



DevOps mit SAP®

Methoden und Werkzeuge für effiziente Integration

- › DevOps in On-Premise-ABAP-Systemen mit abapGIT, CTS+ und mehr
- › SAP-BTP-Services für DevOps in Cloud-Landschaften
- › Management von DevOps in hybriden Szenarien

Raja Gupta
Sandip Jha
Merza Klaghstan



Rheinwerk
Publishing

Kapitel 2

DevOps-Tools

Um DevOps erfolgreich zu implementieren, ist es wichtig, die richtigen Tools und Technologien zur Verfügung zu haben. In diesem Kapitel stellen wir Ihnen einige der gängigen DevOps-Tools vor, die in Verbindung mit SAP eingesetzt werden, um die Softwarebereitstellung und -verteilung zu verbessern.

In Kapitel 1, »Einführung in DevOps«, haben Sie erfahren, was DevOps ist, wie es sich entwickelt hat und für moderne IT-Organisationen unverzichtbar geworden ist. Während die Idee von DevOps oft mit Denkweise, Kultur und Zusammenarbeit in Verbindung gebracht wird, benötigen wir auch eine Reihe von Tools und Frameworks, um DevOps-bezogene Aufgaben zu implementieren.

DevOps-Tools sind Softwaretools und Technologien, die zur Unterstützung der DevOps-Praktiken und -Prozesse eingesetzt werden. Diese Tools sollen Unternehmen dabei helfen, die Geschwindigkeit, Qualität und Zuverlässigkeit der Softwareentwicklung und -bereitstellung zu verbessern. Sie lassen sich in der Regel in die folgenden Kategorien einteilen:

- Tools zur Code- und Versionskontrolle, die wir in Abschnitt 2.1 besprechen.
- Build-Tools, über die Sie in Abschnitt 2.2 mehr erfahren.
- Tools zur Testautomatisierung, die Sie in Abschnitt 2.3 finden.
- Bereitstellungstools, die wir Ihnen in Abschnitt 2.4 näherbringen.
- Überwachungstools, die wir in Abschnitt 2.5 thematisieren.

Zum Abschluss des Kapitels werfen wir in Abschnitt 2.6 einen Blick auf das DevOps-Portfolio von SAP.

2.1 Tools zur Code- und Versionskontrolle

DevOps-Tools werden zur Verwaltung und Verfolgung von Änderungen am Quellcode von Anwendungen eingesetzt. Diese Tools ermöglichen es den Entwicklungsteams, effektiv an einer Codebasis zusammenzuarbeiten, auf frühere Versionen des Codes zurückzugreifen und eine historische Aufzeichnung der Änderungen zu führen. Für alle Entwicklungsteams wird eine einzige Quelle der Wahrheit verwaltet.

Mit diesen Tools können Entwickler*innen jede Änderung am Code verfolgen, zu früheren Versionen des Codes zurückkehren und mit anderen an der gleichen Codebasis zusammenarbeiten. Sie ermöglichen es auch, verschiedene Versionen des Codes zu erstellen, zu testen und in verschiedenen Umgebungen bereitzustellen. Darüber hinaus erleichtern sie die Integration verschiedener Entwicklungszweige, was das Zusammenführen und Lösen von Konflikten erleichtert.

Eines der beliebtesten Tools zur Code- und Versionskontrolle ist Git. In den folgenden Abschnitten werden wir sowohl Git selbst als auch GitHub besprechen, eine webbasierte Plattform für Versionskontrolle und Zusammenarbeit, die Git verwendet.

2.1.1 Git

Git ist ein verteiltes Code- und Versionskontrollsystem, das in der Softwareentwicklung weit verbreitet ist. Es ermöglicht den Entwicklungsteams, Änderungen am Quellcode zu verfolgen und zu verwalten, mit anderen zusammenzuarbeiten und eine historische Aufzeichnung der Codebasis zu führen.

Bei Git wird auf dem Computer aller Entwickler*innen eine lokale Kopie der Codebasis gespeichert. Diese lokale Kopie, ein sogenanntes *Repository*, enthält die gesamte Historie der Codebasis, einschließlich aller früheren Versionen und Änderungen. Wenn jemand Änderungen am Code vornimmt, können diese Änderungen im lokalen Repository übertragen werden. Wenn mehrere Übertragungen vorgenommen wurden, können diese Änderungen in ein entferntes Repository verschoben werden, das in der Regel auf einem Server gehostet wird (siehe Abbildung 2.1).

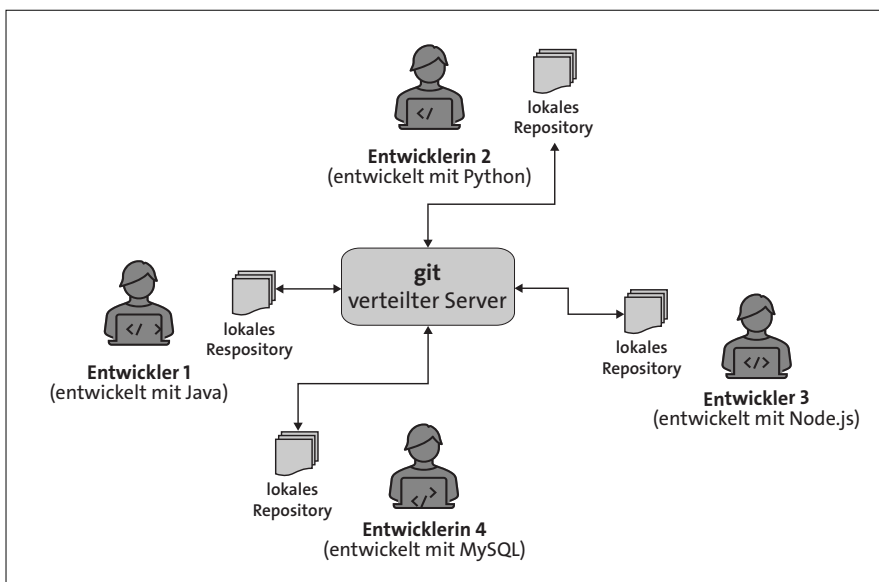


Abbildung 2.1 Git-Prozessablauf

Eines der wichtigsten Merkmale von Git ist die Fähigkeit, Zweige und Merge-Konflikte zu verwalten. Ein *Zweig* (engl. Branch) ist eine separate Kopie der Codebasis, an der unabhängig von der Hauptcodebasis gearbeitet werden kann. So können mehrere Entwickler*innen gleichzeitig an verschiedenen Funktionen oder Fehlerkorrekturen arbeiten, ohne sich gegenseitig in die Quere zu kommen. Wenn Änderungen bereit sind, wieder in die Haupt-Codebasis integriert zu werden, kann Git automatisch alle auftretenden Konflikte erkennen und lösen.

Darüber hinaus ermöglicht Git ein einfaches Rollback zu früheren Versionen der Codebasis sowie die Möglichkeit, die Unterschiede zwischen den Versionen zu erkennen. Git verfügt außerdem über eine große und aktive Gemeinschaft, die eine große Anzahl von Tutorials, Plug-ins und Integrationen zur Verfügung stellt.

Git ist ein *verteiltes Versionskontrollsystem* bei dem alle Mitwirkenden eine lokale Kopie des Hauptspeichers haben. Mit Git können User ein lokales Repository auf ihrem eigenen Computer erstellen, indem sie das entfernte Repository klonen, das sie dann mit einem entfernten, auf einem Server gehosteten Repository verbinden können.

Darüber hinaus können User Änderungen an Dateien in ihrem lokalen Repository vornehmen und diese Änderungen dann übertragen, wodurch ein Schnappschuss des aktuellen Zustands der Dateien gespeichert wird. Diese Übertragungen (engl. Commits) können dann in das entfernte Repository hochgeladen werden, sodass andere User auf die Änderungen zugreifen können.

Git ermöglicht es Usern auch, Änderungen aus dem entfernten Repository in ihr lokales Repository zu ziehen und Zweige zu erstellen, um gleichzeitig an verschiedenen Versionen eines Projekts zu arbeiten. Abbildung 2.2 veranschaulicht den gesamten Ablauf.

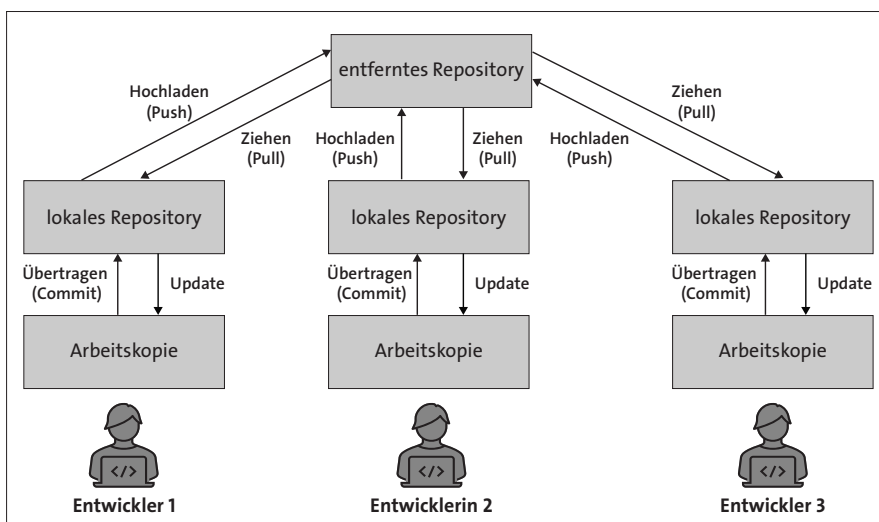


Abbildung 2.2 Funktion von Git

Nachfolgend sind einige der wichtigsten Vorteile von Git aufgeführt:

- Mit Git können Sie die Versionskontrolle einfach handhaben. Sie können Änderungen am Code im Laufe der Zeit verfolgen und bei Bedarf zu früheren Versionen zurückkehren.
- Mit Git können Sie mit anderen zusammenarbeiten. Durch die Verwendung von Git können mehrere Entwickler*innen gleichzeitig an derselben Codebasis arbeiten.
- Mit der Verzweigung ermöglicht Git die parallele Entwicklung verschiedener Funktionen.
- Historie und Auditing sind weitere wichtige Vorteile von Git. Diese Funktionen bieten eine vollständige Historie der am Code vorgenommenen Änderungen.

Da Git ein leistungsfähiges und weit verbreitetes Tool ist, das Entwickler*innen die Zusammenarbeit und Verwaltung von Softwareprojekten erleichtert, ist es in viele SAP-Tools und -Frameworks integriert. SAP Business Application Studio, die Entwicklungsumgebung in SAP BTP, ist bereits mit Git-Befehlen und -Tools vorkonfiguriert. Mit nur wenigen Klicks oder Befehlen können Sie SAP Business Application Studio mit Git-Repositories integrieren.

SAP S/4HANA kommt mit einem Git-fähigen Änderungs- und Transportsystem (engl. Git-enabled Change and Transport System, gCTS). Dieses System ermöglicht es Ihnen, ABAP-Code in GitHub zu speichern und zu verwalten. gCTS nutzt die besonderen Git-Funktionen, einschließlich der Möglichkeit, an lokalen Duplikaten eines zentralen entfernten Git-Repositorys zu arbeiten. Entwicklungsaktivitäten erfolgen in herkömmlichen Editoren, wie z. B. den ABAP Development Tools. Modifikationen an Objekten werden weiterhin in Transportaufträgen dokumentiert. Der aktuelle Status von Objekten kann in entfernte Git-Repositories verschoben werden, was als *Pushing* von Objekten in entfernte Repositories bezeichnet wird, entweder durch Freigabe eines Transportauftrags oder während des Entwicklungsprozesses.

Wenn ABAP-Objekte in das ABAP-System gebracht werden, werden sie von Datenbanktabelleninhalten in Datei- und Ordnerstrukturen umgewandelt, die im lokalen Dateisystem des Anwendungsservers gespeichert werden. Auf diese Datei-Objekte kann dann über Versionierungstools wie Git zugegriffen und sie können in ein entferntes Git-Repository übertragen werden, was die Verteilung von Software über verschiedene Systeme hinweg ermöglicht. Das Abrufen der Software aus dem zentralen Remote-Git-Repository in einem anderen ABAP-System wird als *Pulling* von Objekten aus dem entfernten Repository bezeichnet.

Der gesamte Prozessablauf von gCTS ist in Abbildung 2.3 dargestellt. In Kapitel 3, »DevOps für SAP-On-Premise-Systeme«, erfahren Sie mehr darüber.

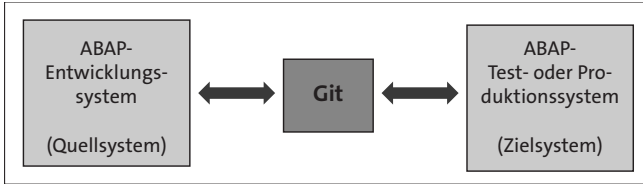


Abbildung 2.3 gCTS-Prozessablauf

Darüber hinaus bieten die CI/CD-Tools von SAP, wie z. B. das Projekt »Piper«, viele gebrauchsfertige Bibliotheken und Tools für die Verwendung von Git. In Kapitel 4, »DevOps mit SAP Business Technology Platform«, erfahren Sie mehr darüber.

2.1.2 GitHub

GitHub ist eine webbasierte Plattform für Versionskontrolle und Zusammenarbeit, die Git als Backend verwendet. Es ermöglicht Ihnen, Code zu hosten und zu überprüfen, Projekte zu verwalten und Software zu erstellen. Außerdem bietet sie Funktionen wie Fehlerverfolgung, Feature Requests, Aufgabenverwaltung und Wikis für jedes Projekt. Es ist auch eine beliebte Plattform für die Entwicklung von Open-Source-Software.

Git vs. GitHub

Es kommt häufig vor, dass Git und GitHub verwechselt werden. Git ist das Tool, während GitHub eine Website ist, auf der Sie Code-Repositories hosten können. GitHub stellt Menschen, die der GitHub-Community beitreten, Git über das Internet zur Verfügung, sodass sie Git nicht auf ihren lokalen Computern installieren müssen.

Lassen Sie uns ein Beispiel betrachten. Angenommen, Sie haben ein paar Videos aufgenommen und mit einem Tool (z. B. Camtasia) bearbeitet. Schließlich laden Sie diese Videos auf OneDrive hoch, wo Sie sie für Ihre Kolleg*innen freigeben, die sie weiter bearbeiten oder kommentieren können. In diesem Fall ist Camtasia Git, während OneDrive GitHub ist.

GitHub ist ein webbasiertes Git-Repository, das alle Funktionen der verteilten Versionskontrolle und Quellcodeverwaltung von Git bietet. Darüber hinaus fügt GitHub seine eigenen Funktionen hinzu, um die Zusammenarbeit leicht und einfach zu gestalten. GitHub ist öffentlich zugänglich. Entwickler*innen auf der ganzen Welt können sich gegenseitig austauschen, interagieren und gegenseitig zu ihrem Quellcode beitragen. Organisationen nutzen GitHub auch, um Quellcodemuster zu veröffentlichen, die von Entwickler*innen als Vorlagen verwendet werden können, um Zeit zu sparen. Auf <http://s-prs.de/v10183001> finden Sie zum Beispiel viele Codebeispiele und Repositories.



Verschiedene Unternehmen

Es ist auch wichtig zu wissen, dass Git und GitHub verschiedenen Unternehmen gehören. Git ist ein Open-Source-Tool, das von der Linux Foundation gepflegt wird. GitHub hingegen wurde 2008 als Einzelunternehmen gegründet, das 2018 von Microsoft übernommen wurde.

Im Folgenden sind einige der wichtigsten Vorteile von GitHub aufgeführt:

- GitHub erleichtert die Teamarbeit und die Zusammenarbeit bei Codierungsprojekten. Unabhängig vom Standort können sich Menschen über das Internet verbinden und ihren Code teilen.
- GitHub bietet eine großartige Option zur Verwaltung von Code-Reviews. Es bietet Tools für die Überprüfung von Code, das Zusammenführen von Pull-Requests und die Verfolgung von Fehlern.
- Durch die Integration mit Git ermöglicht GitHub Versionskontrolle und Zusammenarbeit.
- GitHub bietet eine Plattform für das Hosting und die gemeinsame Nutzung von Dokumentation, Wikis und anderen Ressourcen.
- GitHub bietet einen Cloud-basierten Hosting-Service für Git-Repositories. Es unterstützt auch eine breite Palette von Integrationen mit anderen Tools, wie z. B. CI/CD-Tools.

GitHub kann in SAP BTP verwendet werden, um den Quellcode für auf der Plattform erstellte Anwendungen zu verwalten. Durch die Verwendung von GitHub mit SAP BTP können Sie Ihren Code speichern, verfolgen und freigeben, mit anderen zusammenarbeiten und den Build-, Test- und Bereitstellungsprozess automatisieren.

Die Integration von GitHub mit SAP BTP umfasst in der Regel die folgenden Schritte:

- Sie können ein Repository auf GitHub erstellen, um den Quellcode für Ihre SAP-BTP-Anwendung zu speichern. Dieses Repository kann öffentlich oder privat sein, je nach den Bedürfnissen des Entwicklungsteams.
- GitHub kann mit CI/CD-Tools wie SAP Continuous Integration and Delivery integriert werden, um den Build, das Testen und die Bereitstellung von Anwendungen auf SAP BTP zu automatisieren. Dies ermöglicht Ihnen die kontinuierliche Integration und Auslieferung von Codeänderungen, wodurch der manuelle Aufwand und das Risiko von Fehlern bei der Bereitstellung reduziert werden.
- GitHub bietet den Entwickler*innen eine Plattform, um gemeinsam an Projekten zu arbeiten, Code auszutauschen und sich gegenseitig Feedback zu ihrer Arbeit zu geben. Dies trägt zur Verbesserung der Zusammenarbeit bei und verringert das Risiko von Konflikten zwischen Entwicklern, die an der gleichen Codebasis arbeiten.

- GitHub kann verwendet werden, um den Freigabeprozess für SAP-BTP-Anwendungen zu verwalten, einschließlich der Erstellung und Verwaltung von Freigabezweigen (engl. Release-Branches), der Verwaltung des Versionsverlaufs und der Verfolgung von Problemen und Fehlern.
- GitHub ermöglicht es Ihnen, Code-Reviews durchzuführen, sich gegenseitig Feedback zum Code zu geben und sicherzustellen, dass Codeänderungen überprüft und getestet werden, bevor sie in SAP BTP bereitgestellt werden.

In ähnlicher Weise kann GitHub auch für SAP S/4HANA verwendet werden, um den Quellcode für SAP-Anwendungen zu verwalten und es Ihnen zu ermöglichen, ihren Code zu speichern, zu verfolgen und mit anderen zu teilen.

2.2 Build-Tools

DevOps-Build-Tools sind Softwaretools, die dabei helfen, den Prozess der Softwareerstellung und -freigabe zu automatisieren und damit effizienter und zuverlässiger zu gestalten. Mit diesen Tools können Aufgaben wie die Kompilierung von Code, die Durchführung von Tests, die Paketierung von Software und die Bereitstellung in verschiedenen Umgebungen automatisiert werden.

Der Hauptzweck von Build-Tools besteht darin, die Konsistenz und Zuverlässigkeit von Software-Builds zu gewährleisten und den Zeit- und Arbeitsaufwand für die Erstellung eines neuen Builds zu verringern. So können sich die Entwickler*innen auf das Schreiben von Code konzentrieren, während sich die Build-Tools um den Rest des Build-Prozesses kümmern, einschließlich Abhängigkeitsmanagement, Tests und Bereitstellung.

Zu den beliebtesten DevOps-Build-Tools gehören Maven, Chef und Jenkins. In den folgenden Abschnitten sehen wir uns diese Tools genauer an.

2.2.1 Maven

Maven ist ein von der Apache Software Foundation entwickeltes Open-Source-Tool für Build, Automatisierung und Verwaltung. Das Hauptziel von Maven ist es, mehrere Projekte auf einmal zu erstellen, zu veröffentlichen und bereitzustellen, um das Projektmanagement zu verbessern. Maven ermöglicht es Ihnen, Projekte zusammen mit ihren Abhängigkeiten und ihrer Dokumentation zu erstellen und zu verwalten, indem sie ein sogenanntes *Projektobjektmodell* (POM) verwenden. Mit Maven können Sie mehrere Projekte als bereitstellbare Artefakte (JAR-Dateien, WAR-Dateien usw.) erstellen.

Maven wurde mit Java geschrieben und diente ursprünglich der Erstellung von Java-Projekten. Derzeit kann Maven verwendet werden, um Projekte zu erstellen, die in Java, Scala, Ruby, C# und mehr geschrieben wurden.

Maven verwendet die Datei **pom.xml**, um die Struktur des Projekts, die Abhängigkeiten und die Build-Einstellungen zu definieren. Diese Datei wird verwendet, um den Build-Prozess zu konfigurieren und die für das Projekt benötigten Abhängigkeiten zu verwalten. Maven bietet außerdem eine Reihe von Standardkonventionen für die Erstellung von Java-Projekten, die das Verständnis und die Wartung des Build-Prozesses erleichtern. Maven verfügt außerdem über ein großes Repository an vorgefertigten Bibliotheken, die einfach in ein Projekt eingebunden werden können, wodurch der Zeit- und Arbeitsaufwand für die Einrichtung eines neuen Projekts reduziert wird.

