



# Résumé QCM CI/CD

116 Questions avec Réponses et Pièges à Éviter | Révision Rapide

## Table des Matières

|                           |                            |                                  |
|---------------------------|----------------------------|----------------------------------|
| 1-5: Maven & .NET Basics  | 17-26: SonarCloud & SAST   | 61-80: Security & Best Practices |
| 6-10: GitLab CI/CD Basics | 27-32: Git & Ansible       | 81-100: Advanced Concepts        |
| 11-16: Docker & Tomcat    | 33-40: Jenkins & Pipelines | 101-116: Tools & Commands        |
|                           | 41-60: Commands Deep Dive  |                                  |

## Maven, .NET & Node.js (Q1-5)

1 Fichier définissant dépendances/plugins Maven ?  pom.xml (Project Object Model)

build.gradle (*Gradle*) package.json (*Node*) settings.xml (*config globale*)

2 Compiler Maven SANS tests ?  mvn clean package -DskipTests

mvn compile (*ne package pas*) --no-tests (*syntaxe invalide*)

3 Restaurer dépendances NuGet (.NET) ?  dotnet restore

dotnet install (*n'existe pas*) nuget restore (*pas CLI standard*)

4 Plugin Jenkins pour Maven ?  Maven Integration

Maven Builder Apache Maven Build Maven

5 Différence npm install vs npm ci ?  npm ci = package-lock.json strict (CI/CD déterministe)

ci = devDeps only (*faux, toutes deps*) install plus rapide (*inverse*)

## GitLab CI/CD (Q6-10)

6 Mot-clé pour étapes du pipeline ?  stages

phases (*Maven*) steps (*Jenkins/GitHub*) jobs (*tâches, pas étapes*)

7 Spécifier image Docker pour job ?  image:

container: docker: from: (*Dockerfile*)

8 Clé 'artifacts' permet de ?  Conserver/partager fichiers entre jobs/stages

télécharger deps    variables env    cache Docker

9 Image Docker pour Spring Boot ?  maven:3.9-eclipse-temurin-17

openjdk:17 (pas Maven)    springboot:latest (n'existe pas)

10 Clé 'needs' fait quoi ?  Exécuter job sans attendre fin du stage (DAG)

pré-requis système    tags runner (= tags:)    secrets

## Docker & Tomcat (Q11-16)

11 Vérifier Docker installé ?  docker run hello-world

12 Packaging pour Tomcat externe ?  <packaging>war</packaging>

jar (standalone)    ear (Java EE)

13 Pourquoi spring-boot-starter-tomcat scope 'provided' ?  Tomcat externe fournit déjà ces libs (évite conflits)

14 Classe pour init Spring Boot sur Servlet externe ?  ServletInitializer extends SpringBootServletInitializer

15 Dossier Tomcat pour .war ?  /var/lib/tomcat/webapps/

/deploy/    /applications/    /wars/

16 Récupérer artifacts d'un job build ?  needs: [{job: 'build', artifacts: true}]

## SonarCloud & SAST (Q17-26)

17 Variable pour token SonarCloud ?  SONAR\_TOKEN

18 Goal Maven pour SonarCloud ?  mvn sonar:sonar

sonar:analyze    sonar:scan

- 19 Propriété URL serveur Sonar ?  -Dsonar.host.url
- 20 Code Smell c'est quoi ?  Problème de maintenabilité (pas un bug)  
 faille sécurité (= Vulnerability)
- 21 Config SonarCloud dans Jenkins ?  Manage Jenkins → Configure System → SonarQube servers
- 22 SAST signifie ?  Static Application Security Testing
- 23 CWE-89 = ?  SQL Injection  
 XSS (= CWE-79)
- 24 Outil GitLab pour secrets hardcodés ?  Gitleaks  
 Semgrep (SAST) Trivy (conteneurs)
- 25 Bonne pratique correction vulnérabilité ?  1 vulnérabilité = 1 branche = 1 MR
- 26 Corriger secret dans .env versionné ?  Ajouter .env au .gitignore + variables CI/CD

## Git & Ansible (Q27-32)

- 27 Merge vs Rebase ?  Rebase = historique linéaire | Merge = garde historique fusion
- 28 Fichier Ansible pour hosts ?  inventory
- 29 Tester connectivité SSH Ansible ?  ansible all -m ping
- 30 Fichier YAML Ansible pour tâches ?  Playbook  
 Runbook Recipe (Chef)

