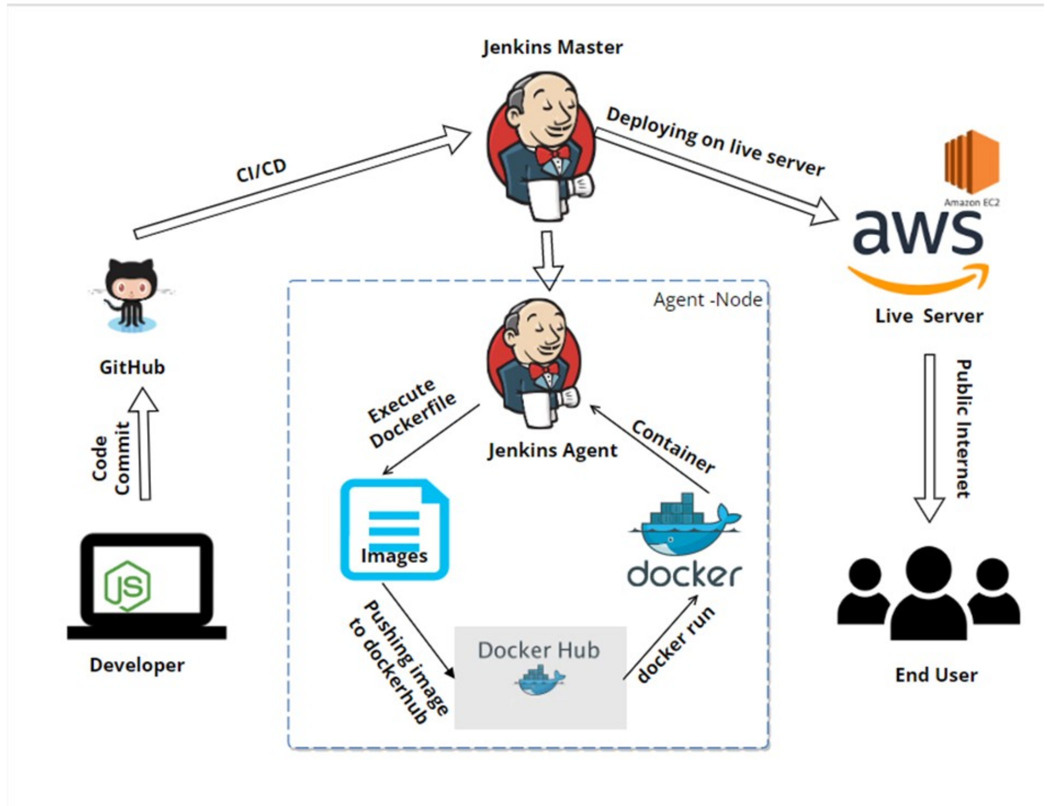


Image Credit: https://asktheman.xyz/

Container Images Are Curated

- A curated image is
  - Built intentionally
  - Versioned
  - Reviewed
  - Security-scanned
  - Used consistently
- Curated images are
  - Subject to governance rules
  - Part of the official Jenkins infrastructure in the organization