Wie bereits erwähnt, verwendet Maven eine POM-Datei, um Projektabhängigkeiten zu verwalten, das Projekt zu erstellen und bereitzustellen und die Projektdokumentation zu generieren. Es liest die POM-Datei, löst Abhängigkeiten auf, führt Plug-ins aus, generiert die Projektdokumentation und stellt das Projekt bereit. Wenn Sie einen Maven-Befehl ausführen, liest es die POM-Datei und verwendet diese Informationen, um die angegebene Aufgabe auszuführen.

Die grundlegenden Schritte der Funktionsweise von Maven sind die folgenden (siehe Abbildung 2.4):

- ❶ Maven liest die POM-Datei, die Informationen über das Projekt enthält, wie z. B. seine Abhängigkeiten, Build-Einstellungen und Plug-ins.
- ❷ Maven löst Abhängigkeiten auf, indem es zunächst prüft, ob die Abhängigkeiten im lokalen Repository vorhanden sind. Wenn alle Abhängigkeiten in den lokalen Repositories vorhanden sind, dann beginnt Maven mit der Erstellung des Projekts.
- ❸ Wenn irgendwelche Abhängigkeiten nicht in lokalen Repositories vorhanden sind, dann lädt Maven die notwendigen Bibliotheken und Plug-ins aus entfernten Repositories herunter, wie z. B. dem Maven Central Repository.
- ❹ Nach dem Herunterladen werden alle Abhängigkeiten in lokalen Repositories gespeichert, sodass sie beim nächsten Mal nicht erneut heruntergeladen werden müssen.
- ❺ Maven verwendet die in der POM-Datei angegebenen Plug-ins, um verschiedene Aufgaben auszuführen, wie z. B. die Kompilierung des Codes, die Durchführung von Tests und die Erstellung einer JAR- oder WAR-Datei.

Maven kann auch zur Erstellung von Projektdokumentation, wie JavaDocs und Testberichten, verwendet werden. Das Tool kann auch verwendet werden, um das Projekt auf einem entfernten Server bereitzustellen, z. B. auf einem Anwendungsserver oder einem entfernten Repository.

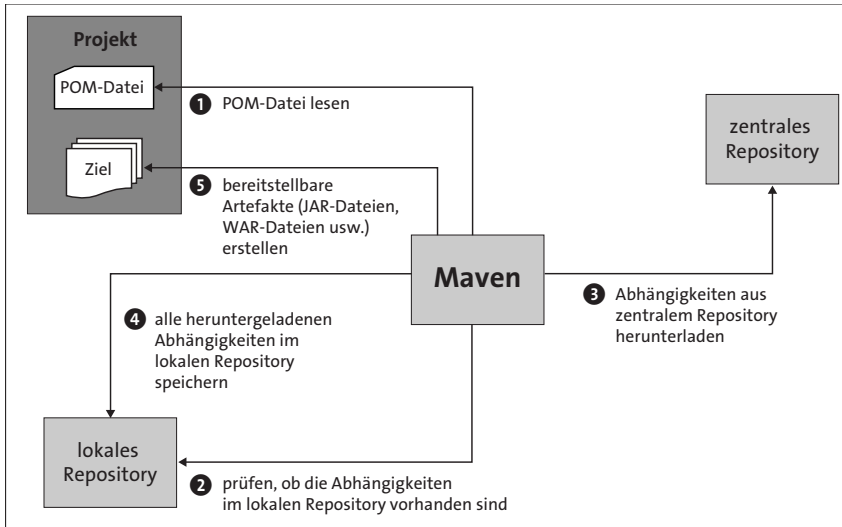


Abbildung 2.4 Funktionsweise von Maven

Maven bietet eine Reihe von Vorteilen für Entwicklungsteams, darunter die nachfolgenden:

■ Konsistenter Build-Prozess

Maven definiert einen Standard-Build-Prozess für Java-basierte Projekte und hilft dabei, sicherzustellen, dass alle Builds konsistent und wiederholbar sind. Dies reduziert das Risiko von Build-Fehlern und erleichtert die Wartung und Aktualisierung von Projekten im Laufe der Zeit.

■ Verwaltung von Abhängigkeiten

Maven erleichtert die Verwaltung von Projektabhängigkeiten, einschließlich des Herunterladens und Verwaltens externer Bibliotheken und der Sicherstellung, dass alle Abhängigkeiten auf dem neuesten Stand und miteinander kompatibel sind.

■ Wiederverwendbare Build-Komponenten

Maven unterstützt die Erstellung und Wiederverwendung von Build-Komponenten, wie z. B. Plug-ins, die über mehrere Projekte hinweg verwendet werden können. Dies trägt dazu bei, den Zeit- und Arbeitsaufwand für die Einrichtung neuer Projekte zu reduzieren und die Konsistenz von Builds über Projekte hinweg zu verbessern.

■ Automatisiertes Testen

Maven lässt sich in Test-Frameworks wie JUnit integrieren und unterstützt die automatisierte Ausführung von Tests, um sicherzustellen, dass Codeänderungen vor dem Einsatz gründlich getestet werden.

■ Kontinuierliche Integration

Maven kann in CI/CD-Tools wie Jenkins integriert werden, um den Build- und Bereitstellungsprozess zu automatisieren und die kontinuierliche Integration und Auslieferung von Codeänderungen zu ermöglichen.

■ Projektinformationen und Berichte

Maven liefert detaillierte Informationen über ein Projekt, einschließlich der Abhängigkeiten, der Build-Historie und der Testergebnisse, um den Status und den Fortschritt des Projekts besser zu verstehen.

Maven kann zur Erstellung und Verwaltung der Abhängigkeiten eines Java-Projekts verwendet werden, das mit SAP S/4HANA oder SAP BTP interagiert. In SAP S/4HANA kann Maven verwendet werden, um benutzungsdefinierte Java-Anwendungen zu erstellen und bereitzustellen, die in das SAP-S/4HANA-System integriert sind. Maven kann verwendet werden, um die Abhängigkeiten dieser Anwendungen zu verwalten, wie z. B. das SAP Cloud SDK für Java, das Bibliotheken für die Verbindung mit SAP S/4HANA und anderen SAP-Cloud-Plattformen bereitstellt.

In SAP BTP kann Maven verwendet werden, um Java-basierte Anwendungen zu erstellen und auf der Cloud-Plattform bereitzustellen. Maven kann auch verwendet werden, um die Abhängigkeiten dieser Anwendungen zu verwalten, wie z. B. das SAP Cloud SDK für Java, das Bibliotheken für die Verbindung mit SAP-BTP-Diensten wie der Cloud-Foundry-Umgebung bereitstellt.

Maven kann auch verwendet werden, um die Abhängigkeiten einer Anwendung zu verwalten, die die SAP-Java-Connector-Bibliothek verwendet, um sich mit einem SAP-System zu verbinden. Diese Bibliothek bietet eine Reihe von Java-APIs für die Verbindung mit SAP-Systemen und kann einfach zu einem Maven-Projekt als Abhängigkeit hinzugefügt werden.

2.2.2 Chef

Chef ist ein Open-Source-Konfigurationsmanagement-Tool, das häufig in DevOps-Umgebungen verwendet wird, um die Bereitstellung und Verwaltung der Infrastruktur zu automatisieren. Es verwendet ein Konzept namens *Infrastructure as Code* (IaC), um den gewünschten Zustand eines Systems zu definieren, und verwendet diesen Code dann zur automatischen Konfiguration und Verwaltung von Servern, Netzwerken und anderen Infrastrukturkomponenten.

Chef verwendet eine domänenspezifische Sprache namens Ruby zur Definition von *Rezepten*, d. h. von Anweisungen zur Konfiguration und Verwaltung verschiedener Teile der Infrastruktur. Diese Rezepte können in *Kochbüchern* gruppiert werden, die von verschiedenen Teams und Projekten gemeinsam genutzt und wiederverwendet werden können.

Eine der wichtigsten Funktionen von Chef ist die Verwendung von *Ressourcen*, d. h. von vorbereiteten Codeblöcken, die zur Ausführung allgemeiner Aufgaben wie der Installation von Software, der Erstellung von Usern und der Konfiguration von Services verwendet werden können. Dadurch können sich Entwickler*innen auf den gewünschten Zustand ihrer Infrastruktur konzentrieren, anstatt sich um die spezifischen Befehle zu kümmern, die für die Konfiguration erforderlich sind.

Chef umfasst auch eine Verwaltungskonsole namens *Chef Manage*, die eine webbasierte Schnittstelle für die Verwaltung von Kochbüchern, Knoten und anderen Aspekten der Infrastruktur bietet. Sie enthält auch eine Berichts- und Analysefunktion namens *Chef Insights*, die es Teams ermöglicht, Probleme mit ihrer Infrastruktur zu überwachen und zu beheben.

Chef kann in SAP-Umgebungen eingesetzt werden, um die Bereitstellung, Konfiguration und Verwaltung von SAP-Systemen und deren Abhängigkeiten zu automatisieren. Chef kann in SAP-Umgebungen unter anderem auf die folgenden Arten eingesetzt werden:

- Chef kann verwendet werden, um die Bereitstellung neuer SAP-Systeme zu automatisieren, einschließlich der Installation der erforderlichen Software und der Konfiguration des Systems, um die spezifischen Anforderungen des Unternehmens zu erfüllen.
- Chef kann verwendet werden, um die Abhängigkeiten von SAP-Systemen, wie Datenbanken, Middleware und andere Anwendungen, zu verwalten. Es kann die Installation und Konfiguration dieser Abhängigkeiten automatisieren und sicherstellen, dass sie in verschiedenen Umgebungen einheitlich konfiguriert sind.
- Chef kann verwendet werden, um die Sicherung und Wiederherstellung von SAP-Systemen zu automatisieren, einschließlich der Konfiguration von Sicherungsplänen und der Erstellung von Notfallwiederherstellungsplänen.

2.2.3 Jenkins

Jenkins ist ein in Java geschriebener Open-Source-Automatisierungsserver, der die Automatisierung des Softwareentwicklungsprozesses unterstützt. Er wird verwendet, um Softwareprojekte kontinuierlich zu erstellen, zu testen und bereitzustellen. Jenkins kann verwendet werden, um Aufgaben wie den Build, das Testen und das Bereitstellen von Codeänderungen zu automatisieren, sowie zur Integration mit anderen Tools im Softwareentwicklungsprozess wie in Versionskontrollsystemen und Problemverfolgungssystemen.

Jenkins ist eine der am weitesten verbreiteten DevOps-Anwendungen auf der Welt. Jenkins wurde ursprünglich 2005 unter dem Namen Hudson von Kohsuke Kawaguchi, einem Ingenieur bei Sun Microsystems, entwickelt. Es handelte sich um ein

Open-Source-Tool für die kontinuierliche Integration, mit dem der Build- und Testprozess für die Softwareentwicklung automatisiert werden sollte. Nach der Übernahme von Sun Microsystems durch Oracle im Jahr 2010 wurde das Hudson-Projekt abgespalten, und Jenkins wurde als von der Community betriebene Alternative entwickelt.

Im Laufe der Jahre hat sich Jenkins zu einer beliebten Plattform für CI/CD entwickelt und wird von vielen Entwickler*innen zur Automatisierung von Softwareentwicklungsprozessen eingesetzt. Es hat eine große und aktive Gemeinschaft von Mitwirkenden und Plug-ins, die seine Funktionalität erweitern. Im Jahr 2023 ist Jenkins ein führendes Open-Source-CI/CD-Tool, das von Unternehmen aller Größenordnungen zur Automatisierung und Rationalisierung ihrer Softwareentwicklungsabläufe eingesetzt wird.

Jenkins ist mit einem Quellcode-Verwaltungstool wie GitHub integriert. Durch die ständige Überwachung des Code-Repositorys auf Änderungen kann Jenkins automatisch Code erstellen und testen, sobald eine Codeaktualisierung in das Repository hochgeladen wird. Schauen wir uns an, wie Jenkins funktioniert (siehe Abbildung 2.5):

- ❶ Jenkins lässt sich mit verschiedenen Quellcode-Verwaltungstools wie GitHub integrieren, um Code für den Build und die Prüfung abzurufen. Der Ablauf beginnt, wenn ein Code-Commit erfolgt.
- ❷ Jedes Mal, wenn Code-Commits erfolgen, löst Jenkins einen Build aus. Jenkins kann auch so konfiguriert werden, dass der Code in bestimmten Intervallen erstellt wird.
- ❸ Zur Ausführung des Builds, der entweder durch ein Code-Commit ausgelöst oder durch einen geplanten Job gestartet wird, checkt Jenkins den Code aus und führt eine Reihe von Aktivitäten durch: Kompilieren des Codes, Ausführen von Unit-Tests, Generieren von Artefakten und so weiter.
- ❹ Während der Ausführung des Builds kann Jenkins auch automatisierte Tests durchführen, um die Qualität des Codes zu überprüfen und sicherzustellen, dass er den erforderlichen Standards entspricht.
- ❺ Nach Abschluss des Builds und Tests liefert Jenkins über die Benutzeroberfläche, E-Mail-Benachrichtigungen oder die Integration mit anderen Tools Feedback zu den Build- und Testergebnissen.
- ❻ Sie können Jenkins auch so konfigurieren, dass der Code in Ihrer Produktionsumgebung oder einer anderen Umgebung bereitgestellt wird, sobald die Build- und Testprozesse erfolgreich abgeschlossen sind.

Während des gesamten Prozesses verfolgt Jenkins die Build-Historie und stellt detaillierte Berichte zur Verfügung, die es einfacher machen, Probleme zu erkennen und zu beheben. Die Plug-in-Architektur von Jenkins ermöglicht eine einfache Anpassung und Integration mit anderen Tools zur Unterstützung spezifischer Anforderungen.

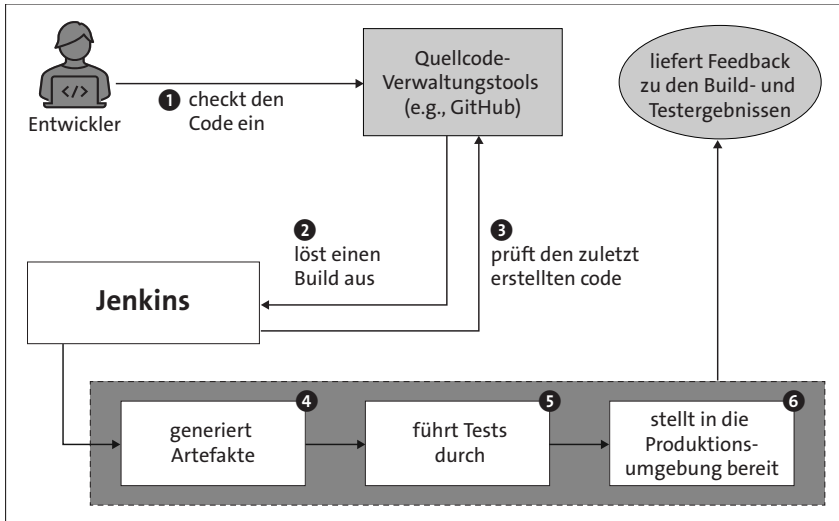


Abbildung 2.5 Funktionsweise von Jenkins

Im Folgenden sind einige der wichtigsten Vorteile von Jenkins aufgeführt:

- **Automatisierung**

Jenkins automatisiert den Build-, Test- und Bereitstellungsprozess und reduziert so den manuellen Aufwand und das Risiko menschlicher Fehler.

- **Geschwindigkeit und Effizienz**

Jenkins ermöglicht die kontinuierliche Integration und Auslieferung von Codeänderungen, wodurch der Entwicklungszyklus beschleunigt und die Effizienz verbessert wird.

- **Skalierbarkeit**

Jenkins ist hoch skalierbar und kann eine große Anzahl von Builds und Bereitstellungen verarbeiten, wodurch es sich für den Einsatz in großen Entwicklungsteams eignet.

- **Anpassungsfähigkeit**

Jenkins bietet eine breite Palette von Plug-ins und APIs, die es Entwicklungsteams ermöglichen, die Funktionalität an ihre spezifischen Anforderungen anzupassen und zu erweitern.

- **Verbesserte Zusammenarbeit**

Jenkins ermöglicht Teams eine effektivere Zusammenarbeit, indem es eine zentralisierte Plattform für CI/CD bereitstellt.

- **Verbesserter Überblick**

Jenkins bietet einen zentralen Überblick über den Build-, Test- und Bereitstellungsprozess und erleichtert so die Identifizierung und Behebung von Problemen.

■ Geringeres Risiko

Durch die Automatisierung des Bereitstellungsprozesses trägt Jenkins dazu bei, das Risiko von Fehlern und Ausfallzeiten zu verringern, was zu qualitativ hochwertigeren Anwendungen führt.

Jenkins kann zusammen mit SAP BTP als CI/CD-Tool verwendet werden, um den Build, das Testen und die Bereitstellung von Anwendungen auf SAP BTP zu automatisieren. Es kann mithilfe von Plug-ins und APIs in SAP BTP integriert werden, um den Bereitstellungsprozess von Anwendungen zu automatisieren und die Gesamteffizienz des Entwicklungszyklus zu verbessern. Darüber hinaus kann Jenkins verwendet werden, um automatische Builds und Bereitstellungen auf der Grundlage von Ereignissen wie Codeänderungen auszulösen, was es zu einem leistungsstarken Tool für DevOps-Teams macht, die mit SAP BTP arbeiten.

Jenkins kann in das Projekt »Piper« integriert werden, um den Build, das Testen und die Bereitstellung von SAP-BTP-Anwendungen zu automatisieren. Durch die Integration von Jenkins in das Projekt »Piper« können Sie CI/CD-Pipelines konfigurieren, die den Prozess des Builds, des Testens und der Bereitstellung von Anwendungen automatisieren und so den Entwicklungsprozess effizienter und zuverlässiger machen. Darüber hinaus ist Jenkins in der Lage, automatisierte Builds und Bereitstellungen zu initiieren, die durch Ereignisse wie Änderungen im Code ausgelöst werden, wodurch eine reibungslose Integration in den Bereitstellungsprozess von Projekt »Piper« gewährleistet wird. In Kapitel 4, »DevOps mit SAP Business Technology Platform«, erfahren Sie mehr darüber.

Jenkins wird auch in SAP S/4HANA intensiv genutzt, um den DevOps-Zyklus zu automatisieren. Es kann mit gCTS und anderen ABAP-Tools integriert werden. In Kapitel 3, »DevOps für SAP-On-Premise-Systeme«, finden Sie dazu weitere Informationen.

2.3 Testautomatisierungstools

CI basiert auf dem grundlegenden Prinzip, dass ein Build überprüfbar sein muss. Es ist von entscheidender Bedeutung, über ein objektives Mittel zu verfügen, um festzustellen, ob ein bestimmtes Build für die nächste Phase des Build-Prozesses bereit ist, und der bequemste Ansatz, um dies zu erreichen, ist die Verwendung von automatisierten Tests. Wenn nicht mit automatisierten Tests gearbeitet wird, müssen zahlreiche Build-Artefakte manuell aufbewahrt und getestet werden, was dem Wesen der CI zuwiderläuft.

DevOps-Testautomatisierungstools sind eine Reihe von Softwaretools, die den Testprozess in einer DevOps-Umgebung automatisieren. Sie werden verwendet, um wiederholbare Tests für Anwendungen durchzuführen, um Fehler zu identifizieren, die

Qualität der Software zu verbessern und den Zeit- und Arbeitsaufwand für manuelle Tests zu reduzieren. Die wichtigsten Verwendungszwecke von Testautomatisierungstools sind die folgenden:

- *Regressionstests*

Automatisierte Testtools können zur Durchführung von Regressionstests verwendet werden, d. h. zum Testen von Softwareänderungen, um sicherzustellen, dass die bestehende Funktionalität nicht beeinträchtigt wurde.

- *Funktionstests*

Bei Funktionstests werden einzelne Funktionen oder Merkmale einer Anwendung, z. B. Eingaben, Ausgaben, Benutzeroberflächen und Arbeitsabläufe, auf ihre korrekte Funktionsweise hin überprüft. Diese Art von Tests hilft bei der Identifizierung von Mängeln und Fehlern in der Anwendung und stellt sicher, dass sie sich für die User wie erwartet verhält. Für die Durchführung und Automatisierung von Funktionstests können automatisierte Testtools eingesetzt werden.

- *Leistungstests*

Das Ziel von Leistungstests ist es, Engpässe zu erkennen, Antwortzeiten, Stabilität und Skalierbarkeit zu messen und sicherzustellen, dass die Anwendung die Leistungsziele und -erwartungen erfüllt. Bei den Leistungstests wird die Anwendung unter verschiedenen Belastungsstufen getestet, z. B. bei starkem User-Verkehr oder großen Datenmengen, um festzustellen, wie sie sich verhält und um etwaige Leistungsprobleme zu erkennen. Automatisierte Testwerkzeuge können für Leistungstests verwendet werden, um die Reaktionszeit, Stabilität und Skalierbarkeit einer Anwendung unter verschiedenen Belastungen zu messen.

