



Automatisiertes Rollout von I4.0 Apps in der Oracle Cloud

öffentlich

Version: 1.0

Status: Final

Autor: Weiß, Ronny

Copyright © 2018 AIS Automation Dresden GmbH. Alle Rechte vorbehalten.

Kein Teil dieser Publikation darf ohne schriftliche Genehmigung der Firma AIS Automation Dresden GmbH vervielfältigt werden.

Alle Warenzeichen oder eingetragenen Warenzeichen von anderen Firmen, die in diesem Dokument genannt werden, dienen nur der Identifikation und sind Eigentum der jeweiligen Herstellerfirmen und werden hiermit anerkannt.

Änderungsnachweis

Beginnend mit der zurzeit bearbeiteten Version des Dokuments enthält die Übersicht Angaben zu allen Änderungen seit der ersten Version.

Vers. Dok.	Freigabe	Autor	Datum	Änderungen
1.0		Ronny Weiß	2018-09-25	Ausgangsversion

Tabelle 1: Änderungsnachweis

Vers. Dok.: Die Version des Dokuments

Freigabe: Freigabedatum der Version des Dokuments – leer, wenn die Version nicht freigegeben ist

Autor: Wenn es mehrere Bearbeiter (siehe Deckblatt) gibt, sind hier diejenigen genannt, die die Änderungen einer Version in das Dokument eingefügt haben. Gibt es nur einen Bearbeiter, kann das Feld frei bleiben.

Datum: Änderungsdatum

Änderung: Kurzbeschreibung aller Änderungen einer Version des Dokuments

Inhalt

1	Schlüsselworte	1
2	Übersicht	2
3	Herausforderung	3
4	Unser Entwicklungsprozess	4
5	Automatisches Rollout alle 2 Wochen	6
6	Fazit und Ausblick.....	8
7	Kontaktadresse	9

1 Schlüsselworte

Industrie 4.0, Oracle Cloud, APEX, TeamFoundationServer, Power Shell

2 Übersicht

Als vor über einem Jahr die Entwicklung unserer EquipmentCloud® begann, sollte das Produkt EquipmentCloud® als Software-as-a-Service stetig weiterentwickelt und verbessert werden. Da wir in unserem Entwicklungsprozess nach Scrum im Zweiwochentakt arbeiten sind die Release Zyklen sehr kurz und es war klar, dass wir automatisierte und sichere Test- sowie einfache, zuverlässige und auditierfähige Rolloutmechanismen benötigen. Doch wie haben wir das ganze umgesetzt und können sicherstellen, dass die Software nach jedem Rollout sauber funktioniert? Genau diese Fragen sollen in diesem Dokument beantwortet werden.



Abbildung 1 - Screenshot EquipmentCloud®

3 Herausforderung

Die kurzen Release Zyklen, das schnelle Wachstum der EquipmentCloud®, sowie die Tatsache das es im produktiven System viele weltweite Endkunden gibt, machten es unumgänglich, dass wir automatisiert testen und unser Rolloutprozess auf professionelle Füße stellen müssen.

Diese Tatsachen führten uns zu folgenden Fragen:

- Wie kann trotz stetigen Änderungen die Funktionalität und Ausfallsicherheit gewährleistet werden?
- Wie kann Sicherheit beim Rollout in Bezug auf die Kundendaten gewährleistet werden?
- Wie erreicht man eine saubere Prozesskontrolle und verhindert das ausrollen nicht getesteter Features?
- Wie findet man eine Balance zwischen Entwicklungsgeschwindigkeit und hoher Verfügbarkeit, bzw. Sicherheit

Diese Fragen werden in den folgenden Abschnitten beantwortet und es wird dargestellt wie es in der EquipmentCloud® aktuell gelöst ist.