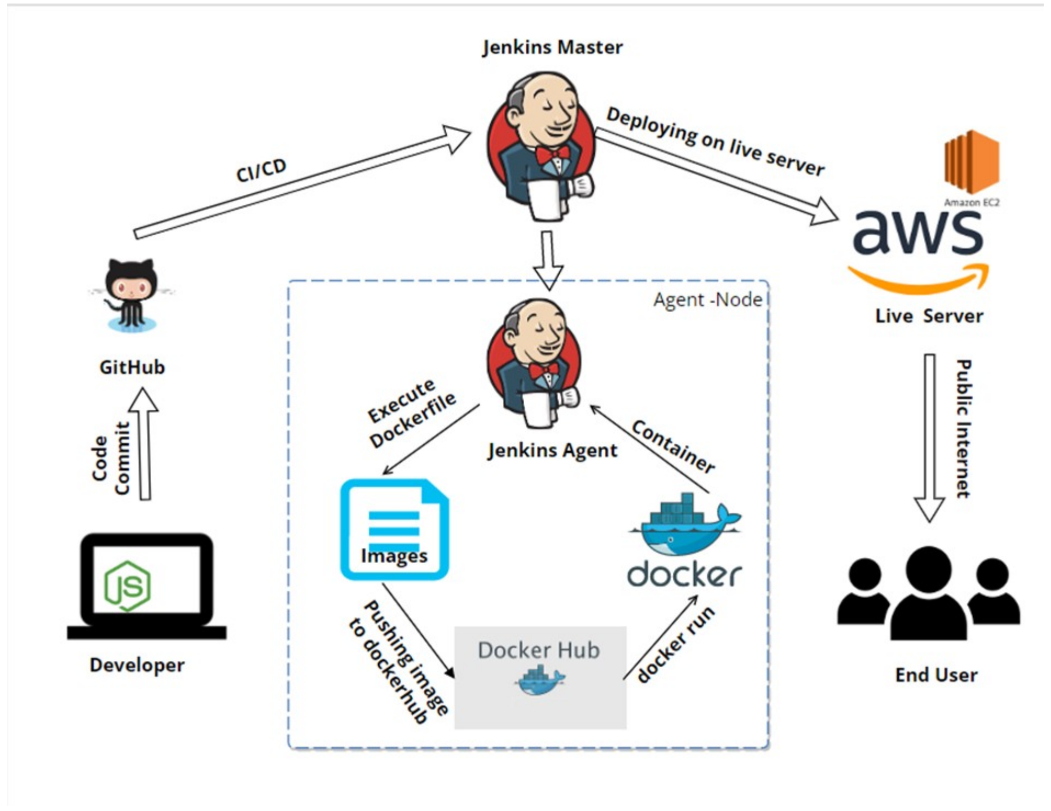


Image Credit: https://asktheman.xyz/

Shared Libraries

- When pipelines are used with
  - Many teams
  - Many repos
  - Many pipelines
  - Ephemeral agents
- Problems occur when
  - Every pipeline is installing tools
  - Every pipeline is reinventing logic
  - Every team is solving the same problems over and over

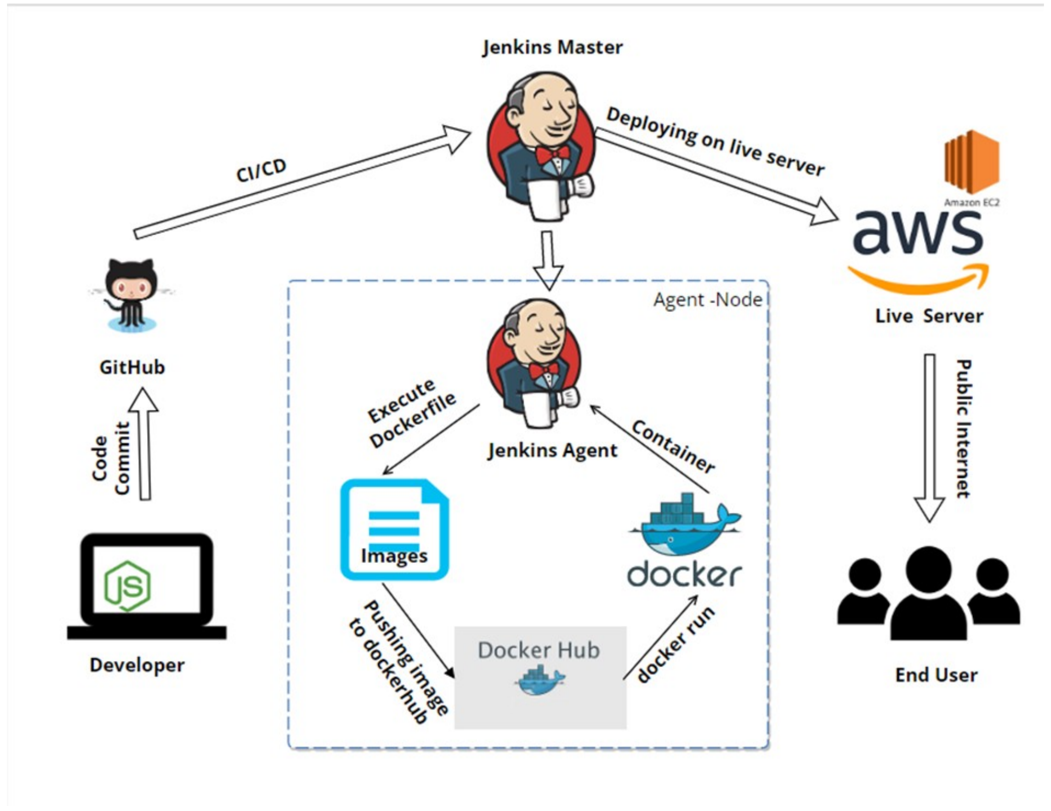


Image Credit: https://asktheman.xyz/

Shared Libraries

- Shared libraries
  - Centralize logic
  - Encapsulate complexity
  - Standardize how tools and images are used
  - Hide implementation details
- The library
  - Selects the right image
  - Ensures tools exist
  - Applies retries, logging, security