- *Sicherheitstests*

Automatisierte Testtools können zur Durchführung von Sicherheitstests verwendet werden, d. h. zum Testen von Softwareanwendungen, um Sicherheitsschwachstellen zu ermitteln und sich vor Sicherheitsbedrohungen zu schützen. Bei Sicherheitstests wird die Anwendung oder das System unter Sicherheitsaspekten getestet, wobei Techniken wie Penetrationstests, Bedrohungsmodellierung und Schwachstellen-Scans zum Einsatz kommen. Der Testprozess umfasst die Identifizierung potenzieller Sicherheitsbedrohungen, die Simulation realer Angriffe und die Bewertung der Fähigkeit der Anwendung oder des Systems, sich vor diesen Angriffen zu schützen.

Zu den beliebtesten DevOps-Testautomatisierungstools gehören Selenium, JUnit, SonarQube und Jenkins. In den folgenden Abschnitten werfen wir einen detaillierten Blick auf jedes dieser Tools.

2.3.1 Selenium

Selenium ist ein Open-Source-Tool, das zum Testen von Webanwendungen verwendet wird. Es bietet die Möglichkeit, Tests zu schreiben, die User-Interaktionen mit einer Webanwendung simulieren und überprüfen, ob sich die Anwendung wie erwartet verhält. Mit Selenium können Sie Tests in mehreren Programmiersprachen schreiben, darunter Java, C#, Python und Ruby, und Tests mit verschiedenen Browsern ausführen, darunter Chrome, Firefox, Safari und Edge. Selenium kann für eine Vielzahl von Testaufgaben verwendet werden, darunter funktionale Tests, Regressionstests und End-to-End-Tests. Es ist ein beliebtes Tool in der DevOps-Toolchain und wird häufig für kontinuierliche Tests und CI verwendet.

Selenium wurde 2004 von Jason Huggins, einem Softwareingenieur bei Thoughtworks, entwickelt. Er schuf Selenium als Tools zur Automatisierung von Tests für eine Webanwendung, an der er arbeitete. Ursprünglich war Selenium eine JavaScript-Bibliothek, die für die Automatisierung von Interaktionen mit einem Webbrowser verwendet werden konnte. Im Laufe der Zeit entwickelte es sich zu einem vollwertigen Test-Framework mit einer Vielzahl von APIs für die Automatisierung von Webbrowsern.

Im Jahr 2006 wurde Selenium an das openQA-Projekt gespendet und zu einem Open-Source-Tool. Das Projekt gewann eine große Anhängerschaft, und die Selenium-Community hat eine Reihe von Verbesserungen und Ergänzungen beigesteuert, darunter die Unterstützung zusätzlicher Programmiersprachen, eine bessere Integration mit anderen Tools und eine verbesserte Browserkompatibilität.

In den letzten Jahren hat sich Selenium zu einem unverzichtbaren Tool in der DevOps-Toolchain entwickelt, das für kontinuierliche Tests und kontinuierliche Integration eingesetzt wird. Die Entwicklung von Selenium Grid, mit dem Tests parallel auf mehreren Rechnern ausgeführt werden können, hat die Skalierbarkeit und Leistung von Selenium-basierten Tests weiter verbessert. Da Webanwendungen immer komplexer und wichtiger werden, ist Selenium zu einem unverzichtbaren Werkzeug für die Sicherstellung der Qualität und Zuverlässigkeit von Webanwendungen geworden.

Selenium ist nicht ein einzelnes Tool, sondern eine Kombination aus mehreren Tools, die zusammenarbeiten. Seine Hauptkomponenten sind in Abbildung 2.6 dargestellt. Schauen wir uns diese Selenium-Komponenten genauer an.

Selenium Remote Control (RC), auch bekannt als Selenium 1, ist ein veraltetes Tool in der Selenium-Suite, das zum Schreiben automatisierter Tests für Webanwendungen verwendet wurde. Es ermöglichte das Schreiben von Tests in mehreren Programmiersprachen, wie Java, C# und Python, und die Ausführung dieser Tests in einer Vielzahl von Browsern. Selenium RC war das Hauptprojekt von Selenium. Heute ist Selenium RC durch die WebDriver-API ersetzt worden und wird nicht mehr aktiv ge-

pfllegt. Es wird empfohlen, Selenium WebDriver für neue Testautomatisierungsprojekte zu verwenden.

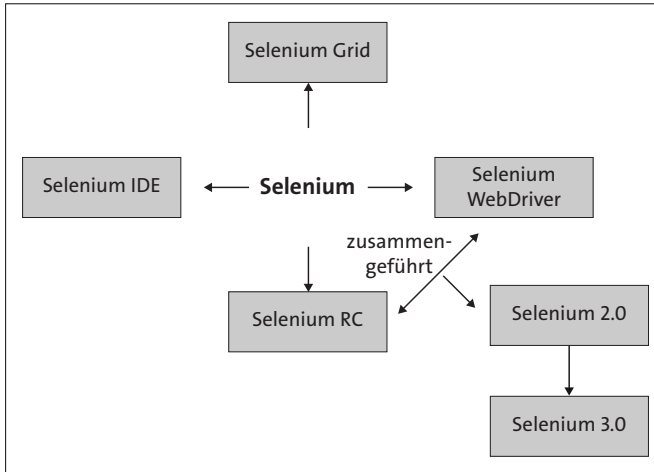


Abbildung 2.6 Verschiedene Komponenten von Selenium

Selenium WebDriver, auch bekannt als Selenium 2, ist der Nachfolger von Selenium RC und die Kernkomponente von Selenium. Er ermöglicht die Integration mit Webbrowsern und die Automatisierung von Benutzungsinteraktionen mit einer Webanwendung. Selenium WebDriver ermöglicht es, Tests in verschiedenen Programmiersprachen zu schreiben, darunter Java, C# und Python, und sie in einem echten Webbrowser auszuführen. Er bietet eine einfache API für die Interaktion mit Webelementen auf einer Seite, wie z. B. das Anklicken von Schaltflächen, das Ausfüllen von Formularen und das Überprüfen von Text. Die WebDriver-API wurde entwickelt, um moderne, dynamische Webanwendungen zu unterstützen, und kann für Funktions-, Regressions- und Lasttests verwendet werden.

Selenium 2.0 entstand nach der Verschmelzung von Selenium RC und Selenium WebDriver zu Selenium WebDriver. Die neueste Version von Selenium ist 3.0, die einige neue Features und Funktionalitäten enthält.

Selenium IDE (integrierte Entwicklungsumgebung, engl. Integrated Development Environment) ist ein browserbasiertes Aufzeichnungs- und Wiedergabetool zum Schreiben und Testen von Selenium-Skripten. Es bietet eine einfache und benutzungsfreundliche Schnittstelle zum Erstellen und Ausführen von Testfällen, was es nicht-technischen Usern erleichtert, mit dem automatisierten Testen zu beginnen. Mit Selenium IDE können User ihre Interaktionen mit einer Webanwendung aufzeichnen, woraufhin die entsprechenden Selenium-Befehle in Echtzeit generiert werden. Das generierte Skript kann dann durch Hinzufügen von Assertions und Kontrollflussanweisungen verändert und erweitert werden, sodass auch komplexe Szenarien getestet werden können.

Selenium IDE ist ein großartiges Tool für die schnelle Erstellung einfacher Testfälle und Prototypen, aber es ist in Bezug auf die Anpassung und Skalierbarkeit im Vergleich zum Schreiben von Tests in einer Programmiersprache mit Selenium WebDriver begrenzt.

Selenium Grid ist ein Tool, mit dem mehrere Instanzen von Selenium-WebDriver-Tests parallel auf verschiedenen Rechnern und/oder Browsern ausgeführt werden können. Dies ermöglicht browserübergreifende Tests und die Verteilung von Tests auf eine große Testumgebung. Es ermöglicht Unternehmen, den Testprozess zu beschleunigen, indem es die Zeit reduziert, die für die Ausführung einer großen Testsuite auf einem einzigen Rechner benötigt wird.

Selenium Grid besteht aus einem Hub und Knoten (engl. Nodes). Der *Hub* fungiert als zentraler Kommunikationspunkt und leitet Testbefehle an die *Knoten* weiter, die die Tests auf bestimmten Browsern ausführen. Jeder Knoten ist für einen bestimmten Browser und ein bestimmtes Betriebssystem konfiguriert, und die Tests werden auf den Knoten parallel ausgeführt, was die Gesamtgeschwindigkeit der Tests erhöht.

Durch den Einsatz von Selenium Grid können Unternehmen ihre Testumgebung einfach skalieren und Tests auf einer Vielzahl von Browsern und Plattformen ausführen, um einen umfassenden Überblick über die Kompatibilität und Leistung der Anwendung in verschiedenen Umgebungen zu erhalten.

Selenium bietet die folgenden Vorteile für Automatisierungstests:

- Selenium unterstützt mehrere Betriebssysteme und Browser, was es zu einem vielseitigen Tool für das Testen von Webanwendungen macht.
- Es ist ein Open-Source-Tool, d. h., es kann kostenlos verwendet werden. Das macht es im Vergleich zu anderen kommerziellen Tools zu einer kostengünstigen Option für die Automatisierung von Tests.
- Selenium kann leicht in CI/CD-Pipelines integriert werden, sodass automatisierte Tests Teil des Softwareentwicklungsprozesses werden.
- Es hat eine große und aktive Gemeinschaft von Entwickler*innen und Usern, was bedeutet, dass es eine Fülle von Ressourcen und Unterstützung zu diesem Tool gibt.
- Selenium unterstützt mehrere Programmiersprachen, darunter Java, Python, C# und Ruby, und ist damit für ein breites Spektrum von Entwickler*innen zugänglich.
- Es kann durch die Verwendung von Plug-ins und Erweiterungen angepasst und erweitert werden, was es zu einem flexiblen Tool für die Automatisierung einer Vielzahl von Tests macht.
- Durch die Automatisierung von Tests kann Selenium dazu beitragen, die Testabdeckung einer Anwendung insgesamt zu verbessern und so Probleme schneller und effizienter zu erkennen und zu beheben.

**Selenium funktioniert nur im Browser**

Schließlich ist es wichtig zu wissen, dass Selenium nicht mit desktopbasierten SAP-Systemen funktioniert. Es handelt sich um eine browserbasierte API, die außerhalb eines Browsers nicht funktioniert.

Eine der Hauptanwendungen von Selenium in SAP ist die Testautomatisierung für SAP-Fiori-Apps. Sie können Selenium auch in SAP S/4HANA Cloud für die Automatisierung von Tests webbasierter SAP-S/4HANA-Anwendungen verwenden. Selenium kann zur Automatisierung verschiedener Arten von Tests verwendet werden, z. B. UI-Tests, Regressionstests, Integrationstests, End-to-End-Tests, Leistungstests, datengetriebene Tests und Sicherheitstests. Da Selenium jedoch nur das Testen der Benutzeroberfläche und webbasierter Anwendungen automatisiert, kann es nicht für SAP-On-Premise-Systeme verwendet werden.

Selenium kann für Automatisierungstests in SAP BTP verwendet werden. Um die Testautomatisierung mit Selenium zu implementieren, können Sie es durch die Installation von Plug-ins in SAP Business Application Studio integrieren. Außerdem können Sie die Testskripte in einer Programmiersprache Ihrer Wahl schreiben. Die Ausführung der Tests kann entweder manuell erfolgen oder in die CI/CD-Pipeline integriert werden. Die Ergebnisse der Tests können analysiert werden, um Probleme mit Ihren SAP-BTP-Anwendungen zu identifizieren.

2.3.2 JUnit

JUnit ist ein Test-Framework für die Programmiersprache Java. Es wird zum Schreiben und Ausführen von Tests für Java-Anwendungen verwendet. JUnit bietet eine einfache und effektive Möglichkeit zum Schreiben und Ausführen von *Unit-Tests*, d. h. kleinen, gezielten Tests, die einzelne Codeeinheiten überprüfen. JUnit bietet Annotationen und Assertions, die das Schreiben von Tests erleichtern, und lässt sich gut in Entwicklungstools wie IDEs und CI-Systeme integrieren.

JUnit wurde in den späten 1990er-Jahren von Kent Beck und Erich Gamma entwickelt. JUnit wurde von der xUnit-Architektur inspiriert und zielte darauf ab, den Testprozess für Java-Anwendungen zu vereinfachen. JUnit Version 1.0 wurde 1998 veröffentlicht und gewann schnell an Popularität unter Java-Entwickler*innen. Es wurde so konzipiert, dass es einfach, effizient und benutzungsfreundlich ist, sodass es für Menschen aller Qualifikationsstufen zugänglich ist. Im Laufe der Jahre wurde JUnit mehrfach aktualisiert und verbessert, u. a. durch die Ergänzung um neue Funktionen und eine bessere Unterstützung komplexer Testszenarien.

Mit der 2006 veröffentlichten Version 4.0 von JUnit wurden wichtige Änderungen am Framework eingeführt, darunter Annotationen und die Unterstützung von Test-suiten, mit denen mehrere Tests gleichzeitig organisiert und ausgeführt werden können.

JUnit 5, das 2017 veröffentlicht wurde, markierte einen wichtigen Meilenstein in der Entwicklung von JUnit. JUnit 5 führte eine modulare Architektur und Unterstützung für Java 8 ein, was die Verwendung und Integration mit anderen Tools und Bibliotheken erleichtert. Außerdem wurden neue Funktionen hinzugefügt, wie z. B. die Möglichkeit, Tests parallel auszuführen, und die Unterstützung für dynamische Tests, mit denen Tests zur Laufzeit generiert werden können.

Heute ist JUnit eines der am weitesten verbreiteten Test-Frameworks für Java-Anwendungen und hat eine große Entwickler*innen- und Nutzer*innengemeinschaft. Seine kontinuierliche Weiterentwicklung stellt sicher, dass es ein wertvolles Tool zur Verbesserung der Qualität und Zuverlässigkeit von Java-Anwendungen bleibt.

JUnit-Tests werden in der Regel automatisch als Teil des Entwicklungsprozesses ausgeführt, sodass die Probleme schnell erkannt und behoben werden können, bevor sie zu größeren Problemen werden. Dies trägt dazu bei, die Gesamtqualität und Stabilität einer Anwendung zu verbessern und den Zeit- und Arbeitsaufwand für manuelle Tests zu verringern. JUnit ist in der Java-Entwickler*innengemeinde weit verbreitet und eine beliebte Wahl für die Automatisierung von Tests in verschiedenen Anwendungen.

JUnit funktioniert, indem in Java geschriebene Tests gegen einzelne Codeeinheiten ausgeführt werden. Die grundlegenden Schritte des Prozesses sind wie folgt (siehe Abbildung 2.7):

1. Testfälle schreiben

Testfälle werden in Java unter Verwendung von JUnit-Annotationen und Assertions geschrieben, um zu definieren, was getestet werden soll und welche Ergebnisse zu erwarten sind.

2. Testfälle ausführen

Der JUnit-Test-Runner führt die Testfälle aus und sammelt die Ergebnisse. Der Test-Runner kann von der Kommandozeile, von einer IDE oder von einem Build-Tool wie Maven oder Gradle ausgeführt werden.

3. Testergebnisse auswerten

Der Test-Runner wertet die Ergebnisse der Testfälle aus, um festzustellen, ob sie bestanden haben oder fehlgeschlagen sind. Wenn ein Test fehlschlägt, wird eine Fehlermeldung generiert, die Auskunft darüber gibt, was schiefgelaufen ist und warum der Test fehlgeschlagen ist.

4. Über Testergebnisse berichten

Der Test-Runner erstellt einen Bericht mit den Testergebnissen, einschließlich Details zu bestandenen und fehlgeschlagenen Tests, Fehlern oder Ausnahmen, die während des Tests aufgetreten sind, und anderen relevanten Informationen.

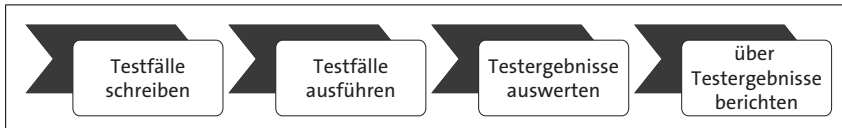


Abbildung 2.7 JUnit-Arbeitsablauf

JUnit unterstützt mehrere Funktionen, die das Schreiben und Ausführen von Tests erleichtern, darunter Annotationen zur Definition von Tests und Testumgebungen, Methoden für Aufbau und Abbau (engl. Setup and Teardown) sowie Test-Listener zur Überwachung der Testausführung. Darüber hinaus bietet JUnit Berichtstools für die Darstellung von Testergebnissen in einer Vielzahl von Formaten.

Die Verwendung von JUnit als Test-Framework für Java-Anwendungen bietet mehrere Vorteile:

- JUnit ermöglicht es Ihnen, den Testprozess zu automatisieren, Zeit und Ressourcen freizusetzen und das Risiko manueller Testfehler zu verringern.
- Es stellt sicher, dass Ihr Code wie beabsichtigt funktioniert, indem es Fehler und Probleme früh im Entwicklungsprozess aufspürt und die Gesamtqualität Ihrer Software verbessert.
- JUnit liefert detaillierte Informationen darüber, warum ein Test fehlgeschlagen ist, was das Auffinden und Beheben von Fehlern in Ihrem Code erleichtert.
- JUnit-Tests dienen als Dokumentation des erwarteten Verhaltens Ihres Codes und erleichtern es dem Entwicklungsteam, Ihren Code zu verstehen und mit ihm zu arbeiten.
- Durch die Automatisierung des Testprozesses erleichtert JUnit die Überprüfung, ob Änderungen an Ihrem Code neue Fehler oder Probleme mit sich gebracht haben, und erhöht das Vertrauen in Ihre Codeänderungen.
- JUnit-Tests lassen sich schnell ausführen, sodass Sie Fehler und Probleme früh im Entwicklungsprozess erkennen und den Zeit- und Arbeitsaufwand für deren Behebung reduzieren können.

JUnit kann in SAP-BTP-Projekte integriert werden. Mithilfe von JUnit können Sie automatisierte Tests für Ihre SAP-BTP-Anwendungen schreiben und ausführen, um deren Funktionalität zu validieren und etwaige Fehler oder Probleme frühzeitig im Entwicklungsprozess zu erkennen. SAP Cloud SDK bietet eine effektive Möglichkeit, JUnit in jedes Java-Projekt in SAP BTP zu integrieren. Mehr dazu erfahren Sie in Kapitel 4, »DevOps mit SAP Business Technology Platform«.

2.3.3 SonarQube

SonarQube ist eine Open-Source-Plattform für kontinuierliches Code-Review und Codequalitätsmanagement. Sie wurde entwickelt, um automatische Überprüfungen durch statische Analyse von Code durchzuführen, um Fehler zu identifizieren. Mit Hilfe von SonarQube können Sie auf detaillierte Berichte zu einer Reihe von Codequalitätsmetriken zugreifen, z. B. zu doppeltem Code, zur Einhaltung von Codierungsstandards, zur Abdeckung von Unit-Tests, zur Codekomplexität, zu Kommentaren, Fehlern und Sicherheitsempfehlungen.

SonarQube bietet automatisierte Tools zur Analyse der Qualität des Codes, einschließlich Metriken für Codekomplexität, Wartbarkeit, Sicherheit und Testabdeckung. In einer DevOps-Umgebung kann SonarQube in die CI/CD-Pipeline integriert werden, um Echtzeit-Feedback zur Codequalität zu liefern und potenzielle Probleme zu erkennen, bevor sie zu realen Problemen werden. Dies trägt dazu bei, die Gesamtqualität des Codes zu verbessern, die Produktivität der Entwickler*innen zu steigern und den Zeit- und Arbeitsaufwand für die Behebung von Fehlern und Problemen zu reduzieren.

SonarQube lässt sich mit einer Vielzahl von Programmiersprachen integrieren, darunter Java, C#, JavaScript und Python, und unterstützt eine Vielzahl von Entwicklungstools wie IDEs, Codeabdeckungstools und Code-Review-Tools. Es bietet eine webbasierte Schnittstelle, die es einfach macht, Daten zur Codequalität anzuzeigen und zu analysieren, den Fortschritt im Zeitverlauf zu verfolgen und mit anderen im Entwicklungsteam zusammenzuarbeiten.

SonarQube wurde ursprünglich im Jahr 2007 als Sonar veröffentlicht. Es wurde von SonarSource entwickelt, einem Unternehmen, das von Simon Brandhof und Freddy Mallet mit dem Ziel gegründet wurde, Entwickler*innen zu helfen, besseren Code zu schreiben und ihre Softwareentwicklungsprozesse zu optimieren.

Die ersten Versionen von SonarQube konzentrierten sich in erster Linie auf die Analyse von Java-Code und lieferten Berichte zu Codequalitätsmetriken wie Codeabdeckung, Codekomplexität und Einhaltung von Codierungsstandards. Im Laufe der Zeit wurde die Unterstützung für weitere Programmiersprachen hinzugefügt, darunter C/C++, C#, Python, Ruby und JavaScript.

Mit der Veröffentlichung von SonarQube 4.0 im Jahr 2013 wurde die Plattform grundlegend überarbeitet und um neue Funktionen erweitert, wie z. B. die Möglichkeit, Code auf Sicherheitsschwachstellen zu untersuchen, technische Schulden zu messen und die Integration mit einer Vielzahl anderer Entwicklungstools zu unterstützen.