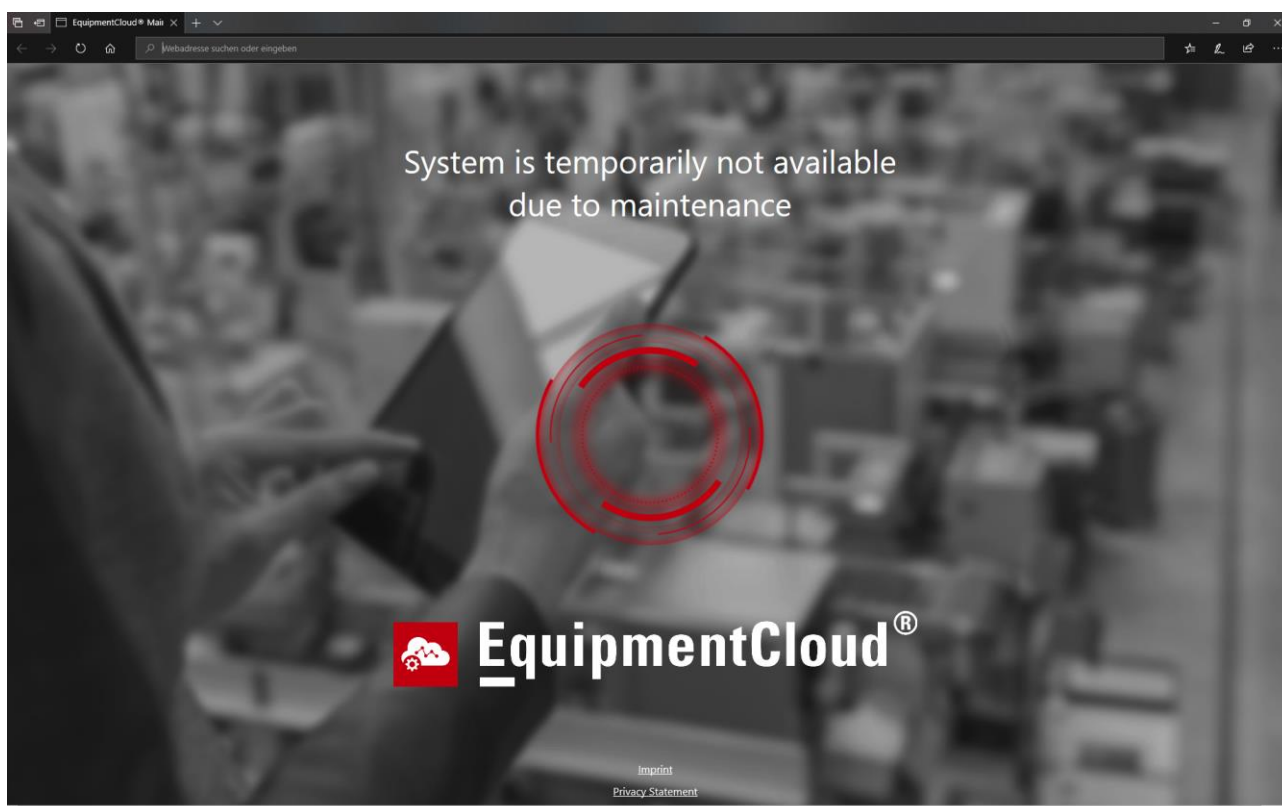


Abbildung 2 – EquipmentCloud® Maintenance Page

4 Unser Entwicklungsprozess

Die gesamte Entwicklung findet Inhouse auf einem sehr produktionsnahen System statt. Dabei arbeiten alle Entwickler im Team auf dem gleichen System.

Nach dem Check-in eines neuen Features laufen automatische Builds. Diese werden geloggt und es finden darin automatische Tests statt, welche sowohl die Funktion als auch die Sicherheit testen. Ein Beispiel dafür sind die Szenario Tests unserer Rest API's. Diese API's werden bewusst konstruierten Angriffen ausgesetzt, um sicherzustellen, dass diese im produktiven System den Anforderungen gewachsen sind und es unmöglich ist ohne Authentifizierung an Daten zu gelangen.

Nach einem Erfolgreichen Build werden die Definitionen auf ein Testsystem ausgerollt. Auf diesem werden dann manuelle Tests durchgeführt.