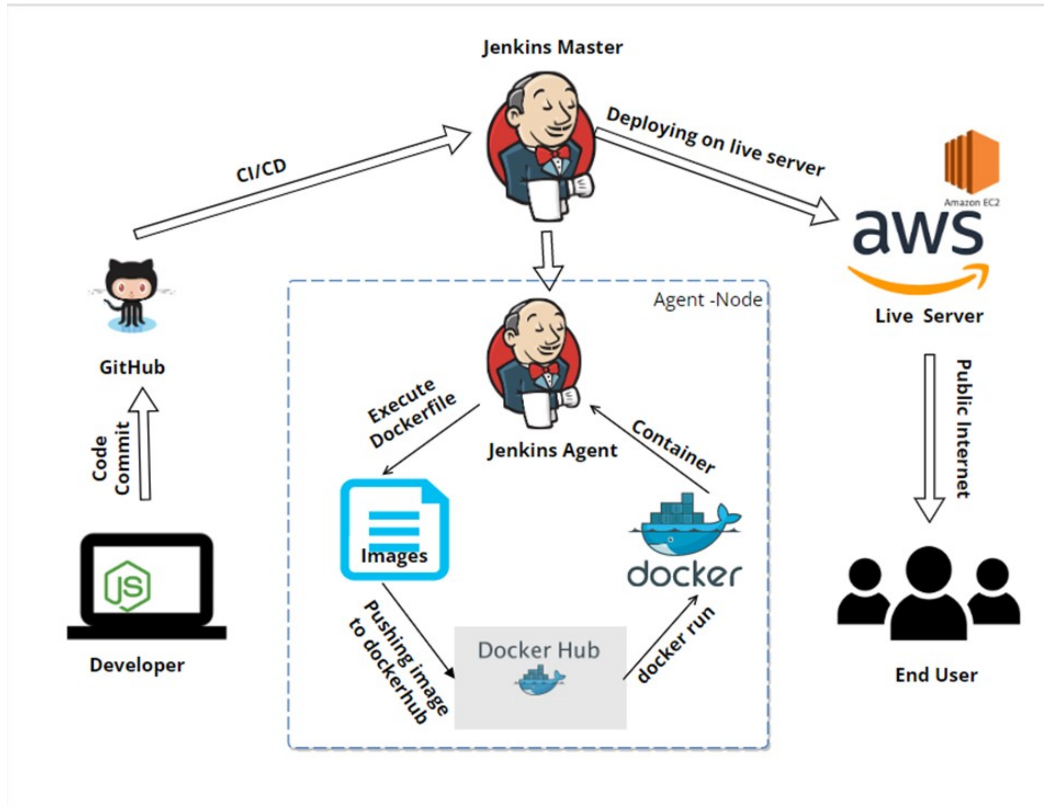


Image Credit: https://asktheman.xyz/

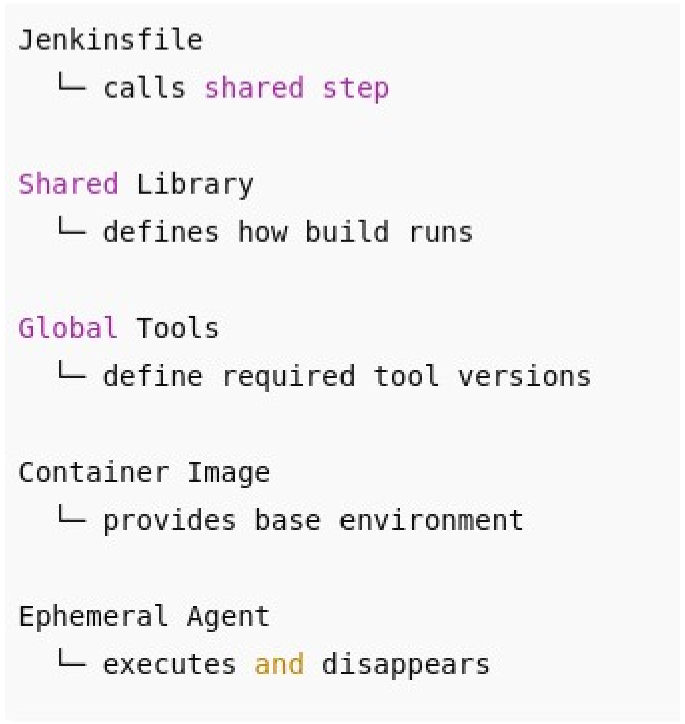
Modern Jenkins Flow

- Pipeline starts
- Jenkins provisions an ephemeral agent
- Agent uses a curated container image
- Global tool definitions standardize versions
- Shared library executes trusted logic
- Agent is destroyed

```
Jenkinsfile
  └─ calls shared step

Shared Library
  └─ defines how build runs

Global Tools
  └─ define required tool versions

Container Image
  └─ provides base environment

Ephemeral Agent
  └─ executes and disappears
```

Modern Jenkins Flow

- If these practices aren't used
  - Pipelines hardcode paths to tools
  - Agents' configurations drift over time
  - Builds behave inconsistently
  - Security issues multiply
  - Jenkins becomes a snowflake again



```
Jenkinsfile
  └─ calls shared step

Shared Library
  └─ defines how build runs

Global Tools
  └─ define required tool versions

Container Image
  └─ provides base environment

Ephemeral Agent
  └─ executes and disappears
```

Image Credit: https://asktheman.xyz/

Questions