Im Jahr 2016 wurde die Anwendung von Sonar in SonarQube umbenannt, und die Software wurde mit dem Fokus auf Skalierbarkeit und Leistung umgestaltet. Seitdem wurde SonarQube kontinuierlich weiterentwickelt und um neue Funktionen er-

gänzt, wie die auf maschinellem Lernen basierende Codeanalyse, Unterstützung für das Codequalitätsmanagement in DevOps-Pipelines und die Integration mit Cloud-basierten Code-Repositories. Heute wird SonarQube von Entwicklungsteams auf der ganzen Welt eingesetzt, um das Codequalitätsmanagement zu automatisieren und sicherzustellen, dass ihr Code den höchsten Qualitäts- und Sicherheitsstandards entspricht.

SonarQube analysiert den Quellcode einer Anwendung und identifiziert potenzielle Probleme und verbesserungswürdige Bereiche auf der Grundlage einer Reihe von vordefinierten Regeln und Richtlinien. Der Prozess kann in vier Schritte unterteilt werden (siehe Abbildung 2.8).

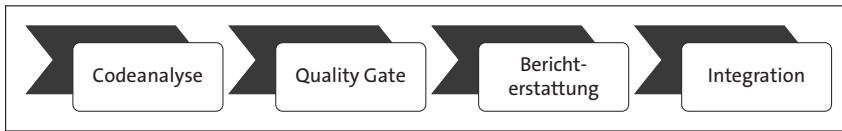


Abbildung 2.8 SonarQube-Aufrufablauf

Schauen wir uns diese Schritte im Detail an:

1. Codeanalyse

SonarQube verwendet eine Reihe von Techniken zur Analyse des Quellcodes einer Anwendung, einschließlich statischer Codeanalyse, Metrikenanalyse und Abhängigkeitsanalyse. Es untersucht den Code auf Probleme wie Sicherheitsschwachstellen, Code-Smells und technische Schulden.

2. Quality Gate

Ein *Quality Gate* ist ein Satz von Regeln und Schwellenwerten, die die Mindestqualitätsstandards für die Anwendung festlegen. Die Ergebnisse der Codeanalyse werden anhand des Quality Gates bewertet, um festzustellen, ob der Code die definierten Standards erfüllt.

3. Berichterstattung

Nach Abschluss der Codeanalyse erstellt SonarQube Berichte, die detaillierte Informationen über die Qualität des Codes liefern. Diese Berichte enthalten Metriken wie Codeabdeckung, Codeduplikation und die Anzahl der identifizierten Probleme. Die Berichte können verwendet werden, um verbesserungswürdige Bereiche zu identifizieren und den Fortschritt im Laufe der Zeit zu verfolgen.

4. Integration

SonarQube kann in verschiedenen Phasen in den Entwicklungsprozess integriert werden, auch während CI/CD. Die Integration kann mit einer Reihe von Entwicklungstools erfolgen, darunter IDEs, Build-Systeme und Quellcode-Verwaltungssysteme.

SonarQube bietet mehrere Vorteile für Softwareentwicklungsteams:

- **Verbesserte Codequalität**

SonarQube führt eine statische Codeanalyse durch, um Bugs, Code-Smells und Sicherheitsschwachstellen zu erkennen. So können Sie Probleme erkennen und beheben, bevor sie zu größeren Problemen werden.

- **Codeabdeckung**

SonarQube kann bei der Messung der Codeabdeckung helfen, indem es anzeigt, welche Teile des Codes nicht getestet werden, und es Ihnen ermöglicht, die Testabdeckung zu verbessern.

- **Integration in DevOps-Tools**

SonarQube kann in eine breite Palette von DevOps-Tools integriert werden, sodass die Codeanalyse automatisiert und in den Entwicklungsprozess integriert werden kann.

- **Konsistente Codequalität**

Durch die Durchsetzung von Codierungsstandards und die Identifizierung von Problemen in Echtzeit trägt SonarQube dazu bei, dass der von einem Team produzierte Code konsistent und wartbar ist.

- **Feedback in Echtzeit**

SonarQube bietet Echtzeit-Feedback zu Codequalitätsproblemen, sodass Sie Probleme sofort angehen können und die Zeit und Kosten für die Behebung von Problemen zu einem späteren Zeitpunkt im Entwicklungsprozess reduziert werden.

- **Geringere technische Schuld**

Durch die frühzeitige Identifizierung und Behebung von Codequalitätsproblemen im Entwicklungsprozess kann SonarQube dazu beitragen, die technischen Schulden zu reduzieren, sodass die Codebasis im Laufe der Zeit leichter gewartet und erweitert werden kann.

SonarQube wird von ABAP-Entwickler*innen intensiv genutzt, um die Qualität ihres Codes zu verbessern. Um SonarQube für die ABAP-Entwicklung zu verwenden, werden in der Regel die folgenden Schritte durchgeführt:

1. Installieren und konfigurieren Sie das ABAP-Plug-in für SonarQube. Das Plug-in ist auf dem SonarQube-Marktplatz erhältlich und kann wie jedes andere Plug-in installiert werden.
2. Sobald das ABAP-Plug-in installiert und konfiguriert ist, kann der Code mithilfe des Plug-ins analysiert werden. Die Analyse kann auf verschiedene Arten erfolgen, z. B. über das ABAP Test Cockpit, über die Eclipse-basierte ABAP-Entwicklungsumgebung (ABAP in Eclipse) oder über einen eigenständigen ABAP-Code-Analyzer.

3. Nach Abschluss der Codeanalyse können die Ergebnisse in der SonarQube-Web-oberfläche überprüft werden. Die Ergebnisse liefern detaillierte Informationen über Codequalitätsprobleme wie Bugs, Code-Smells und Sicherheitsschwachstellen.
4. Auf der Grundlage der Ergebnisse der Analyse können Sie Korrekturmaßnahmen ergreifen, um die Qualität des Codes zu verbessern. Dies kann die Behebung von Fehlern, das Refactoring des Codes zur Beseitigung von Code-Smells und die Behebung von Sicherheitslücken umfassen.
5. Mit SonarQube lassen sich die Fortschritte im Laufe der Zeit verfolgen, wobei Berichte und Dashboards Trends aufzeigen und Bereiche hervorheben, in denen Verbesserungen erzielt wurden oder weitere Arbeit erforderlich ist.

In SAP BTP können Sie SonarQube in Verbindung mit dem Projekt »Piper« verwenden, um die Codeanalyse als Teil der Pipeline durchzuführen. Das Projekt »Piper« ist ein Rahmenwerk für die Erstellung von Pipelines in SAP BTP, die zur Automatisierung der Bereitstellung von Anwendungen verwendet werden können. Mehr über das Projekt »Piper« erfahren Sie in Kapitel 4, »DevOps mit SAP Business Technology Platform«.

2.3.4 Jenkins

Im gleichnamigen Abschnitt 2.2.3 haben Sie Jenkins und seine Verwendung als Build-Tool kennengelernt. Jenkins ist auch ein hervorragendes Tool für die Testautomatisierung und bietet eine Reihe von Plug-ins, die in die Testpipeline integriert werden können, um eine nahtlose Integration mit mehreren Testautomatisierungstools und Frameworks zu ermöglichen. Im Hinblick auf die Testautomatisierung bietet Jenkins Plug-ins, die die Ausführung von Testsuiten, die Sammlung von Dashboard-Ergebnissen und die Bereitstellung von Details zu Testfehlern ermöglichen.

Schauen wir uns an, wie Jenkins die Testautomatisierung unterstützt:

- **Ausführung von automatisierten Testsuiten**

Jenkins bietet Plug-ins für verschiedene Test-Frameworks, darunter Selenium, Cucumber und Appium. Diese können nahtlos in CI-Pipelines integriert werden, um automatisierte Tests für jeden Build auszuführen.

- **Zusammenfassung der Ergebnisse**

Darüber hinaus fassen die meisten Plug-ins die Testergebnisse zusammen und präsentieren sie in Form einer HTML-Seite.

- **Bereitstellung von Trends**

Jenkins zeichnet die Ergebnisse auf und stellt sie in Form eines Trenddiagramms dar. Dies ermöglicht einen umfassenderen Überblick über die Leistung der Tests in der Vergangenheit.

■ Bereitstellung von Details zu Testfehlern

Die Testergebnisse werden tabellarisch erfasst, und Fehler werden zusammen mit den Testergebnissen protokolliert.

Jenkins ist zwar ein effektives Tool für die Implementierung der Testautomatisierung in CI/CD-Pipelines, hat jedoch einige Einschränkungen:

- Da es sich um ein gemeinschaftlich entwickeltes Tool handelt, an dem viele Menschen mitarbeiten, können die verschiedenen Plug-ins für eine einzige Tool-Integration dazu führen, dass die verfügbaren Optionen unübersichtlich werden. Außerdem erfordert die Konfiguration eine steile Lernkurve mit zu vielen Optionen, was es für neue User kompliziert macht.
- Jenkins hat auch eine begrenzte Flexibilität bei der Anpassung von Plug-ins, es fehlen Authentifizierungs- und Autorisierungsregeln und -optionen, und den meisten von der Community entwickelten Plug-ins fehlt eine klare Dokumentation.
- Obwohl Plug-ins zur Unterstützung der Docker-Integration verfügbar sind, ist Jenkins in diesem Bereich noch ausbaufähig, und User müssen sich auf Cloud-Service-Anbieter verlassen, um Jenkins zu hosten.

Trotz dieser Probleme machen die zahlreichen Plug-ins und die starke Unterstützung der Community Jenkins zu einem leistungsstarken Tool für die Testautomatisierung in CI/CD-Pipelines.

2.4 Bereitstellungstools

Die automatisierte Bereitstellung ist für Unternehmen von entscheidender Bedeutung, um das Risiko der Produktionsbereitstellung zu verringern und ein schnelleres Feedback über die Qualität ihrer Software zu erhalten. Mit der Bereitstellungsautomatisierung lässt sich Software auf Knopfdruck in Test- und Produktionsumgebungen bereitstellen.

DevOps-Bereitstellungstools werden zur Automatisierung und Verwaltung der Bereitstellung von Softwareanwendungen und Updates in einer DevOps-Umgebung verwendet. DevOps-Bereitstellungstools bieten in der Regel eine zentralisierte Plattform für die Automatisierung und Verwaltung der verschiedenen Phasen des Softwarebereitstellungsprozesses, von dem Build und Testen bis hin zur Bereitstellung und Überwachung von Anwendungen. DevOps-Tools sind darauf ausgelegt, den Bereitstellungsprozess zu rationalisieren, und bieten mehrere Vorteile, darunter eine verstärkte Entwicklung, eine verbesserte betriebliche Effizienz, eine schnellere Freigabe, eine ununterbrochene Bereitstellung, eine schnellere Innovation, eine verbesserte Zusammenarbeit und einen nahtlosen Ablauf in der Prozesskette. Dies führt zu

einer schnelleren Bereitstellung neuer Funktionen und Updates, zu zuverlässigerer und stabilerer Software sowie zu einer besseren Kommunikation und Zusammenarbeit zwischen den Teams. Insgesamt kann der Einsatz von DevOps-Tools die Effektivität und Effizienz des Softwarebereitstellungsprozesses erheblich verbessern.

Mit DevOps-Tools können sich die Entwicklungsteams auf den Build hochwertiger Anwendungen konzentrieren, während sich die Betriebsteams auf deren Bereitstellung und effizientere Verwaltung konzentrieren können.

Wie typische automatisierte Bereitstellungen funktionieren

Ein automatisierter Bereitstellungsprozess erfordert vom CI-Prozess erstellte Pakete, die in jeder Umgebung, einschließlich der Produktionsumgebung, bereitgestellt werden können sollten. Außerdem werden Skripte zur Konfiguration der Umgebung, zur Bereitstellung der Pakete und zur Durchführung eines Bereitstellungs- oder Rauchtests sowie umgebungsspezifische Konfigurationsinformationen benötigt. Durch die Automatisierung des Bereitstellungsprozesses können Teams eine schnellere und zuverlässigere Bereitstellung von Software gewährleisten, das Fehlerrisiko verringern und eine schnellere Identifizierung und Lösung von Problemen ermöglichen.

Um Software bereitzustellen, müssen mehrere Aufgaben durchgeführt werden, z. B. die Vorbereitung der Zielumgebung durch Installation und Konfiguration der erforderlichen Software oder die Einrichtung einer Produktionsumgebung. Als Nächstes sollten die Pakete bereitgestellt werden, gefolgt von allen mit der Bereitstellung zusammenhängenden Aufgaben wie der Ausführung von Skripten zur Datenbankmigration und der Durchführung der erforderlichen Konfigurationen. Schließlich sollte ein Bereitstellungstest durchgeführt werden, um sicherzustellen, dass das System ordnungsgemäß funktioniert und externe Dienste erreichbar sind. Wenn diese Schritte befolgt werden, kann die Softwarebereitstellung effizient und effektiv durchgeführt werden, während gleichzeitig sichergestellt wird, dass alle potenziellen Probleme erkannt und umgehend behoben werden.

Einige beliebte DevOps-Bereitstellungstools sind unter anderem Docker, Kubernetes, Jenkins, GitLab und Chef. In den folgenden Abschnitten sehen wir uns diese Tools genauer an.

2.4.1 Docker

Docker ist ein sehr interessantes und leistungsfähiges Konzept, das die gesamte Art und Weise, wie Anwendungen entwickelt und bereitgestellt werden, verändert hat. Bevor wir uns mit der Funktionsweise von Docker und seiner Verwendung in Verbindung mit SAP-Lösungen befassen, ist es wichtig, etwas über Docker selbst und die damit verbundene Terminologie zu erfahren.



Was ist Docker?

Docker ist eine beliebte Containerisierungsplattform, die es Ihnen ermöglicht, eine Anwendung und alle ihre Abhängigkeiten in einen einzigen, portablen Container zu packen, der auf jedem System ausgeführt werden kann (siehe Abbildung 2.9).

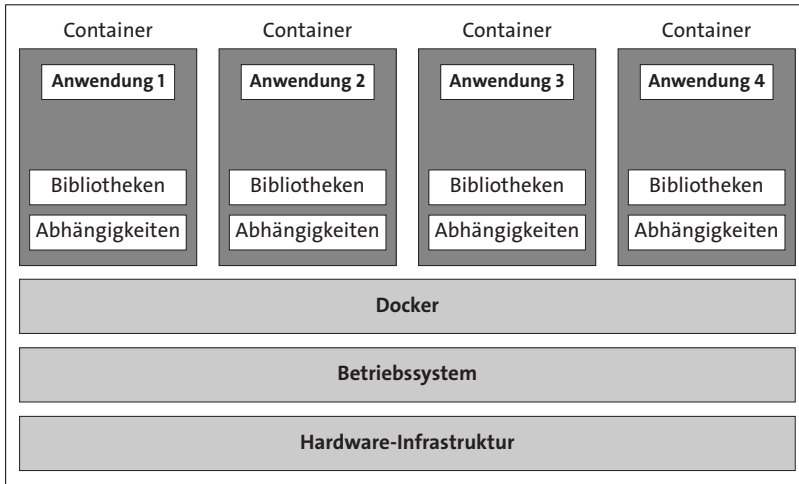


Abbildung 2.9 Docker

Docker-Container sind leichtgewichtig, effizient und können einfach von einer Umgebung in eine andere verschoben werden, was sie zu einer idealen Lösung für die Bereitstellung und Skalierung von Anwendungen in einer Vielzahl von Umgebungen macht, darunter On-Premise-, Cloud- und Hybrid-Umgebungen. Docker bietet auch eine Reihe von Tools und Services für die Verwaltung von containerisierten Anwendungen, einschließlich Containerorchestrierung, Überwachung und Sicherheitsfunktionen.

Mit einfachen Worten: Docker ist ein Tool, mit dem Container erstellt und ausgeführt werden können. Aber was ist Containerisierung, und was sind Container?

Containerisierung ist der Prozess, bei dem eine Anwendung zusammen mit allen zugehörigen Abhängigkeiten, Konfigurationsdateien und Bibliotheken so verpackt wird, dass ein reibungsloser und fehlerfreier Betrieb in verschiedenen Computerumgebungen gewährleistet ist. Die am häufigsten verwendeten Containerisierungsplattformen sind Kubernetes und Docker.

Die Herausforderung, eine zuverlässige Softwareleistung zu gewährleisten, wenn sie von einer Computerumgebung in eine andere verlagert wird, kann durch die Verwendung von Containern gelöst werden. Dies kann der Fall sein, wenn Software von einem Laptop in eine Testumgebung, von einer Staging-Umgebung in die Produktion oder sogar von einer physischen Maschine in einem Rechenzentrum in eine virtuelle Maschine in einer öffentlichen oder privaten Cloud verschoben wird.

Probleme treten häufig auf, wenn die Softwareumgebung am neuen Standort nicht identisch ist, z. B. wenn Tests unter Debian und der Produktionsbetrieb unter Red Hat laufen, was zu unerwarteten Problemen führen kann. Ein weiteres Beispiel ist das Testen mit Python 2.7 und die anschließende Umstellung auf Python 3 in der Produktion, was ebenfalls zu unvorhersehbarem Verhalten führen kann.

Was genau ist also ein Container? Ein *Container* umfasst eine vollständige Laufzeitumgebung, die eine Anwendung und alle erforderlichen Abhängigkeiten, Bibliotheken, Binärdateien und Konfigurationsdateien enthält. Diese Komponenten werden in einem einzigen Paket gebündelt.

Schauen wir uns ein Beispiel an, um das besser zu verstehen. Stellen Sie sich vor, ein Unternehmen plant die Entwicklung einer Java-Anwendung. Zu diesem Zweck richtet das Entwicklungsteam eine Umgebung ein, in der ein Tomcat-Server installiert ist. Nach der Entwicklung der Anwendung muss diese von einem Testteam getestet werden. Das Testteam muss jedoch eine Tomcat-Umgebung von Grund auf neu einrichten, um die Anwendung zu testen. Sobald die Tests abgeschlossen sind, wird die Anwendung auf dem Produktionsserver bereitgestellt, auf dem ebenfalls eine Tomcat-Umgebung für das Hosting der Java-Anwendung installiert werden muss. Die Tomcat-Umgebung wird also in diesem gesamten Prozess dreimal eingerichtet.

Dieselbe Tomcat-Umgebung wiederholt einzurichten, ist eine Verschwendung von Zeit und Mühe. Es kann außerdem zu Versionsinkonsistenzen zwischen verschiedenen Konfigurationen kommen, z. B. wenn die Entwicklungs- und Testteams Tomcat 7 verwenden, der Produktionsserver aber mit Tomcat 9 eingerichtet ist.

Um diesen Zeit- und Arbeitsaufwand zu vermeiden, können Docker-Container eingesetzt werden. Für die Erstellung einer Tomcat-Umgebung kann ein Tomcat-Docker-Image generiert werden, das eine Blaupause für die Bereitstellung mehrerer Container mit derselben Konfiguration ist.

Sie können z. B. ein Basis-Image wie Ubuntu verwenden, das auf Docker Hub leicht verfügbar ist, um das Image zu erstellen. Dasselbe Image kann dann von der testenden Person und der Systemadministration für die Bereitstellung der Tomcat-Umgebung verwendet werden. Docker-Container lösen das Problem, indem sie die Notwendigkeit beseitigen, wiederholt identische Umgebungen für jede Entwicklungsphase einzurichten.

Damit kommen wir zum Thema Docker-Images. Manchmal werden die Docker selbst mit dem Docker-Image verwechselt. Obwohl sie miteinander verbunden sind, gibt es einen feinen Unterschied. Docker-Images dienen als Bausteine für Container, mit denen Sie schnell und einfach Container erstellen können, auf denen bereits Software vorinstalliert ist. Sie sind in hohem Maße portabel, und Sie können entweder vorhandene Images verwenden oder Ihre eigenen von Grund auf neu erstellen.

Im Wesentlichen kann ein Docker-Image als eine Vorlage oder Blaupause angesehen werden, während ein Container eine tatsächlich ausgeführte Instanz dieses Image ist. Docker-Images sind in sich geschlossene Pakete, die alle für die Ausführung einer Anwendung erforderlichen Komponenten enthalten, wie z. B. Code, Laufzeit, Systemtools, Bibliotheken und Konfigurationen. Sie sind so konzipiert, dass sie leichtgewichtig, eigenständig und hocheffizient sind.

Wie funktioniert Docker?

Docker arbeitet mit einer Client-Server-Architektur, bei der der Docker-Daemon die Erstellung und den Betrieb von Containern übernimmt, während der Docker-Client mit dem Daemon kommuniziert. Der Docker-Client ist dafür verantwortlich, dass die User mit Docker interagieren und Befehle an den Docker-Daemon senden können. Ein Docker-Host und der Docker-Hub sind weitere Bausteine des Docker-Ökosystems. Schauen wir uns die Elemente dieses Ökosystems genauer an:

■ Docker-Daemon

Der Docker-Daemon (`dockerd`) ist für die Verwaltung von Docker-Objekten wie Images, Container, Netzwerke und Volumes zuständig, indem er auf Docker-API-Anforderungen wartet.

■ Docker-Client

Docker-User interagieren mit Docker hauptsächlich über den Docker-Client (`docker`). Der Client sendet Befehle, wie z. B. `docker run`, an den Docker-Daemon (`dockerd`), der sie dann ausführt. Mit seinem Command Line Interface (CLI) ermöglicht der Docker-Client den Usern, Befehle zum Erstellen, Ausführen und Beenden von Docker-Anwendungen zu erteilen. Zu den gängigen Befehlen, die der Client verwendet, gehören `docker build`, `docker pull` und `docker run`.