Nach erfolgreichem Test werden die neuen Features auf ein weiteres System (Main) installiert. Der Unterschied zwischen dem Test- und Mainsystem ist der, dass das Testsystem einer neuen Installation gleicht und das Mainsystem sich wie das produktive System immer inkrementell upgedated. Dadurch wird immer das Update und auch eine Neuinstallation getestet.

Die automatischen und manuellen Tests werden dann erneut auf dem Mainsystem ausgeführt.

Erst dann können die Features überhaupt als Definitionen für einen Produktiven Rollout genutzt werden.

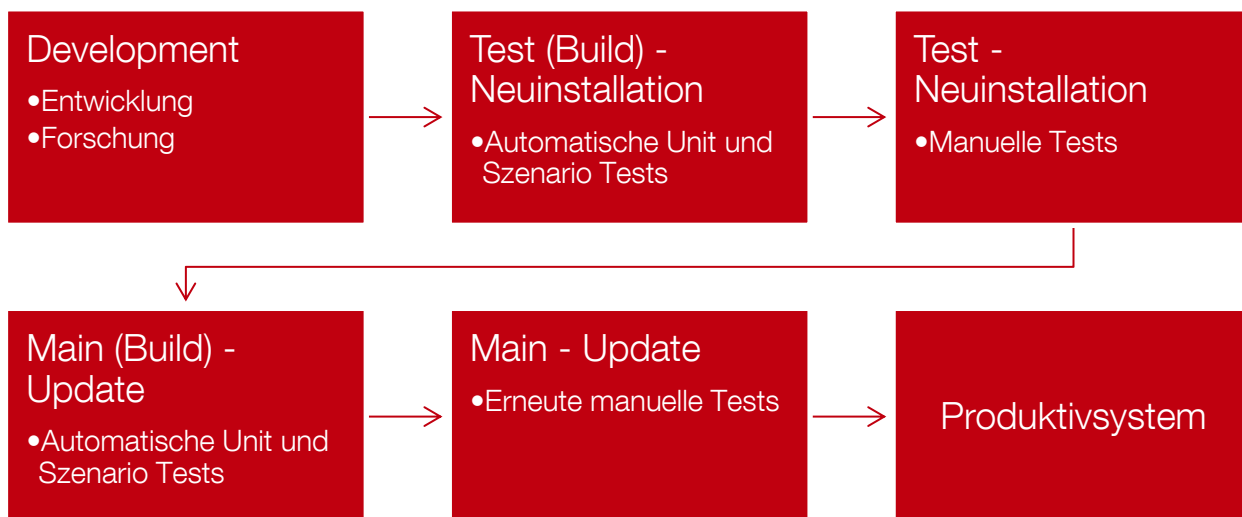


Abbildung 3 – Entwicklungsprozess

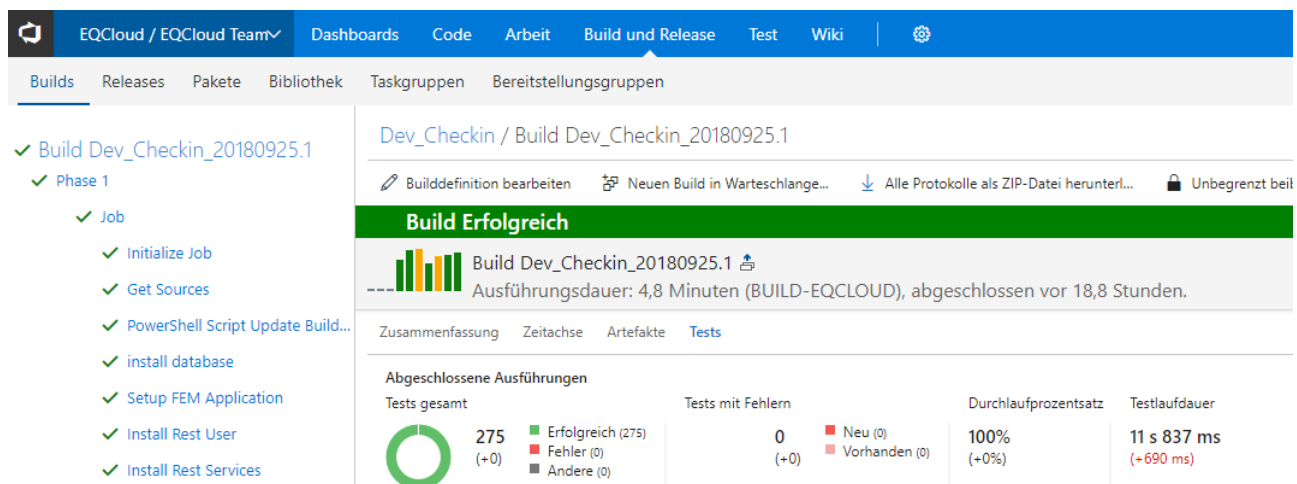


Abbildung 4 - Build Tests

Datei Bearbeiten Search Start Optionen Hilfe

00:00:00 0 0/0

Testplan

- vars
- ResultTexts
- DBREST_Controller_Config
- DBFEM_APEX_Config
- HTTP Request Default Einstellungen
- Upload Default File
- HTTP GET Request Ping
- Hierarchies
- Prepare Rest Hierarchies**

JDBC Request

Name: Prepare Rest Hierarchies

Kommentare:

Variable Name Bound to Pool

Variable Name: FEM_APEX

SQL Query

Query Type: Callable Statement

Query:

```

1 DECLARE
2   VR_APP_RIGHT  BOOLEAN;
3   VR_RES        NUMBER;
4   VR_USER       VARCHAR2( 255 ) := '${v_User}';
5   VR_Pw         VARCHAR2( 255 ) := '${v_Password}';
6   VR_WORLD      NUMBER;
7   VR_UNITESTL1  NUMBER;
8   VR_UNITESTL2A NUMBER;
9   VR_DELETE     NUMBER;
10 BEGIN
11   VR_RES := :NT.IS_LOGIN_VALID(
12     VR_USER,
13     VR_PW
14   );
15   -- check user & rights
16   IF
17     VR_RES = 0
18   THEN
19     RAISE_APPLICATION_ERROR(
20       -20001,
21       'user ' ||
22       VR_USER ||
23       ' not configured'
24     );
25   END IF;
26   VR_APP_RIGHT := :APPLICATION_RIGHTS(