31 **Module Ansible pour copier fichier ?**  copy

file (permissions) scp (cmd système)

32 **Exécuter playbook via SSH dans Jenkinsfile ?**  sh 'ssh user@host ansible-playbook playbook.yml'

## Jenkins & Pipelines (Q33-40)

33 **Jenkins vs GitLab CI/CD format ?**  Jenkins = Jenkinsfile (Groovy) | GitLab = .gitlab-ci.yml (YAML)

34 **Équivalent pom.xml pour Node.js ?**  package.json

35 **Équivalent 'mvn clean package' pour .NET ?**  dotnet restore && dotnet build --configuration Release

36 **Runner pour déployer sur Tomcat local ?**  Shell runner (accès direct services locaux)

Docker runner (isolé)

37 **Restreindre job à branche main ?**  only: [main]

branch: restrict: when:

38 **Artifact en CI/CD = ?**  Fichier généré par build (JAR/WAR) à conserver

39 **GIT\_DEPTH = ?**  Limiter commits clonés (shallow clone)

40 **Empêcher push directs sur branche ?**  Allowed to push: No one

## Commandes Essentielles (Q41-60)

41 **Version Maven ?**  mvn -v

42 **Créer projet console .NET ?**  dotnet new console -n MonProjet

- 43 Clé initiale Jenkins Linux ?  /var/lib/jenkins/secrets/initialAdminPassword
- 44 Où est pom.xml ?  À la racine du projet
- 45 Fichier .NET pour dépendances NuGet ?  \*.csproj  
 packages.config (*ancien format*)
- 46 Pourquoi stages séquentiels, jobs parallèles ?  Garantir dépendances entre étapes (test après build)
- 47 script vs before\_script ?  before\_script = avant chaque job | script = commandes principales
- 48 Job build-dotnet commande ?  dotnet restore && dotnet build --configuration Release
- 49 Syntaxe stage Jenkinsfile ?  stage('Build') { steps {} }
- 50 Script shell dans Jenkins ?  sh 'commande'
- 51 Changer répertoire Jenkins ?  dir('/chemin') {}
- 52 bat vs sh ?  sh = Unix/Linux | bat = Windows cmd
- 53 Vérifier Tomcat actif ?  systemctl status tomcat10
- 54 Pourquoi port 80 nécessite root ?  Ports < 1024 sont privilégiés
- 55 Dépendance scope 'provided' pour Tomcat ?  spring-boot-starter-tomcat
- 56 Copier WAR vers Tomcat ?  sudo cp springboot-app/target/\*.war /var/lib/tomcat10/webapps/
- 57 Redémarrer Tomcat ?  sudo systemctl restart tomcat10

- 58 Pourquoi Shell runner pour Tomcat local ?  Accès direct aux services locaux
- 59 Variables CI/CD pour SonarCloud ?  SONAR\_TOKEN, SONAR\_HOST\_URL, SONAR\_ORGANIZATION
- 60 Pourquoi GIT\_DEPTH: '0' pour Sonar ?  Cloner tout l'historique Git (analyses avancées)

## 🔒 Sécurité & Best Practices (Q61-80)

- 61 Bug vs Code Smell vs Vulnerability ?  Bug = erreur | Code Smell = maintenabilité | Vulnerability = faille sécurité
- 62 Token SonarCloud dans Jenkins ?  Manage Jenkins → Credentials → Add (Secret text)
- 63 Injecter variables Sonar dans pipeline ?  withSonarQubeEnv('SonarCloudServer')
- 64 Pourquoi Sonar APRÈS build ?  Besoin des .class pour certaines analyses
- 65 Pourquoi branche 'develop' ?  Intégrer/tester features avant production (main)
- 66 Comment SAST détecte SQL injection ?  Analyse statique + patterns dangereux (concaténation strings)
- 67 Pourquoi .env JAMAIS commité ?  Contient secrets (clés API, mots de passe)
- 68 Corriger SQL injection ?  Requêtes paramétrées (PreparedStatement)
- 69 Outil GitLab SAST ?  Semgrep
- 70 Corriger secret hardcodé ?  .gitignore + variable CI/CD masquée
- 71 Option 'Masked' pour variable GitLab ?  Project → Settings → CI/CD → Variables

