

9 BEST PRACTICES FÜR DIE VERWALTUNG VON ARTEFAKTEN

Artefakte sind wertvolle Bausteine und Produkte der Softwareentwicklung. In der heutigen schnelllebigen und sich rasant entwickelnden Softwareentwicklungslandschaft ist ein effektives Artefaktmanagement zu einem entscheidenden Faktor für den Geschäftserfolg von Unternehmen geworden. Effektive Artefaktmanagement-Praktiken zu verstehen und zu implementieren, kann die Produktivität erheblich steigern, Prozesse rationalisieren und die erfolgreiche Durchführung von Softwareprojekten sicherstellen.

Dieses E-Book soll Ihnen einen Überblick über neun Best Practices zur Verwaltung von Artefakten verschaffen. Wir gehen auf die einzelnen Best Practices ein und geben Ihnen praktische Tipps für die Umsetzung. Wenn Sie die hier beschriebenen Empfehlungen befolgen, verfügen Sie über das nötige Wissen und die Tools, um Artefakte während ihres gesamten Lebenszyklus effizient verwalten zu können.



9 BEST PRACTICES

- 1** Speichern von Paketen und Artefakten in einem Tool, das für diesen speziellen Dateityp konzipiert ist 03
- 2** Trennung von Artefakten und Abhängigkeiten mit einer lokalen, remote und virtuellen Repository-Struktur 04
- 3** Öffentliche Proxy-Registries für ein lokal zwischengespeichertes Set von Artefakten 05
- 4** Repositorys für jede Entwicklungsphase 06
- 5** Weisen Sie jedem Team ein eigenes Set von lokalen, remote und virtuellen Repositorys zu 07
- 6** Artefakte zwischen SDLC-Umgebungen übertragen – statt sie neu zu generieren 08
- 7** Speichern und Verwalten der Metadaten zu jedem Artefakt, gemeinsam mit diesem 09
- 8** Erstellung von Richtlinien, die die Zugriffsrechte auf Repositorys und das Ausführen von Aktionen regeln 10
- 9** Definition von Bereinigungsrichtlinien für Repository-Hygiene und - Performance 11

1. SPEICHERN VON PAKETEN UND ARTEFAKTEN IN EINEM TOOL, DAS FÜR DIESEN SPEZIELLEN DATEITYP KONZIPIERT IST

Keine zwei Pakettypen sind gleich – sie unterscheiden sich in Struktur, Inhalt und Konfiguration. Obwohl es möglich ist, mehrere Dateitypen in einem generischen Repository zu speichern, schränken Unternehmen damit die Automatisierung ihrer Software-Pipelines ein und reduzieren das Datenmaterial, das während des Entwicklungsprozesses erfasst wird. Aus diesem Grund ist die Speicherung von Paketen und Artefakten in Repositories, die für einen bestimmten Pakettyp entwickelt wurden, in der Regel die beste Lösung.

WARUM DAS WICHTIG IST

Jeder Paketmanager verfügt über ein eigenes Set von Befehlen, Spezifikationen und Datenausgaben. Um eine Automatisierung und nahtlose Integration in Ihre Software-Pipelines zu ermöglichen, müssen die Repositories, in denen Sie Ihre Artefakte speichern, in die verschiedenen Tools, die Sie in Ihren Build-Prozessen verwenden, integriert werden und mit ihnen kommunizieren können.

Das Speichern von Paketen und Artefakten in einem Tool, das speziell auf diesen Dateityp ausgelegt ist, erleichtert auch die Erfassung von Metadaten. Wenn Pakete und Artefakte in einem entsprechenden Tool gespeichert werden, kann dieses Informationen über die Dateien erfassen und speichern, z. B. den Dateityp, das Erstellungsdatum, wer die Datei erstellt hat und andere relevante Informationen.