Der Docker-Client kann sich auf demselben Rechner wie der Docker-Daemon befinden oder alternativ eine Verbindung zu einem Daemon herstellen, der sich auf einem anderen Rechner befindet. Die Kommunikation zwischen dem Docker-Client und dem Daemon erfolgt über eine REST-API, die entweder über UNIX-Sockets oder eine Netzwerkschnittstelle übertragen werden kann.

■ Docker-Host

Ein Docker-Host ist entweder ein physisches Computersystem oder eine virtuelle Maschine, auf der Linux läuft und auf der Container bereitgestellt werden können. Dabei kann es sich um eine beliebige Computerressource handeln, z. B. einen Laptop, einen Server in einem Rechenzentrum oder eine in der Cloud bereitgestellte Ressource. Damit ein Computer als Docker-Host fungieren kann, muss er jedoch unter Linux laufen. Wenn auf dem Computer ein anderes Betriebssystem läuft, z. B. OS X oder Windows, muss eine virtuelle Maschine verwendet werden, auf der Linux läuft.

■ Docker-Hub/Docker-Registry

Docker-Hub, auch bekannt als Docker-Registry, dient als Repository für Docker-Images. Standardmäßig sucht Docker nach Images auf Docker-Hub, einer öffentlichen Registry, die von jedermann genutzt werden kann. Docker-Hub enthält eine umfangreiche Sammlung von Images, aus der eine Vielzahl von Optionen ausgewählt werden kann. Darüber hinaus ist es möglich, eine private Registry für den persönlichen oder organisatorischen Gebrauch einzurichten.

Abbildung 2.10 zeigt, wie diese Komponenten zusammenwirken.

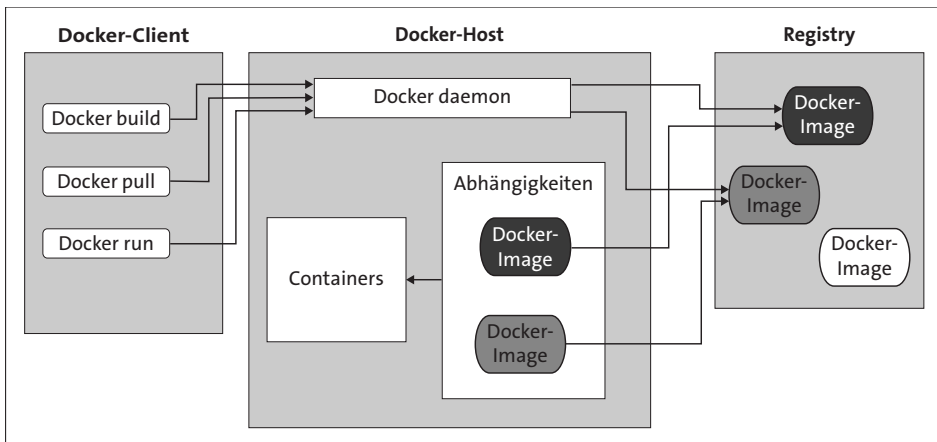


Abbildung 2.10 Funktion von Docker

Es gibt mehrere Vorteile von Docker, unter anderem die folgenden:

■ Konsistenz

Docker bietet eine konsistente Umgebung, in der Ihre Anwendung unabhängig von der zugrunde liegenden Infrastruktur ausgeführt werden kann. Dies bedeutet, dass Sie darauf vertrauen können, dass sich Ihre Anwendung in Entwicklungs-, Test- und Produktionsumgebungen gleich verhält.

■ Portabilität

Mit Docker können Sie Ihre Anwendung und alle ihre Abhängigkeiten in einen einzigen Container packen. Dieser Container kann dann auf jedem System ausgeführt werden, das Docker unterstützt, sodass Sie Ihre Anwendung problemlos zwischen verschiedenen Umgebungen verschieben können.

■ Effizienz

Docker-Container sind leichtgewichtig und können schnell erstellt und wieder gelöscht werden. Dies erleichtert die bedarfsgerechte Skalierung Ihrer Anwendung und kann zu Kosteneinsparungen in Bezug auf die Nutzung der Infrastruktur führen.

■ Isolierung

Docker bietet ein hohes Maß an Isolierung zwischen Containern, wodurch Konflikte zwischen Anwendungen vermieden und das Risiko von Sicherheitslücken verringert werden kann.

Verwendung von Docker mit SAP BTP

Docker wird in SAP BTP gut unterstützt. Tatsächlich ist die SAP BTP, Cloud-Foundry-Umgebung darauf ausgelegt, containerisierte Anwendungen zu unterstützen, einschließlich solcher, die auf Docker basieren. SAP BTP bietet eine Container-Registry, in der Sie Docker-Images speichern und verwalten können, und Sie können diese Images verwenden, um Anwendungen in der SAP BTP, Cloud-Foundry-Umgebung bereitzustellen.

Um eine Docker-basierte Anwendung in die SAP BTP, Cloud-Foundry-Umgebung zu pushen, können Sie die folgenden allgemeinen Schritte ausführen:

1. Zunächst müssen Sie ein Docker-Image erstellen. Zu diesem Zweck können Sie ein Docker-File verwenden, um die Anwendung und ihre Abhängigkeiten zu definieren. Verwenden Sie dann den Befehl `docker build`, um ein Docker-Image zu erstellen, z. B.:

```
docker build -t app-name
```

2. Dann müssen Sie das Docker-Image taggen. Sie können den Befehl `docker tag` verwenden, um das Image mit dem Repository-Namen und der Version zu taggen, z. B.:

```
docker tag my-image:latest my-registry.example.com/my-app:v1
```

3. Nachdem das Image erstellt und mit Tags versehen wurde, können Sie das Docker-Image in die Container-Registry übertragen. Verwenden Sie dazu den Befehl `docker push`:

```
docker push my-registry.example.com/my-app:v1
```

4. Als Nächstes müssen Sie eine Manifestdatei erstellen, um den Namen der Anwendung, das zu verwendende Docker-Image und alle erforderlichen Umgebungsvariablen zu definieren. Ein Beispiel finden Sie in Listing 2.1.

5. Jetzt können Sie die Anwendung mit dem Befehl `cf push` auf Cloud Foundry pushen – z. B.:

```
cf push -f manifest.yml
```

6. Dadurch wird eine Anwendung in der Cloud-Foundry-Umgebung unter Verwendung des angegebenen Docker-Image und der Umgebungsvariablen erstellt. Sie können dann auf die Anwendung unter der ihr zugewiesenen URL zugreifen.

```
---
applications:
- name: my-app
  docker:
    image: my-registry.example.com/my-app:v1
  env:
    MY_ENV_VAR: my-value
```

Listing 2.1 Beispieldatei »manifest.yml«

Darüber hinaus unterstützt SAP BTP auch Kubernetes, eine weitere Containerisierungstechnologie, die häufig zusammen mit Docker verwendet wird und auf die wir im nächsten Abschnitt eingehen werden. SAP BTP, Kyma-Laufzeit unterstützt Docker als eine Containerisierungstechnologie. Kyma basiert auf Kubernetes, einer Plattform zur Orchestrierung von Containern, und Docker ist eine beliebte Wahl zur Erstellung von Container-Images, die auf Kubernetes ausgeführt werden können.

In Kyma können Sie Docker verwenden, um Ihre Anwendung und deren Abhängigkeiten in ein Container-Image zu packen, das dann von Kubernetes bereitgestellt und verwaltet werden kann. Kyma bietet eine Container-Registry, in der Sie Ihre Docker-Images speichern und verwalten können, und Sie können diese Images verwenden, um Ihre Anwendungen als Kubernetes-Implementierungen, Pods oder andere Ressourcen einzusetzen.

2.4.2 Kubernetes

Kubernetes ist ein System zur Container-Orchestrierung. Bevor wir Kubernetes verstehen, sollten wir das Problem verstehen, das zur Entstehung von Kubernetes geführt hat.

Container sind ein effektives Mittel, um Ihre Anwendungen zu verpacken und auszuführen. Wenn es jedoch um Produktionsumgebungen geht, müssen die Container, in denen Ihre Anwendungen gehostet werden, effektiv verwaltet und ihre ständige Verfügbarkeit sichergestellt werden. Im Falle eines Containerausfalls sollte beispielsweise nahtlos ein alternativer Container gestartet werden, um Ausfallzeiten zu vermeiden.

Das Problem, das zur Entstehung von Kubernetes führte, war der Bedarf an einer skalierbaren und zuverlässigen Methode zur Verwaltung von containerisierten Anwendungen in Produktionsumgebungen. Vor Kubernetes war die Verwaltung von containerisierten Anwendungen ein manueller und fehleranfälliger Prozess, der viel manuelle Konfiguration und Verwaltung erforderte. Mit der zunehmenden Verbreitung von Containern wurde klar, dass eine automatisierte und effizientere Methode zu ihrer Verwaltung benötigt wurde.

Kubernetes entstand aus der Erfahrung mit der Verwaltung von Containern in großem Maßstab bei Google, wo es intern zur Verwaltung von Produktions-Workloads verwendet wurde. Die ursprüngliche Version von Kubernetes, Borg genannt, wurde von Google in den frühen 2000er-Jahren entwickelt und diente zur Verwaltung von Hunderttausenden von Containern auf Tausenden von Servern. Borg wurde schließlich als Open Source zur Verfügung gestellt und bildete die Grundlage für Kubernetes.

Kubernetes wurde entwickelt, um eine Plattform für die Bereitstellung, Skalierung und Verwaltung von containerisierten Anwendungen zu schaffen, die skalierbar, zuverlässig und automatisiert ist. Es bietet eine Reihe leistungsstarker und flexibler Tools für die Verwaltung von Containern, darunter Funktionen wie automatischer Lastausgleich, Service-Erkennung und Selbstheilung. Kubernetes bietet außerdem einen plattformunabhängigen Ansatz, was bedeutet, dass es zur Verwaltung von Containern bei verschiedenen Cloud-Anbietern und in lokalen Rechenzentren eingesetzt werden kann.

Kubernetes, auch bekannt als K8s, ist eine Open-Source-Container-Orchestrierungsplattform für die Verwaltung und Bereitstellung von containerisierten Anwendungen in großem Umfang. Es bietet eine Reihe leistungsstarker Funktionen für die Verwaltung von Containern:

- automatische Bereitstellung, Skalierung und Lastausgleich von Containern
- Selbstheilung und Fehlertoleranz, mit automatischem Austausch der Container im Falle eines Ausfalls
- Orchestrierung und Verwaltung von Containerspeichern
- automatische Erkennung und Weiterleitung von Services
- einfache Integration mit Container-Registrierungsstellen und anderen Cloud-Diensten

Kubernetes basiert auf einer *Master-Slave-Architektur*. Der Master fungiert als Steuerungsebene für den gesamten Cluster, während die Knoten oder die Arbeitsmaschinen die containerisierten Anwendungen ausführen. Hier finden Sie einen Überblick über die Funktionsweise von Kubernetes:

- Die User stellen eine Anwendung bereit, indem sie ihren gewünschten Zustand in einer *Kubernetes-Manifestdatei* definieren, in der die Container, ihre Konfigurationen sowie die gewünschte Skalierung und Redundanz der Anwendung angegeben sind.
- Der Kubernetes-API-Server empfängt das Manifest und speichert es in ETCD, einem verteilten Key-Value-Speicher, der als Statusdatenbank des Clusters dient.
- Der Kubernetes-Scheduler weist die Container bestimmten Knoten zu, basierend auf den verfügbaren Ressourcen und anderen Einschränkungen, wie z. B. Knotenaffinität oder Antiaffinität.

- Der Kubernetes-Controller-Manager überwacht kontinuierlich den Zustand des Clusters und nimmt die erforderlichen Anpassungen vor, um sicherzustellen, dass der aktuelle Zustand dem in der Manifestdatei angegebenen gewünschten Zustand entspricht. Wenn z. B. ein Container ausfällt, startet der Controller-Manager ihn automatisch neu.
- Die Kubernetes-Knoten führen die Container aus und melden ihren Status an den Kubernetes-API-Server zurück.
- Der Kubernetes-Service fungiert als Abstraktionsschicht, die die Container dem externen Netzwerk zugänglich macht, den Datenverkehr zwischen den Container-Replikaten ausgleicht und eine stabile IP-Adresse und einen DNS-Namen für den Zugriff auf die Anwendung bereitstellt.
- Schließlich bietet Kubernetes eine Vielzahl von Tools für die Überwachung, Protokollierung und Fehlerbehebung der Anwendung, einschließlich Dashboards, Befehlszeilentools und Integrationen von Drittanbietern.

Beim Vergleich von Kubernetes und Docker ist es wichtig zu verstehen, dass Docker eine unabhängige Software ist, die auf jedem Computer installiert werden kann, um Anwendungen in Containern auszuführen. Sie ermöglicht es, Container auf einem einzigen Betriebssystem zu erstellen, zu verwalten und auszuführen. Durch die Installation von Docker auf mehreren Hosts mit unterschiedlichen Betriebssystemen können Sie die Vorteile von Kubernetes nutzen. Mit Kubernetes können Sie den Prozess der Container-Bereitstellung, der Vernetzung, des Lastausgleichs, der Sicherheit und der Skalierung auf all diesen Knoten über eine einzige Befehlszeile oder ein Dashboard automatisieren.

Obwohl Kubernetes mit jeder Containerisierungstechnologie funktionieren kann, sind zwei häufig verwendete Optionen, die Kubernetes einbinden kann, rkt und Docker. Docker ist die am weitesten verbreitete Containerisierungstechnologie, und die Integration zwischen Docker und Kubernetes ist hochentwickelt und wird häufig angewendet.

In ähnlicher Weise bietet Docker, Inc. seine eigene Container-Orchestrierungs-Engine namens Docker Swarm an. Das Unternehmen hat jedoch die wachsende Beliebtheit von Kubernetes erkannt und seine eigene Kubernetes-Distribution in Docker Desktop (verfügbar für macOS und Windows) integriert.

Für die SAP-Welt ist die SAP BTP, Kyma-Laufzeitumgebung eine Kubernetes-basierte Plattform, die eine Reihe von vorgefertigten Microservices und Erweiterungen bietet, die sich leicht in Ihre Anwendungen integrieren lassen. Mit Kyma können Sie Ihren Kubernetes-Clustern ganz einfach Funktionen wie Protokollierung, Überwachung, Sicherheit und Authentifizierung hinzufügen.

2.4.3 Jenkins

In Abschnitt 2.2.3 und Abschnitt 2.3.4 haben Sie Jenkins und seine Verwendung als Build- und Testautomatisierungstool kennengelernt. Jenkins kann auch zur Automatisierung der Bereitstellung verwendet werden.

Es wird in der Regel als Automatisierungstool für die Bereitstellung verwendet, um die Bereitstellung von Anwendungen in verschiedenen Umgebungen zu automatisieren. Jenkins bietet als Automatisierungstool für die Bereitstellung mehrere Vorteile, darunter einen rationalisierten Bereitstellungsprozess, der den Zeit- und Arbeitsaufwand für die Bereitstellung und Konfiguration von Anwendungen reduziert. Automatisierte Tests sind ein weiterer Vorteil von Jenkins, da mit ihnen sichergestellt werden kann, dass Codeänderungen vor der Bereitstellung gründlich getestet werden. Darüber hinaus kann die Verwendung von Jenkins für die Bereitstellungsautomatisierung die Effizienz verbessern und die Zuverlässigkeit der Bereitstellung erhöhen, wodurch das Risiko von Fehlern oder Problemen in Produktionsumgebungen verringert wird. Schließlich lässt sich Jenkins problemlos in andere Tools wie Git, Docker und Kubernetes integrieren, was die Bereitstellung von Anwendungen über verschiedene Plattformen und Umgebungen hinweg vereinfacht.

Lassen Sie uns ein Beispiel betrachten. Angenommen, Sie möchten Jenkins verwenden, um die Bereitstellung von SAP BTP zu automatisieren. Eine Möglichkeit, dies zu erreichen, ist die folgende Vorgehensweise:

1. Installieren und konfigurieren Sie Jenkins auf Ihrem lokalen Rechner oder Server.
2. Installieren Sie das SAP BTP Command Line Interface und melden Sie sich über die CLI bei Ihrem SAP-BTP-Konto an.
3. Erstellen Sie ein Jenkins-Projekt für Ihre Anwendung und konfigurieren Sie die erforderlichen Build-Schritte, einschließlich der Erstellung der Anwendung und der Erstellung eines bereitstellbaren Artefakts.
4. Installieren Sie das Cloud-Foundry-Plug-in für Jenkins, mit dem Sie Anwendungen in SAP BTP bereitstellen können.
5. Konfigurieren Sie das Cloud-Foundry-Plug-in mit den erforderlichen SAP-BTP-Anmeldedaten und Endpunktinformationen.
6. Fügen Sie eine neue Post-Build-Aktion zu Ihrem Jenkins-Projekt hinzu, um die Anwendung mithilfe des Cloud-Foundry-Plug-ins in SAP BTP bereitzustellen.
7. Testen Sie Ihre Bereitstellung, indem Sie Ihr Jenkins-Projekt ausführen und überprüfen, ob Ihre Anwendung erfolgreich in SAP BTP bereitgestellt wurde.

Bitte beachten Sie, dass dies nur eine Möglichkeit ist, dies zu tun. Sie können Jenkins auch mit anderen SAP-BTP-Services oder -Tools integrieren.

2.4.4 GitLab

GitLab ist ein webbasierter Git-Repository-Manager, der Quellcodeverwaltung, CI/CD-Pipelines und verschiedene andere Funktionen im Zusammenhang mit DevOps bietet. Mit GitLab können Teams Code-Repositories verwalten und zusammenarbeiten, Probleme und Bugs verfolgen, Softwaretests und -bereitstellung automatisieren und vieles mehr.

Im Hinblick auf DevOps ist GitLab besonders nützlich, um CI/CD zu erleichtern. Es ermöglicht die Automatisierung der Build-, Test- und Bereitstellungsprozesse, was die Geschwindigkeit und Qualität der Softwareentwicklung erheblich verbessern kann.

Darüber hinaus verfügt GitLab über eine Reihe weiterer Funktionen, die für DevOps nützlich sind, wie Versionskontrolle, Code-Review, Fehlerverfolgung und Tools für die Zusammenarbeit. GitLab lässt sich auch mit einer Vielzahl anderer DevOps-Tools und -Plattformen integrieren, z. B. Kubernetes, Docker, AWS und Azure.

GitLab wurde 2011 von Dmitry Zaporozhets und Valery Sizov als Git-Repository-Manager entwickelt. Es wurde 2012 als Open-Source-Projekt veröffentlicht, und 2013 startete GitLab seinen ersten kostenpflichtigen Dienst, GitLab Enterprise Edition.

Im Jahr 2015 führte GitLab seine CI/CD-Pipeline ein, mit der Teams ihre Build-, Test- und Bereitstellungsprozesse automatisieren können. Diese Funktion wurde zu einem der wichtigsten Verkaufsargumente von GitLab.

Im Jahr 2017 erhielt GitLab eine Finanzierung in Höhe von 20 Millionen US-Dollar und veröffentlichte GitLab 10.0, mit dem eine Reihe neuer Funktionen eingeführt wurde, darunter die Integration von Kubernetes und die Unterstützung für die Container-Registrierung. Seitdem hat GitLab immer wieder neue Funktionen hinzugefügt, z. B. Sicherheitsscans, Code-Reviews und Incident-Management. Im Jahr 2020 wurde GitLab von Thoma Bravo übernommen, einem Private-Equity-Unternehmen, das weiterhin in die Entwicklung und das Wachstum des Unternehmens investiert.

GitLab bietet eine Vielzahl von Vorteilen für Software-Entwicklungsteams, darunter die folgenden:

- **Umfassende DevOps-Plattform**

GitLab bietet ein komplettes Toolset für den gesamten Lebenszyklus der Softwareentwicklung, einschließlich Quellcodeverwaltung, CI/CD, Sicherheitsscans, Code-Review und Incident-Management.

- **Schnellere Entwicklungszyklen**

Mit den CI/CD-Pipelines von GitLab können Sie die Build-, Test- und Bereitstellungsprozesse automatisieren, was die Entwicklungszyklen erheblich verkürzen und die Geschwindigkeit der Softwarebereitstellung erhöhen kann.

■ Verbesserte Zusammenarbeit

GitLab bietet eine Reihe von Tools für die Zusammenarbeit, wie z. B. Fehlerverfolgung, Zusammenführungsanfragen und Code-Review, die es Entwickler*innen leichter machen, zusammenzuarbeiten und zu einem Projekt beizutragen.

■ Verbesserte Sicherheit

GitLab enthält integrierte Sicherheitsscanner, mit denen Schwachstellen im Code erkannt und Sicherheitsverletzungen verhindert werden können.

■ On-Premise-Hosting-Optionen

GitLab kann On-premise bereitgestellt werden, was besonders für Unternehmen interessant sein kann, die Sicherheits- oder Compliance-Anforderungen haben, die den Einsatz von Cloud-basierten Diensten verbieten.

