## Tagging

In diesem Dokument werden das Git-Konzept der Kennzeichnung und der git tag Befehl erläutert. Tags sind Refs, die auf bestimmte Punkte im Git-Verlauf verweisen. Tagging wird im Allgemeinen verwendet, um einen Punkt im Verlauf zu erfassen, der für eine markierte Version (dh v1.0.1) verwendet wird. Ein Tag ist wie ein Branch, der sich nicht ändert. Im Gegensatz zu Branches haben Tags nach ihrer Erstellung keinen weiteren Commit-Verlauf.

## Tag erstellen

Führen Sie den folgenden Befehl aus, um ein neues Tag zu erstellen:

git tag <tagname>

Ersetzen Sie <tagname> mit dem Status des Repos zum Zeitpunkt der Erstellung des Tags durch einen semantischen Bezeichner. Ein übliches Muster ist die Verwendung von Versionsnummern wie git tag v1.4. Git unterstützt zwei verschiedene Arten von Tags, kommentierte- und Lightweight-Tags. Im vorherigen Beispiel wurde ein Lightweight-Tag erstellt. Lightweight-Tags und mit Anmerkungen versehene Tags unterscheiden sich in der Menge der von ihnen gespeicherten Metadaten. Es wird empfohlen, mit Anmerkungen versehene Tags als öffentlich und Lightweight-Tags als privat anzusehen. Mit Anmerkungen versehene Tags speichern zusätzliche Metadaten wie den Namen des Taggers, die E-Mail-Adresse und das Datum. Dies sind wichtige Daten für eine Veröffentlichung. Lightweight-Tags sind im Wesentlichen 'Lesezeichen' für einen Commit. Sie sind lediglich ein Name und ein Zeiger auf einen Commit. Sie sind nützlich, um schnelle Links zu relevanten Commits zu erstellen.

## Tags auflisten

Um gespeicherte Tags in einem Repo aufzulisten, führen Sie Folgendes aus:

git tag

Dies wird eine Liste von Tags ausgeben:

v0.10.0

v0.10.0-rc1

v0.11.0

v0.11.0-rc1

v0.11.1

v0.11.2

v0.12.0

v0.12.0-rc1

v0.12.1

v0.12.2

v0.13.0

v0.13.0-rc1

v0.13.0-rc2

## Tags löschen

Das Löschen von Tags ist unkompliziert. Durch Übergeben der –d Option und einer Tag-ID an git tag wird das identifizierte Tag gelöscht.

$ git tag

v1

v2

v3

$ git tag -d v1

$ git tag

v2

v3

In diesem Beispiel wird git tag ausgeführt, um eine Liste der Tags anzuzeigen, die v1, v2, v3 anzeigen. Anschließend wird git tag -d v1 ausgeführt, wodurch das Tag v1 gelöscht wird.