Außerdem erleichtert es diese Best Practice, organisiert

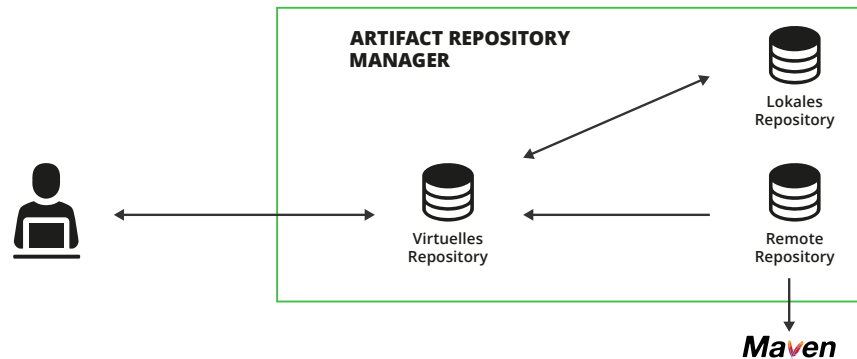
zu bleiben. Ein Tool, das für einen bestimmten Dateityp entwickelt wurde, kann verwendet werden, um Pakete und Artefakte in Ordnern, Tags und anderen Kategorien zu organisieren. Dadurch können User die benötigten Pakete und Artefakte leicht finden. Darüber hinaus ist es so einfacher, die verschiedenen Versionen von Paketen und Artefakten im Auge zu behalten.



MEHR ERFAHREN:

[Package Management Tool Consolidation](#)

2. TRENNUNG VON ARTEFAKTEN UND ABHÄNGIGKEITEN MIT EINER LOKALEN, REMOTE UND VIRTUELLEN REPOSITORY-STRUKTUR



WARUM DAS WICHTIG IST

Aus Sicherheits-, Struktur- und Compliance-Gründen benötigen Unternehmen eine einfache Möglichkeit, Artefakte zu speichern und zu organisieren, damit klar ist, welche Artefakte proprietär sind und welche aus öffentlichen Quellen stammen (z. B. Abhängigkeiten). Um intellektuelle Artefakte besser von nicht-intellektuellen Artefakten zu trennen, ist es am besten, getrennte Repositories für diese beiden unterschiedlichen Klassifizierungen von Artefakten zu verwenden.

Durch die Verwendung von lokalen und remote Repositories können Unternehmen ihre intellektuellen von nicht-intellektuellen Artefakten trennen. Der Einsatz virtueller Repositories vereinfacht den Administrations- und Governance-Aufwand bei Änderungen.

Moderne Softwareanwendungen bestehen aus einer Vielzahl von Paketen und Abhängigkeiten. Dazu gehören intern gefertigte Komponenten und Open-Source-Projekte, die Sie zur Beschleunigung Ihrer Releases verwenden. Als Teil eines Projekts benötigen Sie zum Beispiel Logging-Funktionen. Anstatt ein komplettes Logging-Framework von Grund auf neu zu entwickeln und zu testen, können Sie auf OSS-Logging-Frameworks wie log4j zurückgreifen, die bereits weiter entwickelt sind und sich bewährt haben.

Es empfiehlt sich, eine Repository-Struktur zu verwenden, die sowohl lokale als auch remote und virtuelle Repositories umfasst.

In **lokalen** Repositories können Sie alle Ihre proprietären Artefakte speichern, die Sie entsprechend Ihren Anforderungen entwickeln.

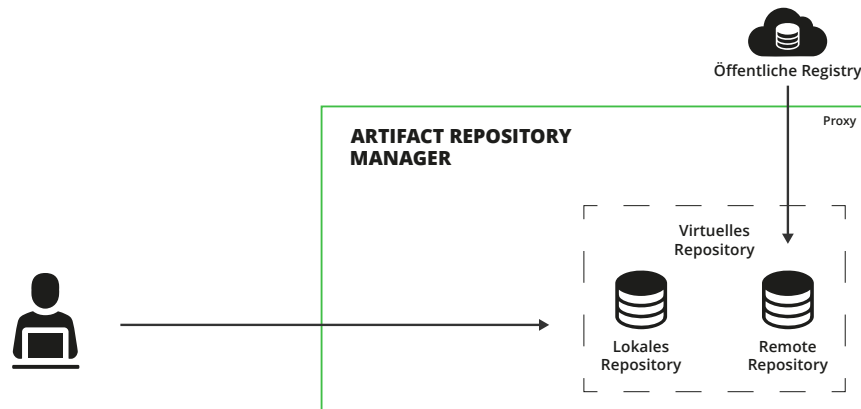
Verwenden Sie **remote** Repositories, um Pakete aus öffentlichen OSS-Repositories wie Maven Central, Docker Hub, npm usw. zwischenspeichern.