72 Pourquoi 1 vuln = 1 branche = 1 MR ?  Revue ciblée, rollback facile, traçabilité

73 Pourquoi Ansible utilise SSH ?  Agentless (pas d'agent à installer/maintenir)

74 Clé publique vs privée SSH ?  Publique = partagée serveurs | Privée = secrète locale

75 Pourquoi ssh-copy-id ?  Ajoute clé dans authorized\_keys avec bonnes permissions

76 Groupes dans inventory Ansible ?  Cibler ensembles de serveurs avec playbooks spécifiques

77 Exécuter shell sur tous hosts ?  ansible all -m shell -a 'whoami && hostname'

78 Module command vs shell ?  shell = pipes, redirections | command = exécution directe

79 copy vs fetch Ansible ?  copy = envoie vers hosts | fetch = récupère depuis hosts

80 Module stat avant copy ?  Vérifier si fichier existe (idempotence)

## Concepts Avancés (Q81-100)

81 Idempotence Ansible ?  Même résultat peu importe combien de fois exécuté

82 Syntaxe artifact avec expiration ?  artifacts: paths: [...] expire\_in: 1 week

83 Agent Jenkinsfile déclaratif ?  agent any

runner (*GitLab*)

84 Variable chemin projet GitLab ?  \$CI\_PROJECT\_DIR

\$WORKSPACE (*Jenkins*)

85 **Commande complète SonarCloud Maven ?**

mvn clean verify sonar:sonar -Dsonar.projectKey=... -Dsonar.host.url=... -Dsonar.token=...

86 **mvn clean fait quoi ?**  Supprime target/ (fichiers générés)

87 **mvn compile vs mvn package ?**  compile = code source | package = compile + test + JAR/WAR

88 **-DskipTests ?**  Skip exécution tests (mais compile les tests)

89 **mvn test ?**  Compile code + compile tests + exécute tests

90 **Créer projet Maven interactif ?**  mvn archetype:generate

91 **mvn install vs mvn package ?**  install = package + copie dans ~/.m2/repository

92 **Créer branche ET se positionner ?**  git checkout -b nouvelle-branche

93 **git merge vs git rebase ?**  merge = commit fusion | rebase = historique linéaire

94 **git push origin develop ?**  Envoie branche develop locale vers remote origin

95 **git stash ?**  Sauvegarde temporaire sans commit

96 **Annuler modifs fichier non stagé ?**  git checkout -- fichier

97 **dotnet restore ?**  Télécharge packages NuGet

98 **dotnet build mode Release ?**  --configuration Release

99 **dotnet run ?**  Compile + exécute en une commande

100 **npm install vs npm ci ?** ✓ ci = package-lock.json strict | install = peut modifier

## 🛠️ Outils & Commandes (Q101-116)

101 **npm run build ?** ✓ Exécute script 'build' de package.json

102 **Installer comme devDependency ?** ✓ npm install package --save-dev

103 **ansible all -m ping ?** ✓ Teste SSH + disponibilité Python

✗ ICMP ping (*pas réseau*)

104 **Playbook avec inventory spécifique ?** ✓ ansible-playbook -i inventory.ini playbook.yml

105 **Dry-run Ansible ?** ✓ --check

106 **Shell sur groupe 'webservers' ?** ✓ ansible webservers -m shell -a 'df -h'

107 **ssh-keygen -t rsa -b 4096 ?** ✓ Paire clés RSA 4096 bits (publique + privée)

✗ clé privée (*JAMAIS!*)

109 **Redémarrer Tomcat ?** ✓ sudo systemctl restart tomcat10

110 **Logs temps réel service ?** ✓ journalctl -u tomcat10 -f

111 **docker run hello-world ?** ✓ Télécharge + exécute image test

112 **Tous conteneurs (+ arrêtés) ?** ✓ docker ps -a

113 sh vs bat Jenkinsfile ?  sh = Unix/Linux | bat = Windows

114 Changer répertoire Jenkinsfile ?  dir('/path') { ... }

115 Plusieurs commandes .gitlab-ci.yml ?  script: - cmd1 - cmd2 - cmd3

116 Accéder variable CI/CD ?  \$VARIABLE ou \${VARIABLE}

%VAR% (Windows cmd)

 **Astuce Révision:** Concentre-toi sur les pièges  - ce sont les erreurs les plus fréquentes à l'examen!

Généré le 4 Janvier 2026 | 116 Questions CI/CD