■ Support auf Unternehmensebene

GitLab bietet Support auf Unternehmensebene, was für größere Organisationen, die spezielle Support-Ressourcen und maßgeschneiderte Lösungen benötigen, von großem Nutzen sein kann.

2.4.5 Chef

Chef ist ein beliebtes Konfigurationsmanagement-Tool für DevOps, das IaC-Praktiken durch die Automatisierung der Konfiguration, Bereitstellung und Verwaltung von Servern und Anwendungen ermöglicht. Chef ermöglicht es Entwicklungsteams und der Systemadministration, ihre Infrastruktur und Anwendungen als Code zu definieren, indem sie eine domänenspezifische Sprache (engl. Domain-specific language, DSL) namens *Chef DSL* verwenden.

Mit Chef können Sie Kochbücher erstellen, die definieren, wie Ihre Infrastruktur und Anwendungen konfiguriert werden sollen, und diese Kochbücher dann verwenden, um die Bereitstellung Ihrer Software über mehrere Server oder Umgebungen zu automatisieren. Chef bietet auch eine zentrale Verwaltungskonsole, *Chef Automate* genannt, mit der User ihre Infrastruktur und Anwendungen überwachen und verwalten können.

Chef hilft Unternehmen dabei, Konsistenz und Zuverlässigkeit in ihren Infrastruktur- und Anwendungsimplementierungen zu erreichen, indem sichergestellt wird, dass alle Konfigurationen standardisiert und reproduzierbar sind. Dies trägt dazu bei, Fehler zu reduzieren und Ausfallzeiten zu minimieren, während gleichzeitig die Sicherheit und Compliance durch die Durchsetzung konsistenter Konfigurationen über alle Systeme hinweg verbessert wird.

Chef wurde 2008 von Adam Jacob, Barry Steinglass und Nathan Hane entwickelt und hat sich seitdem zu einem beliebten Tool für die Verwaltung von IaC entwickelt. Es wurde ursprünglich entwickelt, um das Problem der Verwaltung komplexer Infra-

strukturen in großen Webunternehmen zu lösen. Die Gründer von Chef, die zu dieser Zeit bei einem Webunternehmen namens HJK Solutions arbeiteten, waren frustriert über den zeitaufwendigen und fehleranfälligen Prozess der manuellen Konfiguration von Servern und beschlossen, ein Tool zu entwickeln, das diesen Prozess automatisieren konnte.

Die erste Version von Chef wurde 2009 veröffentlicht und gewann schnell an Popularität in der DevOps-Community. Die Chef-Infra-DSL von Chef definiert IaC und bietet eine breite Palette von Ressourcen, Modulen und Kochbüchern, mit denen User die Konfiguration ihrer Server und Anwendungen automatisieren können.

Im Jahr 2020 kündigte Chef an, dass es von einem traditionellen Softwarelizenzierungsmodell zu einem Open-Source-Modell übergehen und seine Kernprodukte und -funktionen kostenlos zur Verfügung stellen würde. Dieser Schritt hat dazu beigetragen, die Reichweite und Akzeptanz von Chef in der DevOps-Community weiter zu erhöhen.

2.5 Überwachungstools

Überwachungstools sind ein wesentlicher Bestandteil von DevOps und werden verwendet, um die Leistung und den Zustand einer Anwendung oder eines Systems zu überwachen. Die DevOps-Überwachung umfasst die Überwachung und Messung des Zustands und der Leistung von Systemen und Anwendungen, um Probleme frühzeitig zu erkennen und zu beheben. Es sammelt Daten zu verschiedenen Parametern wie CPU-Auslastung, Festplattenspeicher und Antwortzeiten von Anwendungen. Durch die frühzeitige Erkennung von Problemen hilft die DevOps-Überwachung, eine Verschlechterung der Services und Ausfälle zu verhindern.

Wenn ein Unternehmen eine DevOps-Kultur und einen DevOps-Ansatz verfolgt, ist die Überwachung von entscheidender Bedeutung für die Erkennung von Systemproblemen, bevor sie zu Fehlern führen. Eine effektive Überwachung hilft dabei, Bedenken hinsichtlich der Entwicklungseffizienz und der Systemkomplexität auszuräumen, z. B. um sicherzustellen, dass keine Codeschwachstellen eingeführt werden, während der Code schnell ausgeliefert wird, um mehrere Systemkomponenten im Auge zu behalten und um den notwendigen Einblick in ein Projekt zu erhalten, das wie eine undurchdringliche Blackbox erscheinen mag.

Zu den wichtigsten Verwendungszwecken von DevOps-Überwachungstools gehören die folgenden:

- **Probleme erkennen und schnell beheben**

DevOps-Überwachungstools helfen den Teams, Probleme im System schon frühzeitig zu erkennen und proaktiv zu beheben, bevor sie zu größeren Probleme heranwachsen.

■ System- und Anwendungsleistung verfolgen

DevOps-Überwachungstools liefern Echtzeitdaten zu wichtigen Leistungsmetriken wie CPU-Auslastung, Speichernutzung und Antwortzeiten, sodass Teams Leistungsprobleme schnell erkennen und beheben können.

■ Einblick in das System gewähren

DevOps-Überwachungstools bieten Einblick in das gesamte System und ermöglichen es den Teams, die Ursache von Problemen schnell zu erkennen und fundierte Entscheidungen zu treffen.

■ Kontinuierliche Verbesserungen ermöglichen

DevOps-Überwachungstools bieten Einblicke in die Leistung des Systems im Laufe der Zeit und ermöglichen es den Teams, verbesserungswürdige Bereiche zu identifizieren und datengestützte Entscheidungen zur Optimierung des Systems zu treffen.

■ Die Zusammenarbeit zwischen Teams erleichtern

DevOps-Überwachungstools bieten eine gemeinsame Sicht auf das System, sodass verschiedene Teams durch die Bereitstellung zuverlässiger und leistungsfähiger Anwendungen effektiver zusammenarbeiten und auf ein gemeinsames Ziel hinarbeiten können.

Zu den gängigen Überwachungstools für DevOps gehören Dynatrace, Grafana und Splunk. In den folgenden Abschnitten sehen wir uns jedes dieser Tools im Detail an.

2.5.1 Dynatrace

Dynatrace ist eine Software-Intelligence-Plattform, die Echtzeit- und automatisierte Überwachung von Cloud-nativen Anwendungen und der zugrunde liegenden Infrastruktur bietet. Sie bietet eine umfassende Reihe von Tools für die Beobachtung, einschließlich Überwachung der Anwendungsleistung (engl. Application Performance Monitoring, APM), Infrastrukturüberwachung, Überwachung des digitalen Erlebnisses und Geschäftsanalytik.

Die Plattform nutzt künstliche Intelligenz und maschinelles Lernen, um Leistungsprobleme automatisch zu erkennen und zu diagnostizieren, und liefert umsetzbare Erkenntnisse und Warnungen, um Teams bei der schnellen Lösung von Problemen und der Optimierung der Leistung zu unterstützen. Dynatrace wird häufig in DevOps und IT-Betrieb eingesetzt, um die Verfügbarkeit, Skalierbarkeit und Zuverlässigkeit komplexer, Cloud-nativer Systeme sicherzustellen.

Dynatrace kann als Überwachungstool in SAP BTP eingesetzt werden, um die optimale Leistung und Verfügbarkeit von Anwendungen und Services zu gewährleisten. Mit Dynatrace können Sie Ihren gesamten Anwendungstapel überwachen, ein-

schließlich Cloud-Plattformen, Betriebssysteme, Datenbanken und Middleware. Dynatrace bietet Echtzeiteinblicke in den Zustand und die Leistung Ihrer Anwendungen und Services und ermöglicht es Ihnen, Probleme schnell zu erkennen und zu beheben, bevor sie sich auf die End-User auswirken. Es liefert Ihnen außerdem die nötigen Informationen, um die Leistung Ihres SAP-Systems zu optimieren. Dazu gehören Funktionen wie Sichtbarkeit auf Codeebene, Datenbanküberwachung und Überwachung von Usern.

Darüber hinaus kann Dynatrace Ihnen dabei helfen, die Überwachung und Analyse Ihres SAP-Systems zu automatisieren, wodurch der Bedarf an manueller Überwachung reduziert wird und Sie sich auf die Behebung von Problemen und die Verbesserung der Leistung konzentrieren können. Dynatrace kann auch mit anderen Tools und Technologien integriert werden, die in SAP-Systemen verwendet werden, wie z. B. SAP Solution Manager und SAP HANA, was es zu einem leistungsstarken Tool für die Verwaltung und Optimierung von SAP-Systemen macht.

Schauen wir uns zwei spezifische Dynatrace-Erweiterungen an, die in SAP-Ökosystemen verwendet werden:

- **Dynatrace-ActiveGate-Erweiterungen**

Diese ermöglichen die Integration zwischen Dynatrace-Überwachung und Remote-Technologien, die eine Schnittstelle zur Verfügung stellen. Diese Integration ermöglicht eine umfassende Überwachung von ABAP-Systemen und verbessert die Kommunikation zwischen den IT-Abteilungen des Unternehmens und den SAP-Basis-Teams. Kürzlich gab Dynatrace bekannt, dass seine SAP-Überwachungsfunktionen für die Unterstützung von SAP-HANA-Datenbanken erweitert wurden.

- **Dynatrace-Überwachungserweiterung für ABAP**

Diese Erweiterung liefert schnelle und leicht verständliche Informationen darüber, wie gut Ihr SAP-System für alle User funktioniert. Dies wird durch die Berichterstattung über jede Transaktionscode-Ausführung von jedem SAP-User erreicht.

2.5.2 Grafana

Grafana ist eine Open-Source-Software, die eine umfassende Lösung für die Durchführung und Visualisierung von Datenanalysen, die Überwachung von Anwendungen und das Abrufen von Metriken über anpassbare Dashboards bietet.

Unabhängig davon, wo die Metriken gespeichert sind, ermöglicht Grafana den Usern, ihre Metriken abzufragen, zu erforschen, zu visualisieren und Alarme zu setzen. Darüber hinaus hilft es bei der Verfolgung des User- und Anwendungsverhaltens, bei der Identifizierung der Fehlerhäufigkeit in Produktions- und Vorproduktionsumgebungen und vielem mehr.

Die von Grafana angebotenen Tools ermöglichen die Umwandlung von Zeitreihendatenbanken (engl. Time Series Database, TSDB) in Visualisierungen und Diagramme, die die technische Analyse und Überwachung von Daten über einen bestimmten Zeitraum ermöglichen, was gemeinhin als *Zeitreihenanalyse* bezeichnet wird.

Einige der Hauptmerkmale von Grafana sind die folgenden:

- Grafana unterstützt die Integration mit verschiedenen Datenquellen wie Kubernetes-Clustern, Raspberry Pis, Google Sheets, Graphite, Prometheus, Elasticsearch, InfluxDB, MySQL und PostgreSQL und ermöglicht so die Verbindung und Abfrage von Daten aus verschiedenen Quellen.
- Grafana bietet eine benutzungsfreundliche Oberfläche zum Erstellen und Anpassen von Dashboards, die Visualisierungen wie Diagramme, Tabellen und Messgeräte enthalten können.
- Grafana verfügt über ein integriertes Warnsystem, mit dem Warnungen basierend auf ihren Datenabfragen und Schwellenwerten eingerichtet werden können. User können Benachrichtigungen per E-Mail, Slack oder über andere Kanäle erhalten.
- Grafana verfügt über eine Plug-in-Architektur, die es den Usern ermöglicht, die Funktionalität mit zusätzlichen Funktionen und Integrationen zu erweitern.
- Grafana bietet robuste Authentifizierungs- und Autorisierungsfunktionen, einschließlich Unterstützung für LDAP, OAuth und andere Authentifizierungsanbieter.
- Grafana unterstützt die Zusammenarbeit im Team, indem es mehreren Usern die Arbeit am selben Dashboard oder an derselben Datenquelle ermöglicht. User können ihre Dashboards auch mit anderen teilen und den Zugriff über eine rollenbasierte Zugriffskontrolle steuern.
- Grafana kann mit Tools wie Kubernetes problemlos in einer Cloud-nativen Umgebung eingesetzt werden und ist damit eine ideale Lösung für Unternehmen, die Container-Orchestrierungsplattformen verwenden.

Grafana kann in SAP BTP verwendet werden, um Daten aus verschiedenen Quellen zu visualisieren und zu analysieren. SAP BTP bietet eine Reihe von Services wie Datenbanken, IoT-Services und Funktionen für maschinelles Lernen, die unter anderem zur Generierung von Daten verwendet werden können. Mit Grafana können User eine Verbindung zu diesen Datenquellen herstellen und benutzungsdefinierte Dashboards erstellen, die bei der Überwachung und Visualisierung von Daten aus diesen Quellen helfen.

Eine Anwendung für Grafana in SAP BTP ist die Überwachung von IoT-Geräten. SAP BTP bietet einen IoT-Service, der die Sammlung von Daten von IoT-Geräten wie Sensoren oder Maschinen ermöglicht. Diese Daten können dann mit Grafana analysiert

und visualisiert werden, um Einblicke in die Leistung und das Verhalten dieser Geräte zu erhalten.

Ein weiteres Beispiel ist die Verwendung von Grafana zur Überwachung der Leistung von Anwendungen, die auf SAP BTP laufen. Mit Grafana können User eine Verbindung zu SAP BTP herstellen und benutzungsdefinierte Dashboards erstellen, die Leistungsmetriken wie Antwortzeit, Durchsatz und Fehlerraten anzeigen.

2.5.3 Splunk

Splunk ist eine Softwareplattform zum Durchsuchen, Analysieren und Visualisieren von maschinengenerierten Daten. Sie sammelt und indiziert Daten aus verschiedenen Quellen wie Anwendungen, Server, Netzwerke und Sicherheitsgeräte und ermöglicht es, Daten aus verschiedenen Quellen in Echtzeit zu suchen, zu analysieren und zu korrelieren. Splunk bietet eine leistungsstarke Suchsprache, mit der Sie Daten mithilfe einer Vielzahl von Methoden wie Feldsuche, reguläre Ausdrücke und bedingte Ausdrücke suchen und analysieren können.

Splunk ist hoch skalierbar und kann On-premise, in der Cloud oder in hybriden Umgebungen eingesetzt werden. Es verfügt über ein großes Ökosystem von Apps und Add-ons, die zusätzliche Funktionen und Integrationen mit Systemen von Drittanbietern bieten.

Einige der wichtigsten Funktionen von Splunk sind die folgenden:

- Splunk bietet eine zentrale Plattform zum Sammeln und Indizieren von Daten aus einer Vielzahl von Quellen wie z. B. Server, Anwendungen und Netzwerke. Es kann eine Vielzahl von Datentypen und -formaten verarbeiten, darunter Protokolldateien, Maschinendaten und strukturierte Daten.
- Splunk kann Daten in Echtzeit überwachen und Warnungen ausgeben, wenn bestimmte Bedingungen erfüllt sind. Warnungen können per E-Mail, SMS oder über andere Kanäle gesendet werden.
- Splunk bietet eine leistungsstarke Suchsprache, mit der Sie Daten mithilfe einer Vielzahl von Methoden wie Feldsuche, reguläre Ausdrücke und bedingte Ausdrücke suchen und analysieren können. Außerdem bietet es Funktionen für maschinelles Lernen, die für die Erkennung von Anomalien und prädiktive Analysen genutzt werden können.
- Splunk bietet eine Reihe von Visualisierungs- und Berichtstools, mit denen Sie individuelle Dashboards, Diagramme und Grafiken erstellen können. Es bietet eine benutzungsfreundliche Oberfläche zum Erstellen und Anpassen von Berichten, die in verschiedenen Formaten exportiert werden können.
- Splunk bietet Funktionen, die Unternehmen dabei helfen, Sicherheits- und Compliance-Anforderungen zu erfüllen, z. B. Bedrohungserkennung in Echtzeit und Reaktion auf Vorfälle, Auditing und Reporting sowie Compliance-Reporting.

- Splunk ist hoch skalierbar und kann On-premise, in der Cloud oder in hybriden Umgebungen bereitgestellt werden. Es verfügt über ein großes Ökosystem von Apps und Add-ons, die zusätzliche Funktionen und Integrationen mit Systemen von Drittanbietern bieten.

Splunk kann in SAP BTP zum Sammeln, Überwachen und Analysieren von maschinell erzeugten Daten aus verschiedenen Quellen verwendet werden. Eine Anwendung für Splunk in SAP BTP ist die Überwachung und Analyse von Anwendungsprotokollen: Anwendungen, die auf SAP BTP laufen, erzeugen Protokolle, die von Splunk gesammelt und indiziert werden können. Splunk kann dann verwendet werden, um diese Protokolle zu durchsuchen und zu analysieren und so Einblicke in die Leistung und das Verhalten dieser Anwendungen zu erhalten.

Splunk kann auch für die Sicherheitsüberwachung in SAP BTP verwendet werden. Mit Splunk können Sicherheitsereignisse aus verschiedenen Quellen wie Firewalls, Intrusion-Detection-Systeme und Identitäts- und Zugriffsmanagementsysteme gesammelt und analysiert werden. Splunk kann zur Identifizierung von Sicherheitsbedrohungen und Anomalien eingesetzt werden und ermöglicht so eine rechtzeitige Reaktion auf potenzielle Sicherheitsvorfälle.

2.6 DevOps-Portfolio von SAP

Bislang haben wir die wichtigsten DevOps-Tools für alle DevOps-Phasen untersucht. Während viele dieser Tools nahtlos mit SAP-Lösungen und -Plattformen zusammenarbeiten, gibt es auch eine breite Palette von Tools und Services, die von SAP angeboten werden. Werfen wir einen kurzen Blick auf diese, wobei wir uns speziell auf die in SAP S/4HANA angebotenen DevOps-Tools und die von SAP BTP angebotenen Tools konzentrieren. In späteren Kapiteln beschäftigen wir uns näher mit diesen Tools und betrachten auch praktische Beispiele aus der Praxis.

2.6.1 In SAP S/4HANA angebotene DevOps-Tools

SAP S/4HANA ist die Enterprise Resource Planning (ERP) Business Suite, die auf der In-Memory-Datenbank SAP HANA läuft. Sie ist der Nachfolger von SAP ERP und SAP Business Suite und wurde entwickelt, um Geschäftsprozesse zu rationalisieren und zu vereinfachen und gleichzeitig Echtzeiteinblicke in den Geschäftsbetrieb zu ermöglichen.

Das Hauptziel von SAP S/4HANA war es, das traditionelle ERP-System zu modernisieren und seine Geschwindigkeit und Benutzungsfreundlichkeit zu verbessern. Dies wurde durch ein vereinfachtes Datenmodell, eine schlankere Architektur und eine neu gestaltete Benutzeroberfläche mit der kachelbasierten SAP Fiori UX er-

Einleitung

Die Zeiten, in denen Grundsätze wie »berühre niemals ein laufendes System« oder »Entwicklung und Echtzeit-Support sind zwei verschiedene Dinge« galten, sind vorbei. Es ist keine Überraschung, dass IT-Organisationen unter Druck stehen, schneller zu liefern, um der Konkurrenz voraus zu sein und das Nutzererlebnis für End-User kontinuierlich zu verbessern.

Nehmen Sie ein beliebiges erfolgreiches IT-Unternehmen, einschließlich Amazon, Microsoft, Google und SAP: Sie alle haben bewiesen, dass es durchaus möglich ist, neue Lösungen kontinuierlich und schnell zu ändern und bereitzustellen. Unternehmen sind heute bestrebt, neuen Code monatlich, wöchentlich und manchmal sogar täglich in die Produktion zu bringen. Vor allem nach der Umstellung auf die Cloud gibt es einige Unternehmen, deren Ingenieurfachkräfte den Code Tausende Male pro Tag bereitstellen. Dies wurde durch den (nicht mehr ganz so neuen) Trend *DevOps* erreicht, der es IT-Organisationen auf der ganzen Welt ermöglicht hat, beispiellose Geschwindigkeit, Stabilität, Produktivität und vor allem Kundenzufriedenheit zu erreichen.

DevOps mit SAP wurde als aufschlussreiches und umfassendes Handbuch für Fachleute konzipiert, die DevOps-Praktiken im Kontext von SAP implementieren möchten. Es erforscht die Schnittstelle zwischen DevOps und SAP und richtet sich an SAP-Fachleute, Softwareentwickler*innen, IT-Verantwortliche und DevOps-Ingenieurfachkräfte, die DevOps-Praktiken in ihrer SAP-Landschaft einführen und ihre Softwarebereitstellungsprozesse optimieren möchten.

Das Buch beginnt mit einer Einführung in die Grundlagen von DevOps und deren Bedeutung in der heutigen Softwareentwicklungslandschaft. Anschließend stellen wir die in DevOps verwendeten Tools und Technologien wie Git, Jenkins, Docker, Kubernetes und andere vor und erläutern, wie sie in eine SAP-Landschaft integriert werden können. Danach gehen wir auf die spezifischen Implementierungen und Herausforderungen bei der Umsetzung von DevOps im Kontext von SAP On-premise und SAP Business Technology Platform (SAP BTP) ein. Das Buch behandelt auch, wie man DevOps in einer hybriden Landschaft effektiv implementiert. Am Ende bietet es Informationen zu Best Practices. Im gesamten Buch betonen wir die Bedeutung der Zusammenarbeit und Kommunikation zwischen verschiedenen Teams, einschließlich Entwicklungsteams, Betriebsteams und Stakeholdern. Außerdem betonen wir die Notwendigkeit einer Denkweise der kontinuierlichen Verbesserungsmentalität, bei der die Teams stets bemüht sind, ihre Prozesse und Arbeitsabläufe zu optimieren.