Virtuelle Repositories bieten eine einzigartige Möglichkeit, die Verwaltung von Repositories zu vereinfachen. Da es sich um eine Kombination von lokalen und remote Repositories handelt, können Sie ein virtuelles Repository als den einzigen Endpunkt für die Erstellung neuer proprietärer Artefakte verwenden, indem Sie OSS-Paketabhängigkeiten auflösen und intellektuelle Artefakte in Artifactory veröffentlichen.



[Best Practices for Structuring and Naming Artifactory Repositories](#)

3. ÖFFENTLICHE PROXY-REGISTRYS FÜR EIN LOKAL ZWISCHENGESPEICHERTES SET VON ARTEFAKTEN



Heutzutage bilden Open-Source-Pakete die Grundlage für jede im Einsatz befindliche Software. Developer und Build-Systeme verlassen sich häufig auf diese Open-Source- und Drittanbieter-Libraries, die in der Regel auf Remote-Systemen im Internet gehostet werden, um Softwareanwendungen zu erstellen. Obwohl diese Praxis weit verbreitet ist, stehen Unternehmen vor einigen Herausforderungen, wenn sie auf öffentlich gehostete Artefakte zurückgreifen.

WARUM DAS WICHTIG IST

Unternehmen müssen Zuverlässigkeit, Verfügbarkeit, Sicherheit und Nachvollziehbarkeit sicherstellen, wenn sie mit Artefakten aus öffentlichen Repositories arbeiten. Probleme können auftreten, wenn Artefakte/ Abhängigkeiten in öffentlichen Repositories nicht mehr verfügbar sind oder Netzwerkengpässe zu

Zuverlässigkeitsproblemen führen. Außerdem kann die Nachvollziehbarkeit schwierig sein, wenn Abhängigkeiten direkt aus dem Internet heruntergeladen werden. Die lokale Zwischenspeicherung von Artefakten in remote Repositories ermöglicht einen zuverlässigen und konsistenten Zugriff auf Pakete, ohne dass ständige Downloads erforderlich sind.

SECURITY BEST PRACTICE

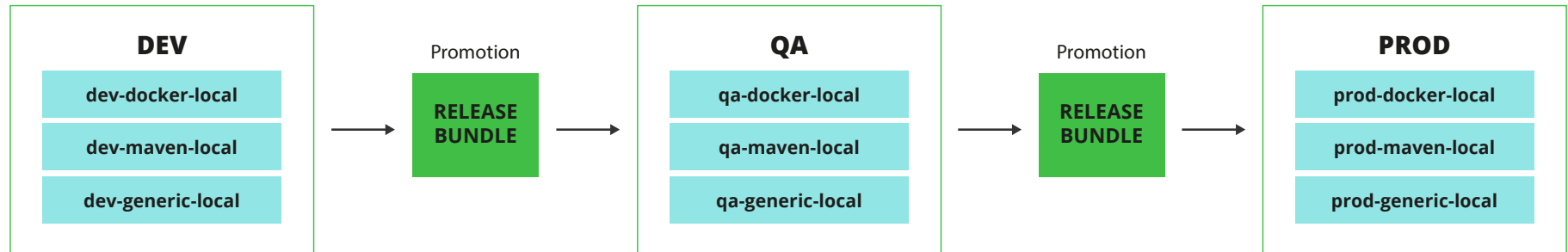
Erlauben Sie Entwicklern nicht, Pakete direkt aus dem Internet herunterzuladen.

Unternehmen, die bewährte Best Practices für Sicherheit einführen möchten, sollten ihren Artefakt-Repository-Manager als Vermittler zwischen Entwicklern und dem Internet einsetzen, indem sie öffentliche Registrys als Proxy nutzen. Selbst wenn ein böses Paket es nicht in den Build schafft, kann ein Developer, der Malware auf sein Gerät herunterlädt, das gesamte Netzwerk und System gefährden. Der Zugriff auf alle Pakete über ein Tool wie Artifactory bietet eine erste Abwehr- und Kontrollebene.



[Managing Open Source Security Risks and Vulnerabilities](#)

4. REPOSITORYS FÜR JEDE ENTWICKLUNGSPHASE

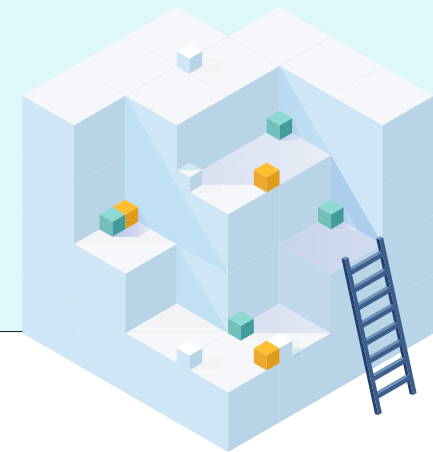


Unternehmen sollten Repositorys entsprechend aller Phasen ihres SDLC unterhalten. Software-Releases bestehen aus mehreren Builds/ Paketen/Artefakten, und Unternehmen bereiten Software für den Release vor, indem sie diese durch die Phasen des SDLC bewegen und in jeder Phase unterschiedliche Tests machen.

WARUM DAS WICHTIG IST

Durch das Einrichten von Repositorys für jede Phase des SDLC können Unternehmen alle Komponenten für einen potenziellen Release in Repositorys verschieben, die für die jeweilige Phase geeignet sind. Außerdem kann dadurch kontrolliert werden, wann die Software freigegeben wird (in der Regel, wenn sie bestimmte Prüfungen bestanden hat), wer auf die Komponenten zugreifen kann – abhängig von der Phase des SDLC – und dass zu keinem Zeitpunkt ein Build für einen Teil des Releases erneut ausgeführt wird. Dieser Ansatz hilft Unternehmen, die Integrität und Sicherheit zu verbessern und den Status von Software-Releases transparent nachzuverfolgen.

 **MEHR ERFAHREN**
Rethinking the SDLC



5. WEISEN SIE JEDEM TEAM EIN EIGENES SET VON LOKALEN, REMOTE UND VIRTUELLEN REPOSITORYS ZU

Jedem Team oder "Projekt" sollte ein eigenes Set von Repositorys für die Softwareentwicklung zugewiesen werden. Darüber hinaus sollte die verwendete Artefakt-Management-Lösung über eine Art Verwaltungsorgan verfügen, um Ressourcen wie Repositorys, Builds, Release-Bundles und Pipelines zu gruppieren. Auf diese Weise können Ressourcen leichter an ein bestimmtes Team delegiert werden.

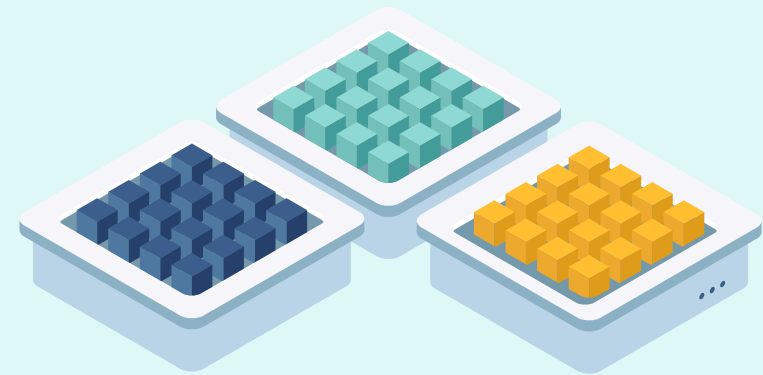
WARUM DAS WICHTIG IST

Das gilt aus mehreren Gründen als Best Practice. Zum einen werden Artefakte in Unternehmen meist instinktiv in einem einzigen System gespeichert, obwohl es für bestimmte Teams aus Gründen der Einfachheit und Sicherheit sinnvoller sein kann, auf eine vordefinierte Untergruppe von Artefakten zuzugreifen.

Zum anderen erleichtert die Zuweisung eigener Repositorys für jedes Team das Auffinden und den Zugriff auf die Komponenten, an denen gearbeitet werden soll, die gemeinsame Nutzung der Komponenten von verschiedenen Teams und – falls erforderlich – die Replikation oder Zusammenführung bestimmter Repositorys.