Das Buch bietet praktische Anleitungen für die Implementierung von DevOps-Prinzipien in SAP. Es geht auch auf häufige Probleme ein, die bei der Einführung von Dev-

Ops auftreten, z. B. den Umgang mit Compliance- und Sicherheitsanforderungen, das Management von Veränderungen in großen und komplexen SAP-Umgebungen und die Messung des Erfolgs von DevOps-Initiativen.

In dem Buch werden auch die Vorteile von DevOps für SAP erörtert, wie z. B. schnellere Markteinführung, verbesserte Qualität, erhöhte Agilität und reduzierte Kosten.

Mit einfachen Worten: Dieses Buch ist ein One-Stop-Shop für alle, die ein umfassendes Wissen über DevOps mit SAP suchen. Egal, ob Sie eine erfahrene SAP-Fachkraft oder ein Neuling auf diesem Gebiet sind, dieses Buch vermittelt Ihnen das Wissen und die Werkzeuge, die Sie benötigen, um DevOps-Praktiken in Ihrer SAP-Umgebung erfolgreich zu implementieren. Es hilft Ihnen auch, die einzigartigen Herausforderungen zu bewältigen, die mit der Implementierung von DevOps in On-Premise-, Cloud- oder Hybrid-Landschaften einhergehen, und bietet Strategien, die sicherstellen, dass Ihre DevOps-Initiativen erfolgreich und nachhaltig sind.

Zielpublikum

Dieses Buch richtet sich an alle Personen oder Teams, die DevOps-Expertise mit SAP-Plattformen und -Lösungen erlangen möchten. Sowohl SAP-Entwicklungsteams als auch SAP-Partner oder Unternehmen, die SAP einsetzen, müssen DevOps einführen, um kürzere Entwicklungszyklen zu erreichen, was sich in einer schnelleren Realisierung von Bereitstellungen, Produktveröffentlichungen und Fehlerbehebungen niederschlägt. Dieses Buch kann IT-Fachleuten, Entwickler*innen und Architektur-Fachkräften helfen, die mit SAP-Software arbeiten und daran interessiert sind, DevOps-Praktiken einzuführen, um die Geschwindigkeit, Qualität und Zuverlässigkeit ihrer Softwareentwicklungs- und Bereitstellungsprozesse zu verbessern.

Das Buch kann insbesondere für die folgenden Gruppen relevant sein:

- SAP-Entwicklungsteams, die lernen möchten, wie sie DevOps-Praktiken in ihren SAP-Projekten implementieren können und die sich für Themen wie kontinuierliche Integration, kontinuierliche Bereitstellung und automatisierte Tests interessieren
- Betriebsfachkräfte, die für die Verwaltung von SAP-Umgebungen zuständig sind und sich für Themen wie Infrastructure as Code (IaC), Automatisierung und Überwachung interessieren
- IT-Verantwortliche und Führungskräfte, die für die Überwachung von SAP-Projekten verantwortlich sind und verstehen möchten, wie DevOps-Praktiken die Effizienz und Effektivität ihrer Softwareentwicklungs- und -bereitstellungsprozesse verbessern können

- Architekt*innen, die für die Entwicklung von SAP-Lösungen verantwortlich sind und lernen möchten, wie sie DevOps-Praktiken in ihre Architektur- und Designentscheidungen einbeziehen können
- Berater*innen und Systemintegrator*innen, die mit SAP-Kunden arbeiten und diese bei der Einführung von DevOps-Praktiken als Teil ihrer Softwareentwicklungs- und Bereitstellungsprozesse unterstützen möchten

Ziel dieses Buchs

Das Ziel dieses Buchs ist es, Ihnen zu verdeutlichen, warum DevOps-Prinzipien für jedes Unternehmen, das in der heutigen Welt im Rahmen einer unvermeidlichen digitalen Transformation wettbewerbsfähig sein oder überleben will, von entscheidender Bedeutung sind. Es bietet ein umfassendes Handbuch für Fachleute, die DevOps-Praktiken im Kontext von SAP implementieren möchten.

Das Buch soll Ihnen helfen, die Prinzipien und Praktiken von DevOps zu verstehen und sie im spezifischen Kontext von SAP anzuwenden, einschließlich der Herausforderungen der SAP-Architektur und der Integration mit anderen Systemen. Es vermittelt ein tiefgreifendes Verständnis von DevOps, verschiedenen DevOps-Tools und deren Implementierung mit SAP-Plattformen und -Lösungen. Ziel des Buchs ist es, praktische Anleitungen für die Implementierung von DevOps-Prinzipien in SAP zu geben. Dazu gehören Themen wie kontinuierliche Integration und Bereitstellung, Infrastrukturautomatisierung, Konfigurationsmanagement, Überwachung und Protokollierung sowie Zusammenarbeit und Kommunikation.

Darüber hinaus vermittelt dieses Buch ein tiefgreifendes Verständnis für die Funktionsweise des gesamten DevOps-Portfolios von SAP-Produkten wie SAP S/4HANA und SAP BTP. Anhand vieler praktischer Beispiele und Implementierungsleitfäden erhalten Sie eine kristallklare Vorstellung davon, wie Sie ein komplettes End-to-End-DevOps-Szenario für SAP-Lösungen in verschiedenen Landschaften, einschließlich On-premise, Cloud und Hybrid gestalten können. Schließlich erfahren Sie auch etwas über die Herausforderungen und Best Practices bei DevOps.

Das Buch enthält Empfehlungen und Best Practices, die Ihnen einen Überblick darüber geben, was Sie bei der Planung von Entwicklungsprojekten auf SAP BTP beachten sollten.

Ein weiteres Ziel des Buchs ist es, häufig auftretende Bedenken und Herausforderungen bei der Einführung von DevOps in SAP zu thematisieren, wie z. B. der Umgang mit Compliance- und Sicherheitsanforderungen, das Management von Veränderungen in großen und komplexen SAP-Umgebungen und die Messung des Erfolgs von DevOps-Initiativen.

Insgesamt besteht das Ziel von *DevOps mit SAP* darin, Ihnen das Wissen und die Tools zu vermitteln, die Sie benötigen, um DevOps-Praktiken in Ihrer SAP-Umgebung erfolgreich zu implementieren, und Ihnen dabei zu helfen, die einzigartigen Herausforderungen zu bewältigen, die mit der Implementierung von DevOps im SAP-Kontext auftreten.

Umfang dieses Buchs

Das Buch deckt ein breites Spektrum an Themen im Zusammenhang mit DevOps und seiner Anwendung in der SAP-Umgebung ab.

- **Kapitel 1** bietet einen Überblick über DevOps und dessen Bedeutung in der heutigen Softwareentwicklungslandschaft. Es bietet einen Überblick über die Prinzipien und Praktiken von DevOps, einer Softwareentwicklungsmethodik, die den Schwerpunkt auf Zusammenarbeit, Kommunikation und Automatisierung zwischen Softwareentwicklungsteams und IT-Betriebsfachkräften legt. Das Kapitel beginnt mit einer Definition von DevOps und dessen Ursprüngen und erläutert Mythen und Fakten über DevOps. Anschließend werden die Vorteile von DevOps erörtert, wie z. B. eine erhöhte Liefergeschwindigkeit, eine verbesserte Qualität und eine gesteigerte Kundenzufriedenheit. Das Kapitel behandelt auch Schlüsselkonzepte von DevOps wie kontinuierliche Integration, kontinuierliche Lieferung und kontinuierliche Bereitstellung und erklärt, wie diese Praktiken Teams helfen können, Software schneller und zuverlässiger zu liefern. Abschließend werden die kulturellen Aspekte von DevOps erörtert, einschließlich der Bedeutung von Kommunikation und Zusammenarbeit für die erfolgreiche Implementierung von DevOps-Praktiken.
- **Kapitel 2** bietet einen detaillierten Überblick über die verschiedenen Tools und Technologien, die in der DevOps-Methodik verwendet werden. Zu Beginn des Kapitels werden die Bedeutung von DevOps-Tools und die verschiedenen Kategorien, in die diese Tools fallen, erörtert. Das Kapitel bietet dann einen Überblick über die verschiedenen Tools in jeder Kategorie: Code- und Versionskontrolltools, Build-Tools, Testautomatisierungstools, Bereitstellungstools und Überwachungstools.

Jede Tool-Kategorie wird ausführlich erläutert und mit Beispielen beliebter Tools aus jeder Kategorie versehen. Das Kapitel befasst sich auch mit den Vorteilen, die der Einsatz dieser Tools in einer DevOps-Umgebung mit sich bringt, wie z. B. schnellere und zuverlässigere Softwarebereitstellung, verbesserte Zusammenarbeit und Kommunikation zwischen den Teammitgliedern sowie verbesserte Sichtbarkeit und Kontrolle über den Softwareentwicklungsprozess. Darüber hinaus wird

erörtert, wie diese Tools zusammen mit SAP-Systemen und -Lösungen eingesetzt werden können.

Gegen Ende des Kapitels werden auch verschiedene SAP-interne DevOps-Tools und -Services behandelt, und es wird ein Überblick über das DevOps-Portfolio in SAP On-Premise- und Cloud-Lösungen einschließlich SAP BTP gegeben.

Insgesamt dient dieses Kapitel als umfassender Leitfaden für Softwareentwicklungs- und IT-Betriebsfachkräfte, die die verschiedenen Tools und Technologien der DevOps-Methodik verstehen und nutzen möchten.

- **Kapitel 3** bietet einen detaillierten Leitfaden für die Implementierung von DevOps-Praktiken in On-Premise-SAP-Systemen, wobei der Schwerpunkt auf SAP S/4HANA On-premise liegt. Das Kapitel beginnt mit einer Einführung in DevOps und verschiedene Tools und Implementierungsoptionen für SAP S/4HANA.

Es gibt dann einen Überblick über abapGit, ein Tool, das Versionskontrolle und Zusammenarbeit bei der ABAP-Entwicklung ermöglicht. Es erläutert die Installation, Konfiguration und Beispielprojekte für abapGit. Darüber hinaus behandelt das Kapitel das Change and Transport System (CTS), das verbesserte Change and Transport System (CTS+), das Git-basierte Change and Transport System (gCTS) und das zentrale Change and Transport System (cCTS) sowie das kontinuierliche Testen in ABAP mit Tools wie ABAP Unit und Code Inspector.

Das Kapitel bietet eine schrittweise Anleitung für die Einrichtung einer ABAP-Continuous-Integration-and-Delivery-(CI/CD-)Pipeline, die eine kontinuierliche Integration mit Tools wie Jenkins, eine kontinuierliche Auslieferung mit SAP-Transportmanagementtools und eine kontinuierliche Bereitstellung mit automatisierten Deployment-Tools umfasst.

Abschließend wird ein End-to-End-DevOps-Szenario in SAP S/4HANA On-premise vorgestellt, das zeigt, wie die verschiedenen DevOps-Praktiken und -Tools integriert werden können, um Software schneller und zuverlässiger bereitzustellen.

- **Kapitel 4** bietet einen umfassenden Leitfaden für die Implementierung von DevOps-Praktiken mithilfe der von SAP BTP angebotenen Tools und Services. Das Kapitel beginnt mit einem Überblick über das DevOps-Portfolio von SAP BTP, das Tools für die verschiedenen Phasen des DevOps-Lebenszyklus umfasst, z. B. für Planung, Entwicklung, Testen, Freigabe, Bereitstellung und Betrieb.

Das Kapitel bietet detaillierte Informationen zu den verschiedenen Tools, die SAP BTP für jede Phase des DevOps-Lebenszyklus anbietet. Dazu gehören das SAP BTP Cockpit, das eine zentrale Verwaltungskonsole für SAP-BTP-Services bietet, sowie CI/CD-Services zur Automatisierung des Build-, Test- und Bereitstellungsprozesses. Das Kapitel bietet eine detaillierte praktische Anleitung für verschiedene CI/CD-Angebote von SAP: SAP Continuous Integration and Delivery, das Projekt »Piper« und der Continuous Integration and Delivery Best Practices Guide.

Das Kapitel befasst sich auch mit den Services und Frameworks, die in den Entwicklungs-, Build- und Testphasen verwendet werden, und geht ausführlich auf SAP Business Application Studio ein, eine integrierte Entwicklungsumgebung für die Erstellung und Bereitstellung von Cloud-nativen Anwendungen mit dem SAP Cloud Application Programming Model. Außerdem wird die Verwendung von Multitarget-Anwendungen (MTAs) für die Erstellung komplexer Anwendungen sowie das SAP Cloud SDK, eine Reihe von Bibliotheken und Tools für die Erstellung von Cloud-nativen Anwendungen auf SAP BTP, behandelt.

Im weiteren Verlauf des Kapitels werden Services für die Freigabe- und Bereitstellungsphasen behandelt, wie der SAP-Cloud-Transport-Management-Service, mit dem Sie Lösungsartefakte (z. B. MTA-Archive) zusammen mit deren Inhalten (z. B. SAP-Integration-Suite-Inhalte) bereitstellen und die Bereitstellung und den Betrieb über mehrere SAP-BTP-Konten hinweg verwalten können. Das Kapitel behandelt auch Services, die mit den Phasen Betrieb und Überwachung in Verbindung stehen, wie SAP Alert Notification Service für SAP BTP und SAP Automation Pilot. Schließlich wird am Ende des Kapitels ein End-to-End-DevOps-Szenario für Cloud-Landschaften mit SAP BTP behandelt.

Insgesamt dient dieses Kapitel als umfassender Leitfaden für SAP-Entwicklungsteams und IT-Betriebsfachkräfte, die DevOps-Praktiken mit SAP-BTP-Tools und -Services implementieren möchten. Das Kapitel bietet praktische Anleitungen und Beispiele aus der Praxis und zeigt, wie Unternehmen SAP BTP nutzen können, um Software schneller und zuverlässiger liefern.

- **Kapitel 5** gibt einen Überblick über DevOps-Praktiken im Kontext von SAP BTP, ABAP-Umgebung. Das Kapitel beginnt mit einer Einführung in die SAP BTP, ABAP-Umgebung und deren Bedeutung.

Anschließend werden die verschiedenen CI/CD-Tools in SAP BTP, ABAP-Umgebung, wie Git-Codeverwaltung, ABAP-Umgebungspipeline und Jenkins-Job-Erstellung besprochen. Die Git-Codeverwaltung ermöglicht es Entwicklungsteams, gemeinsam an Codeänderungen zu arbeiten und Änderungen zu verfolgen, während die ABAP-Umgebungspipeline den Build-, Test- und Bereitstellungsprozess automatisiert.

In diesem Kapitel wird auch untersucht, wie ein Jenkins-Job mit der ABAP-Umgebungspipeline erstellt werden kann, um den Entwicklungs- und Bereitstellungsprozess zu automatisieren. Außerdem wird die Rolle von SAP Cloud ALM für das Testmanagement hervorgehoben, das eine Reihe von Tools für die Verwaltung von Testszenarien, Testfällen und Testplänen für SAP-Anwendungen bietet.

Schließlich wird in diesem Kapitel die Verwendung von SAP Cloud Transport Management für SAP BTP, ABAP-Umgebung, beschrieben, mit dem Transporte und Änderungen über Systeme und Landschaften hinweg verwaltet und kontrolliert werden können, um sicherzustellen, dass Codeänderungen sicher und effizient implementiert werden.

Insgesamt bietet dieses Kapitel eine aufschlussreiche Einführung in DevOps-Praktiken in der SAP BTP, ABAP-Umgebung sowie in die CI/CD-Tools, die zur Optimierung des Softwareentwicklungsprozesses zur Verfügung stehen.

- **Kapitel 6** befasst sich mit DevOps-Praktiken für Unternehmen, die mit einer hybriden SAP-Landschaft arbeiten. Es behandelt verschiedene Tools und ihren Zweck in hybriden Szenarien, einschließlich CI/CD in hybriden Landschaften, Betriebsüberwachung mit SAP Solution Manager und die Integration von SAP Cloud Transport Management mit SAP Solution Manager.

Das Kapitel behandelt auch die entscheidende Rolle von SAP Solution Manager bei der Verwaltung des DevOps-Lebenszyklus, der Integration verschiedener Tools und der Bereitstellung von End-to-End-Sichtbarkeit und -Kontrolle. Darüber hinaus wird die DevOps-Befähigung für hybride Lösungen mit Git und dem Projekt »Piper« für SAP Data Intelligence erörtert. Abschließend wird ein End-to-End-DevOps-Szenario für eine hybride Landschaft durchgespielt.

Insgesamt bietet dieses Kapitel einen umfassenden Leitfaden für Unternehmen zur Nutzung von DevOps-Praktiken und -Tools, um Innovationen zu beschleunigen, die Agilität zu verbessern und die Leistung und Kosten der hybriden SAP-Landschaft zu optimieren.

- **Kapitel 7** bietet einen detaillierten Überblick über verschiedene Best Practices, die Unternehmen für eine erfolgreiche DevOps-Implementierung einsetzen können. Das Kapitel deckt mehrere Bereiche ab, darunter CI/CD, Microservices, IaC, Überwachung und Protokollierung sowie Kommunikation und Zusammenarbeit.

Das Kapitel unterstreicht die Bedeutung einer standardisierten und automatisierten CI/CD-Pipeline, um den Softwareentwicklungsprozess zu optimieren und die Effizienz zu steigern. Außerdem werden Empfehlungen für die Implementierung einer Microservices-Architektur gegeben, um monolithische Anwendungen in kleinere, unabhängige Dienste aufzuteilen, die einfacher zu verwalten und zu skalieren sind.

In diesem Kapitel werden die Vorteile der Verwendung von IaC für die konsistente und wiederholbare Verwaltung und Bereitstellung der Infrastruktur erörtert und Best Practices für die Implementierung von IaC vorgestellt. Darüber hinaus enthält das Kapitel Best Practices für die Überwachung und Protokollierung, einschließlich der Bedeutung der Einrichtung von Warnungen und Benachrichtigungen, der Verwendung einer zentralisierten Protokollierung und der Implementierung eines Dashboards zur Verfolgung von Metriken und KPIs.

Am Ende des Kapitels wird die Bedeutung von Kommunikation und Zusammenarbeit für eine erfolgreiche DevOps-Implementierung hervorgehoben. Es wird empfohlen, eine DevOps-Kultur einzuführen, die Transparenz, Zusammenarbeit und funktionsübergreifende Teams fördert.

Insgesamt bietet das Kapitel praktische Empfehlungen und Best Practices, die Unternehmen auf ihrem Weg zu DevOps anwenden können, um ihre Agilität zu verbessern, die Markteinführungszeit zu verkürzen und qualitativ hochwertige Softwareprodukte zu liefern.

- **Kapitel 8** konzentriert sich auf die entscheidende Rolle der Sicherheit im DevOps-Prozess und bietet praktische Empfehlungen für Unternehmen zur Implementierung von Sicherheitsmaßnahmen in ihrer DevOps-Pipeline. Das Kapitel deckt mehrere sicherheitsrelevante Bereiche in DevOps ab, darunter Secure Code Analytics, automatisierte Sicherheitstests, Überwachung von Sicherheitslücken und Implementierung.

In diesem Kapitel wird betont, wie wichtig es ist, die Sicherheit von Anfang an in den Entwicklungsprozess einzubeziehen, indem sichere Codierungspraktiken angewandt und Tools zur sicheren Codeanalyse implementiert werden, um Sicherheitsanfälligkeiten frühzeitig im Entwicklungszyklus zu erkennen. Es enthält Empfehlungen für die Integration automatisierter Sicherheitstests in die CI/CD-Pipeline, um sicherzustellen, dass die Sicherheitstests kontinuierlich und effizient durchgeführt werden.

In diesem Kapitel wird erörtert, wie wichtig die Überwachung von Sicherheitsanfälligkeiten in Produktionssystemen ist, und es wird empfohlen, Tools wie Schwachstellen-Scanner und Penetrationstests einzusetzen, um Sicherheitslücken zu erkennen und zu beheben. Außerdem wird auf die Notwendigkeit hingewiesen, Sicherheitsrichtlinien und -standards festzulegen und zu befolgen sowie regelmäßige Sicherheitsaudits und Risikobewertungen durchzuführen. Darüber hinaus enthält das Kapitel Empfehlungen für die Implementierung von Sicherheitskontrollen wie Zugriffskontrollen, Verschlüsselung sowie Überwachung und Protokollierung zum Schutz von Anwendungen und Systemen vor externen und internen Bedrohungen.

Insgesamt bietet das Kapitel praktische Anleitungen für Unternehmen zur Einführung einer umfassenden Sicherheitsstrategie und zur Integration von Sicherheit in ihre DevOps-Pipeline. Auf diese Weise können Unternehmen Sicherheitsrisiken mindern, die Einhaltung von Vorschriften verbessern und ihre Vermögenswerte und ihren Ruf schützen.

- **Kapitel 9**, das letzte Kapitel, bietet eine Roadmap für die zukünftige Entwicklung von DevOps im Allgemeinen und DevOps mit SAP.