Wenn Sie mit sehr großen Repositorys arbeiten, kann es kompliziert werden. Indem Sie den verschiedenen Teams ihr eigenes Set an Repositorys zuweisen, nehmen Sie dem zentralen Administrator einen Teil der Last und des Wartungsaufwands ab. So weiß ein zentraler Administrator möglicherweise nicht, was er entfernen oder updaten kann, ohne dass dies Auswirkungen

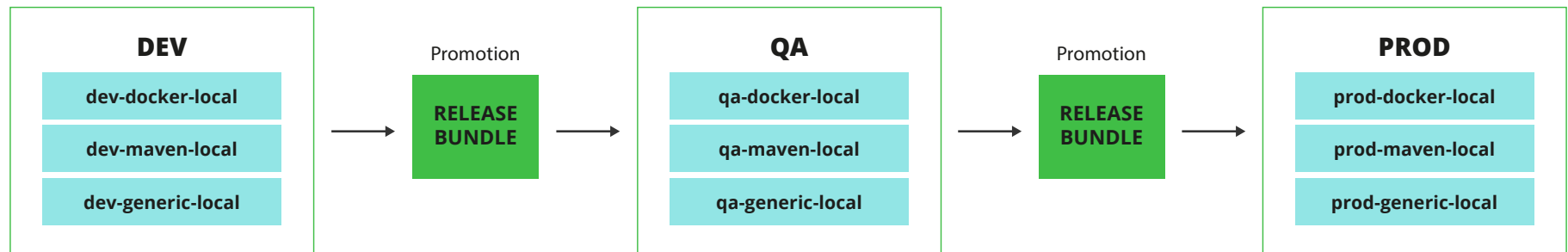
auf ein bestimmtes Team hat. Letztlich ist es auch finanziell ein kluger Schachzug, da eine effizientere und nachvollziehbare Zuordnung von Kosten zu bestimmten Teams ermöglicht wird.



MEHR ERFAHREN

[JFrog Federated Repositories](#)

6. ARTEFAKTE ZWISCHEN SDLC-UMGEBUNGEN WEITERGEBEN – NICHT NEU ERSTELLEN



Der Softwareentwicklungs-Lebenszyklus (Software Development Life Cycle = SDLC) umfasst verschiedene Phasen, die von der anfänglichen Planung und dem Entwurf über die Programmierung bis hin zu Tests, der Bereitstellung und der Wartung reichen. Ein wesentliches Element des Softwareentwicklungs-Lebenszyklus ist die Übertragung von Artefakten von einer Umgebung in eine andere. Um den Prozess so effizient und effektiv wie möglich zu gestalten, empfiehlt es sich, die Artefakte weiterzugeben, anstatt sie für jede Phase neu zu erstellen.

WARUM DAS WICHTIG IST

Die Weitergabe von Artefakten über den gesamten Lebenszyklus der Softwareentwicklung hinweg trägt dazu bei, Konsistenz, Zuverlässigkeit und Rückverfolgbarkeit zu gewährleisten und gleichzeitig Zeit und Ressourcen zu sparen. Durch die Weitergabe anstelle der Neugenerierung von Artefakten können Developer sicher sein, dass der Code, den sie testen, mit der Version übereinstimmt, die letztendlich bereitgestellt wird, und gleichzeitig den Reifeprozess der Software klar nachvollziehen. Die Durchführung von Security-Scans als Teil des Promotionprozesses kann dazu führen, dass die Scan-Ergebnisse als eine Art Ampel fungieren, um zu garantieren, dass die Builds sicher sind, wenn sie

in die nächste Phase des SDLC übergehen. Dieser Ansatz trägt dazu bei, Fehler zu reduzieren, die Quality Assurance zu verbessern, die Produktivität zu steigern und die Zeit bis zur Markteinführung zu verkürzen – das macht ihn zu einem wertvollen Ansatz für jedes Developer-Team.



MEHR ERFAHREN:

[How Does Build Promotion Work?](#)



7. SPEICHERN UND VERWALTEN DER METADATEN JEDES ARTEFAKTS

Zur Erinnerung: Metadaten sind zusätzliche Attribute, die mit einem Artefakt verknüpft sind. Diese Attribute bieten Administratoren die Möglichkeit, Artefakte effektiv zu organisieren. Sehen wir uns jetzt die verschiedenen Arten von Metadaten an.