Zusammenfassend lässt sich sagen, dass das Buch eine umfassende Ressource für alle ist, die mit SAP-Entwicklung, -Betrieb oder -Verwaltung zu tun haben. Das Buch ist praxisorientiert und umsetzbar und vermittelt Ihnen das Wissen und die Tools, die Sie für die erfolgreiche Implementierung von DevOps-Praktiken in Ihrer SAP-Umgebung benötigen.

Wie Sie dieses Buch lesen sollten

Dieses Buch ist auch für Sie geeignet, falls Ihnen DevOps völlig neu ist. Es gibt keine Voraussetzung, um die ersten beiden Kapitel verstehen zu können. Kapitel 1 legt den Grundstein für Ihre Lernreise. Dieses Kapitel ist das wichtigste, um zu verstehen, was DevOps ist und warum es für jedes Unternehmen, ob klein oder groß, so wichtig geworden ist. Hier werden Sie verstehen, warum DevOps viel mehr als nur eine technische Lösung ist – es ist eine Evolution von Kultur, Prozessen und Tools.

Kapitel 2 bietet eine Liste von Tools und erläutert deren Details. Während die Idee von DevOps oft mit Denkweise, Kultur und Zusammenarbeit in Verbindung gebracht wird, benötigen Sie eine Vielzahl von Tools und Frameworks, um DevOps-bezogene Aufgaben zu implementieren. Gehen Sie jedes einzelne davon durch, aber stellen Sie vor allem sicher, dass Sie die ganzheitliche Sicht auf den DevOps-Zyklus verstehen. Dies ist das Kapitel, zu dem Sie zurückkehren und Ihr Wissen auffrischen können, wenn Sie in den nachfolgenden Kapiteln mit plattform- und lösungsspezifischen Implementierungen arbeiten.

Ab Kapitel 3 konzentriert sich das Buch mehr auf eine bestimmte Plattform und Lösungen. Kapitel 3 konzentriert sich auf On-Premise-SAP-Systeme wie SAP S/4HANA. Ein wenig ABAP-Kenntnisse sind erforderlich, um das gesamte Kapitel, insbesondere die praktischen Beispiele, zu verstehen. Stellen Sie sicher, dass Sie den praktischen Beispielen folgen und praktische Erfahrungen sammeln.

In Kapitel 4 geht es um das neueste und wichtigste Angebot der SAP BTP und die DevOps-Implementierung in diesem Rahmen. Dieses Kapitel konzentriert sich auf das DevOps-Portfolio von SAP BTP. Daher erwarten wir, dass Sie einige Kernkonzepte kennen. Zunächst einmal sollten Sie ein grundlegendes Verständnis von Cloud Computing, dem Platform-as-a-Service-Modell (PaaS) und Cloud-nativer Entwicklung haben. Außerdem sollten Sie ein allgemeines Verständnis von SAP BTP haben: was SAP BTP ist, welche Arten von Anwendungsfällen in SAP BTP realisiert werden können, das SAP Cloud Application Programming Model usw. Es könnte eine gute Idee sein, sich mit dem SAP Cloud Application Programming Model vertraut zu machen und einige grundlegende Anwendungen zu erstellen – und Sie können diese Anwendungen nutzen, um die praktische Arbeit zu erledigen und ein klares Verständnis des Themas zu erhalten.

SAP Cloud Application Programming Model

Wenn Sie mehr über das SAP Cloud Application Programming Model wissen möchten, empfehlen wir das Buch »Anwendungsentwicklung auf der SAP Cloud Plattform« von Klaus Kopecz, erschienen bei SAP PRESS (siehe www.sap-press.de/5110).



Kapitel 4 bietet viel praktisches Wissen. Nutzen Sie diese praktischen Beispiele, um Ihr eigenes DevOps-Modell und Ihre eigene Pipeline zu konfigurieren. Gegen Ende des Kapitels tauchen wir tief in verschiedene Betriebs- und Überwachungstools ein. Versuchen Sie, diese zu installieren/konfigurieren und die Echtzeitleistung Ihrer eigenen Anwendung zu überprüfen. Kapitel 4 behandelt hauptsächlich das DevOps-Portfolio von SAP BTP, Cloud-Foundry-Umgebung. Obwohl die meisten dieser DevOps-Services auch für SAP BTP, ABAP-Umgebung anwendbar sind, kann die tatsächliche Implementierung abweichen.

Kapitel 5 enthält Details zur DevOps-Implementierung in SAP BTP, ABAP-Umgebung. Bevor Sie mit Kapitel 5 beginnen, sollten Sie ein Grundverständnis von ABAP haben. Wenn es um die tatsächliche Implementierung einer Anwendung in SAP BTP geht, müssen Sie zwischen der Cloud-Foundry- und der ABAP-Umgebung wählen. Die ABAP-Umgebung ist hauptsächlich für Personen gedacht, die über ABAP-Kenntnisse verfügen und diese nutzen möchten. Wenn Sie nicht in diese Kategorie fallen, können Sie dieses Kapitel auslassen. Wir empfehlen jedoch, auch Kapitel 5 abzuschließen, damit Sie ein ganzheitliches Verständnis von DevOps in SAP BTP über alle Umgebungen hinweg erlangen.

Während Kapitel 3 sich auf On-Premise-SAP-Systeme und Kapitel 4 und 5 auf SAP BTP konzentrieren, müssen wir in der Praxis oft mit hybriden Landschaften umgehen. DevOps für hybride Landschaften wird in Kapitel 6 behandelt. Dies ist ein sehr interessantes und wichtiges Kapitel, das Ihnen einen Einblick in eine tatsächliche Entwicklungslandschaft gibt. Gehen Sie dieses Kapitel unbedingt gründlich durch und absolvieren Sie alle praktischen Übungen.

Wenn Sie alles bis Kapitel 6 durchgearbeitet haben, sollten Sie ein ganzheitliches und detailliertes Verständnis von DevOps in SAP-Ökosystemen haben. Es ist jedoch sehr wichtig, von den Erfahrungen anderer zu lernen. Der Best-Practices-Leitfaden in Kapitel 7 hilft Ihnen, die wichtigsten Dos und Don'ts bei der DevOps-Implementierung zu verstehen. Bevor Sie mit einer DevOps-Implementierung in der Praxis beginnen, sollten Sie dieses Kapitel gründlich lesen. Dies ist ein weiteres Kapitel, auf das Sie bei verschiedenen Szenarien möglicherweise immer wieder zurückgreifen müssen.

Kapitel 8 behandelt die Sicherheitsaspekte von DevOps. Auch wenn es optional erscheinen mag, sind diese Informationen äußerst wichtig. Denken Sie immer daran, dass selbst das beste System oder die beste Lösung der Welt nutzlos ist, wenn sie nicht sicher ist.

In Kapitel 9 schließlich können Sie die Roadmap für die zukünftige Entwicklung von DevOps im Allgemeinen und DevOps mit SAP durchgehen.

Auf einen Blick

1	Einführung in DevOps	27
2	DevOps-Tools	59
3	DevOps für SAP-On-Premise-Systeme	109
4	DevOps mit SAP Business Technology Platform	181
5	DevOps für SAP BTP, ABAP-Umgebung	247
6	DevOps für hybride SAP-Systeme	291
7	Best Practices	345
8	Sicherheit	367
9	Ausblick	405

Inhalt

Danksagung	15
Einleitung	17

1 Einführung in DevOps 27

1.1 Was ist DevOps?	27
1.1.1 Probleme mit DevOps lösen	28
1.1.2 Definition von DevOps	30
1.1.3 Vier Prinzipien von DevOps	31
1.1.4 Die wichtigsten Vorteile von DevOps	32
1.2 Kontinuierliche Integration und Auslieferung	33
1.3 Wie funktioniert DevOps?	36
1.3.1 Planungsphase	36
1.3.2 Entwicklungsphase	37
1.3.3 Build-Phase	38
1.3.4 Testphase	38
1.3.5 Freigabephase	39
1.3.6 Bereitstellungsphase	39
1.3.7 Betriebsphase	40
1.3.8 Überwachungsphase	41
1.3.9 Leitlinien für DevOps	42
1.4 Historie von DevOps	42
1.5 Mythen und Missverständnisse über DevOps	43
1.6 Wer braucht DevOps?	45
1.7 DevOps-Kultur	48
1.8 Herausforderungen bei der Implementierung von DevOps	50
1.8.1 Tools und Services	50
1.8.2 Kulturwandel	52
1.8.3 Fehlende Vision und Unterstützung durch das Topmanagement	53
1.8.4 Zu starker Fokus auf Technologien	55
1.9 Zusammenfassung	57

2 DevOps-Tools 59

2.1	Tools zur Code- und Versionskontrolle	59
2.1.1	Git	60
2.1.2	GitHub	63
2.2	Build-Tools	65
2.2.1	Maven	65
2.2.2	Chef	68
2.2.3	Jenkins	69
2.3	Testautomatisierungstools	72
2.3.1	Selenium	74
2.3.2	JUnit	77
2.3.3	SonarQube	80
2.3.4	Jenkins	83
2.4	Bereitstellungstools	84
2.4.1	Docker	85
2.4.2	Kubernetes	91
2.4.3	Jenkins	94
2.4.4	GitLab	95
2.4.5	Chef	96
2.5	Überwachungstools	97
2.5.1	Dynatrace	98
2.5.2	Grafana	99
2.5.3	Splunk	101
2.6	DevOps-Portfolio von SAP	102
2.6.1	In SAP S/4HANA angebotene DevOps-Tools	102
2.6.2	DevOps-Tools in der SAP Business Technology Platform	105
2.7	Zusammenfassung	107

3 DevOps für SAP-On-Premise-Systeme 109

3.1	Einführung in DevOps in SAP S/4HANA	109
3.1.1	DevOps-Funktionen in SAP S/4HANA	110
3.1.2	Entwicklung des Change and Transport Systems	113
3.2	abapGit	117
3.2.1	Einführung	118
3.2.2	Installation	118

3.2.3	Konfiguration	122
3.2.4	Implementierung eines Beispielprojekts	129
3.2.5	Entwicklungsrichtlinien und Best Practices	130
3.3	Change and Transport System	136
3.3.1	Einführung	136
3.3.2	Konfiguration	140
3.3.3	Durchführung eines Beispielprojekts	145
3.4	Verbessertes Change and Transport System	145
3.4.1	Einführung	146
3.4.2	Konfiguration	147
3.4.3	Durchführung eines Beispielprojekts	149
3.4.4	Integration mit SAP Solution Manager	149
3.5	Zentrales Change and Transport System	150
3.5.1	Einführung	150
3.5.2	Konfiguration	152
3.5.3	Durchführung eines Beispielprojekts	154
3.5.4	Integration mit SAP Solution Manager	155
3.6	Kontinuierliche Integration für ABAP On-premise mit Git-fähigem CTS ...	155
3.6.1	Einführung	156
3.6.2	Konfiguration	156
3.6.3	Einrichtung und Durchführung	157
3.6.4	Erstellung eines GitHub-Repositorys	157
3.6.5	Authentifizierung in gCTS	158
3.6.6	Integration mit der ABAP Workbench	158
3.6.7	Integration mit dem Change Request Management	159
3.7	SAP HANA Transport für ABAP	159
3.8	Kontinuierliches Testen in ABAP	162
3.8.1	Einführung	162
3.8.2	Unit-Testen mit GitHub-Actions	166
3.8.3	SAP Solution Manager Test Suite und Focused Build für Tests	169
3.9	ABAP-Pipeline für kontinuierliche Integration und Auslieferung	171
3.9.1	SAP Continuous Integration and Delivery	171
3.9.2	SAP Solution Manager	171
3.9.3	Projekt »Piper«	172
3.9.4	ABAP Development Tools für Eclipse	176
3.9.5	Tools von Drittanbietern	176
3.10	End-to-End-DevOps-Szenario für SAP S/4HANA	176
3.10.1	Planungsphase	177
3.10.2	Entwicklungs-, Build- und Testphasen	178

3.10.3	Freigabe- und Bereitstellungsphasen	179
3.10.4	Betriebs- und Überwachungsphasen	180
3.11	Zusammenfassung	180

4 DevOps mit SAP Business Technology Platform 181

4.1	DevOps-Portfolio von SAP BTP	181
4.2	Services für die Planungsphase	184
4.2.1	SAP-BTP-Landschaftseinrichtung	185
4.2.2	Best Practices und Leitlinien für die Planungsphase	187
4.2.3	Kontinuierliche Integration und Auslieferung in SAP BTP	188
4.2.4	Konfigurieren und Ausführen einer CI/CD-Pipeline für ein SAP-Fiori-Projekt	194
4.2.5	Konfigurieren und Ausführen einer CI/CD-Pipeline für ein SAP-Cloud-Application-Programming-Model-Projekt	209
4.2.6	Konfigurieren und Ausführen einer CI/CD-Pipeline mit dem Projekt »Piper«	212
4.2.7	Die Auswahl der richtigen CI/CD-Lösung von SAP	215
4.3	Services für die Entwicklungs-, Build- und Testphasen	216
4.3.1	SAP-BTP-Entwicklungsumgebung und -Programmiermodelle	217
4.3.2	SAP Business Application Studio	218
4.3.3	SAP Cloud Application Programming Model	224
4.3.4	Multitarget Application	226
4.3.5	SAP Cloud SDK	230
4.4	Services für die Freigabe- und Bereitstellungsphasen	231
4.4.1	Continuous Delivery	231
4.4.2	SAP Cloud Transport Management	232
4.5	Services für die Betriebs- und Überwachungsphasen	238
4.5.1	SAP-Alert-Notification-Service für SAP BTP	239
4.5.2	SAP Automation Pilot	241
4.6	Ein End-to-End-DevOps-Szenario für Cloud-Landschaften mit SAP BTP	242
4.6.1	Planungsphase	243
4.6.2	Entwicklungs-, Build- und Testphase	244
4.6.3	Freigabe- und Bereitstellungsphasen	245
4.6.4	Betriebs- und Überwachungsphasen	246
4.7	Zusammenfassung	246

5	DevOps für SAP BTP, ABAP-Umgebung	247
5.1	Einführung in SAP BTP, ABAP-Umgebung	248
5.1.1	Was ist SAP BTP, ABAP-Umgebung?	248
5.1.2	Entwicklung von ABAP	250
5.1.3	Notwendigkeit einer ABAP-Umgebung in SAP BTP	252
5.1.4	Vorteile der ABAP-Umgebung in SAP BTP	253
5.2	Tools für die kontinuierliche Integration und Auslieferung in SAP BTP, ABAP-Umgebung	253
5.2.1	Git-Codeverwaltung	254
5.2.2	APIs	255
5.2.3	ABAP-Umgebungspipeline	256
5.2.4	Jenkins und Projekt »Piper«	261
5.3	Entwicklung von Anwendungen mit SAP Business Application Studio	267
5.3.1	ABAP RESTful Application Programming Model	268
5.3.2	ABAP Development Tools für die ABAP-Umgebung	271
5.3.3	SAP Business Application Studio	272
5.4	SAP Cloud ALM für das Testmanagement	278
5.4.1	Produktübersicht	278
5.4.2	Testmanagement-Funktion von SAP Cloud ALM	280
5.5	SAP Cloud Transport Management für SAP BTP, ABAP-Umgebung	287
5.5.1	Konfiguration für den Transportexport	287
5.5.2	Konfiguration für den Transportimport	289
5.6	Zusammenfassung	290
6	DevOps für hybride SAP-Systeme	291
6.1	Einführung in das hybride Änderungsmanagement	292
6.1.1	Intelligente Unternehmen mit SAP	292
6.1.2	Hybride SAP-Systeme	294
6.1.3	Hybrides Änderungsmanagement	297
6.2	Verfügbare Tools und ihr Zweck	299
6.2.1	Kontinuierliche Integration und Auslieferung	300
6.2.2	Projekt »Piper«	302
6.2.3	SAP Cloud Transport Management	305
6.2.4	Verbessertes Change and Transport System	306
6.2.5	Application Lifecycle Management	306

6.3	Tool-Integration beim Änderungsmanagement in hybriden Szenarien	313
6.3.1	SAP Cloud Transport Management mit CTS+ und SAP Solution Manager integrieren	313
6.3.2	SAP Cloud ALM mit externen Vorfallsystemen integrieren	319
6.4	Kontinuierliche Entwicklung	322
6.4.1	Entwicklung	322
6.4.2	Testmanagement	323
6.4.3	Freigabemanagement	325
6.5	Betriebsüberwachung mit SAP Solution Manager	326
6.5.1	Integrationsüberwachung	326
6.5.2	Jobmanagement	328
6.5.3	Ausnahmemanagement	329
6.5.4	Datenkonsistenzmanagement	330
6.5.5	Geschäftsprozessüberwachung	332
6.5.6	Nutzererlebnisüberwachung	333
6.6	DevOps für SAP Data Intelligence	334
6.6.1	Überblick über die Lösung	334
6.6.2	Kontinuierliche Integration und Auslieferung	335
6.6.3	Interner Transport	339
6.6.4	Überwachung	339
6.7	End-to-End-DevOps-Szenario für eine hybride Landschaft	340
6.7.1	Planungsphase	341
6.7.2	Entwicklungs-, Build- und Testphasen	342
6.7.3	Freigabe- und Bereitstellungsphasen	342
6.7.4	Betriebs- und Überwachungsphasen	343
6.8	Zusammenfassung	344
7	Best Practices	345
7.1	Kontinuierliche Integration und kontinuierliche Auslieferung	346
7.1.1	Kontinuierliche Integration	346
7.1.2	Kontinuierliche Auslieferung	352
7.2	Microservices	358
7.3	Infrastructure as Code	360
7.4	Überwachung und Protokollierung	362

7.5	Kommunikation und Zusammenarbeit	363
7.6	Fail-Fast	364
7.7	Zusammenfassung	366

8 Sicherheit 367

8.1	Gängige Sicherheit-Tools und -Konzepte	368
8.2	Sicherheitsbezogene Codeanalysen	370
8.2.1	Manuelle Techniken zur sicherheitsbezogenen Codeanalyse	370
8.2.2	ABAP-bezogene Scans	374
8.2.3	Java-bezogene Scans	379
8.2.4	SAP-Fiori-bezogene Scans	389
8.3	Automatisierte Sicherheitstests	397
8.4	Überwachung von Sicherheitslücken	399
8.4.1	Scannen auf Schwachstellen	399
8.4.2	Kontinuierliche Überwachung	401
8.4.3	Best Practices	402
8.5	Implementierung	403
8.6	Zusammenfassung	404

9 Ausblick 405

9.1	Aufkommende DevOps-Technologien	405
9.1.1	SAP Build	406
9.1.2	Code-Inspector	407
9.1.3	SAP Focused Run	407
9.1.4	Cloud-basierte Technologien	408
9.2	DevOps für SAP BTP, Kyma-Laufzeit	410
9.3	Zukünftige Richtungen von DevOps in SAP	410
9.3.1	Künstliche Intelligenz für den IT-Betrieb	411
9.3.2	No Operations	411
9.3.3	GitOps	412
9.3.4	DevSecOps	412
9.3.5	Low-Code/No-Code	413
9.3.6	Die Rolle der Cloud-Einführung bei DevOps	413

9.4 Die ersten Schritte auf Ihrer DevOps-Reise 414

9.5 Wie man auf dem Laufenden bleibt 415

9.6 Zusammenfassung 418

Die Autoren 419

Index 421

Prinzipien und Tools für Ihre SAP-Entwicklungsprojekte

DevOps in On-Premise-Landschaften

Implementieren Sie DevOps in Ihrem On-Premise-SAP-System mithilfe bewährter Methoden. Lernen Sie, wie Sie Werkzeuge wie abapGit und Git-enabled Change and Transport System (gCTS) installieren, konfigurieren und einsetzen.

DevOps in Cloud-Umgebungen

Richten Sie die DevOps-Cloud-Services und -Frameworks auch in SAP BTP ein. Von der Entwicklung mit SAP BTP über die ABAP-Umgebung bis hin zur Bereitstellung mit dem Projekt »Piper« lernen Sie alle aktuellen Tools kennen.

DevOps in hybriden Landschaften

Die Umsetzung von DevOps in hybriden SAP-Systemen hat ihre eigenen Herausforderungen. Lernen Sie, wie Sie diese mit SAP Solution Manager, SAP Cloud ALM, SAP Cloud Transport Management und anderen Services bewältigen.

Auf einen Blick

- Für On-Premise, Cloud und hybrid
- Werkzeuge wie Git, Maven, Jenkins und Co.
- Continuous Integration and Delivery (CI/CD)
- DevOps in SAP S/4HANA und SAP BTP
- Verbesserte Versionsverwaltung
- Richtlinien für alle Phasen der Entwicklung
- Effizientes Application Lifecycle Management
- Operation Monitoring
- Sicherheitstools und -konzepte
- Best Practices für den Einsatz

»DevOps und SAP – so gelingt der Kulturwandel«



Das Autorenteam

Dr. Merza Klaghstan und Raja Gupta sind Solution Architects bei SAP. Zusammen haben sie über 20 Jahre Erfahrung mit SAP-Entwicklungsprojekten. Sandip Jha ist Entwicklungsarchitekt bei SAP und verfügt über mehr als ein Jahrzehnt Erfahrung in der SAP-Produktentwicklung.