1. Allgemeine Metadaten:

- Artefaktnamen: Der Name des Artefakts
- Artefakt-Version: Die Versionsnummer des Artefakts
- Artefakt-Typ: Der Typ oder das Format des Artefakts (z. B. JAR, WAR, Docker-Image)
- Artefakt-Größe: Die Größe der Artefaktdatei in Bytes
- Prüfsumme des Artefakts: Der Prüfsummenwert (z. B. MD5, SHA1, SHA256) der Artefaktdatei

2. Build-bezogene Metadaten:

- Build-Name: Der Name des mit dem Artefakt verknüpften Builds
- Build-Nummer: Die Nummer oder Kennzeichnung des Builds
- Zeitstempel des Builds: Der Zeitstempel, der angibt, wann der Build erstellt wurde

3. Metadaten zu den Abhängigkeiten:

- Abhängigkeiten: Informationen über die Abhängigkeiten des Artefakts, einschließlich ihrer Namen und Versionen

4. Benutzerdefinierte Metadaten:

- Benutzerdefinierte Metadaten: Zusätzliche projektspezifische Metadaten können definiert und mit Artefakten verknüpft werden

WARUM DAS WICHTIG IST

Metadaten spielen bei der Softwareentwicklung eine wichtige Rolle. Sie ermöglichen es Unternehmen, die Historie eines Artefakts oder Builds zu verstehen und nachzuvollziehen, was damit geschehen ist, z. B. Validierungen der Software auf dem Weg zum Release. Metadaten sind auch wichtig, um festzustellen, ob sich an einem Paket etwas geändert hat, was auf potenzielle Sicherheitsprobleme hinweisen könnte. Die Erfassung detaillierter Metadaten ist für die Rückverfolgbarkeit von Software unerlässlich und kann sogar zum Triggern von Workflows, Webhooks oder Plugins verwendet werden.



[Collect and Manage Your Binary Metadata Using Build-Info](#)

8. ERSTELLUNG VON RICHTLINIEN, DIE DIE ZUGRIFFSRECHTE AUF REPOSITORYS UND DAS AUSFÜHREN VON AKTIONEN REGELN

Unternehmen sollten über klare Regeln und Richtlinien verfügen, um steuern zu können, welche Personen und Systeme wann auf Software-Artefakte zugreifen dürfen. Es ist wichtig, dass Ihre Artefaktmanagement-Lösung in der Lage ist, diese Richtlinien via RBAC und/oder Integrationen mit anderen Identity-Management-Lösungen umzusetzen.



WARUM DAS WICHTIG IST

Software-Artefakte sind das Herzstück der Software-Entwicklung. Da die Pakete urheberrechtlich geschützten Code und Informationen enthalten, besteht ein großer Teil des Artefakt-Managements darin, sicherzustellen, dass keine Unbefugten auf Assets zugreifen können, die nicht für sie bestimmt sind.



MEHR ERFAHREN

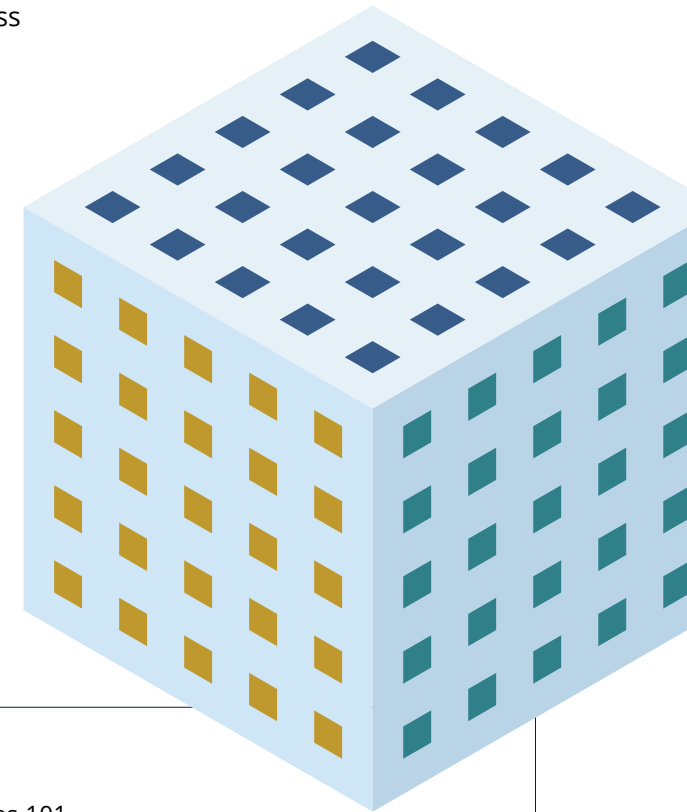
[Manage Project Roles and Members in JFrog](#)

9. DEFINITION VON BEREINIGUNGSRICHTLINIEN FÜR REPOSITORY-HYGIENE UND -PERFORMANCE

Die Anzahl der heute produzierten Binärdateien ist nahezu unüberschaubar. Mit der Zeit sammeln sich immer mehr Binärdateien an, die irgendwann eine große Menge an Speicherplatz benötigen. Das Ausmisten der Dateien kann mehrerlei bedeuten. Einerseits kann es sich um das tatsächliche Löschen der Binärdateien aus dem Repository-Manager handeln oder es bedeutet, wie in den meisten Fällen, dass die Binärdateien in einer kostengünstigeren Speicher-Lösung archiviert werden (dies ist besonders für stark regulierte Branchen relevant). In diesem Szenario werden die archivierten Binärdateien von dem Repository-Manager in die Archivierungslösung übertragen.

WARUM DAS WICHTIG IST

Genau wie in der physischen Welt wird unser "Zeug" zu einem ernsthaften Problem, wenn es sich immer weiter anhäuft und nicht richtig organisiert oder verwaltet wird. Cleanup-Richtlinien für Artefakt-Repositorys sind wichtig, um zu gewährleisten, dass die verwendeten Artefakte sicher und aktuell sind. Diese Richtlinien unterstützen die Integrität des Repositorys, indem sie verhindern, dass alte, anfällige oder schädliche Artefakte darin verbleiben. Durch das Entfernen von Artefakten, die nicht mehr verwendet werden, können Cleanup-Richtlinien auch die Gesamtgröße des Repositorys reduzieren. Dies trägt nicht nur dazu bei, dass das Repository übersichtlich und effizient bleibt, sondern kann auch die Wartungskosten senken.



MEHR ERFAHREN

[Custom Cleanup Strategies 101](#)

FAZIT

Wenn Sie diese Best Practices in Ihre Prozesse zur Verwaltung von Artefakten integrieren, sind Sie auf dem besten Weg zum Erfolg. Sie werden von optimierten Arbeitsabläufen, einer konstruktiveren Zusammenarbeit und besseren Projektergebnissen profitieren. Denken Sie daran, dass es bei der Verwaltung von Artefakten nicht nur darum geht, organisiert zu bleiben, sondern auch darum, Ihren gesamten Entwicklungszyklus zu optimieren und abzusichern.

Wir hoffen, Ihnen mit diesem E-Book das Wissen und die Mittel zur Verfügung zu stellen, die Sie benötigen, um Ihre Artefaktmanagement-Praktiken zu revolutionieren. Übernehmen Sie die Best Practices, passen Sie sie an Ihre spezifischen Bedürfnisse an und bemühen Sie sich um eine kontinuierliche Verbesserung. Mit einer soliden Grundlage im Artefaktmanagement sind Sie bestens gerüstet, um die Herausforderungen der modernen Unternehmenslandschaft zu meistern und hervorragende Ergebnisse zu erzielen.

Für einen tieferen Einblick in die Verwaltung von Artefakten, lesen Sie unser Whitepaper:

[Artifact Management: Best Practices for Enterprise Success](#)

ÜBER JFROG

JFrog Ltd. hat sich zum Ziel gesetzt, eine Welt zu schaffen, in der Software reibungslos vom Entwickler zum Endgerät geliefert wird. Angetrieben von der Vision einer "Liquid Software" ist die JFrog Software-Lieferkettenplattform ein einziges System, das Unternehmen ermöglicht, Software schnell und sicher zu erstellen, zu verwalten und zu verteilen und dabei zu gewährleisten, dass diese zugänglich, nachvollziehbar und manipulationssicher ist. Die integrierten Sicherheitsfunktionen helfen außerdem, Bedrohungen und Schwachstellen zu identifizieren, sich vor ihnen zu schützen und sie zu beheben. JFrogs hybride, universelle Multi-Cloud-Plattform ist sowohl als selbstgehosteter als auch als SaaS-Service bei den wichtigsten Cloud-Service-Anbietern verfügbar. Millionen von Nutzern und mehr als 7.000 Kunden weltweit, darunter ein Großteil der Fortune 100, verlassen sich auf JFrog-Lösungen, um die digitale Transformation sicher zu bewältigen.