27     VR_USER,
28     'ALTER_HIERARCHY'
29   );
30   IF
31     NOT VR_APP_RIGHT
32   THEN
33     RAISE_APPLICATION_ERROR(
34

```

Parameter values:

Parameter types:

Variable names:

Result variable name:

Query timeout (s):

Handle ResultSet: Store as String

Abbildung 5 - Szenario Test for Rest API in JMeter

5 Automatisches Rollout alle 2 Wochen

Nach jedem Sprint werden sämtliche Kundensysteme aktualisiert. Hierfür gibt es für interessierte Kunden einen Review, in denen alle neuen Features vorgestellt werden. Anschließend wird das Release für das produktive System erstellt und kann nur durch Freigabe einer weiteren befugten Person ausgerollt werden.

Zunächst wird lediglich ein Testcontainer aktualisiert. Auf diesem wird erneut ein Smoketest ausgeführt. Ist dieser Test erfolgreich, so wird zunächst der Templatecontainer, welcher für das Anlegen von Neukunden dient, und zum Schluss die Kundencontainer aktualisiert. Das Update selbst erfolgt dabei durch Releasedefinitionen im TeamFoundationServer. Dieser triggert sogenannte Taskgruppen, welche die eigentlichen Power Shell, SQL und DPC-Skripte enthalten.

Der Vorteil beim Rollout über die TFS-Definitionen ist, dass man die Last (Anzahl der Parallelen Updates) genau steuern kann und durch Parameter, sowie Password Verschlüsselung in der Lage ist neben einer Konfiguration des Updates auch die Sicherheit zu garantieren, sodass die Skripte keine Passwörter oder andere vertrauliche Informationen enthalten.

Die Parametrisierung wird genutzt um die zu installierbaren Apps wahlweise auszurollen und zu konfigurieren. Dabei nutzen wir innerhalb der SQL-Skripte JSON, um einen fehlerfreien Ablauf auch über viele Skriptdateien hinweg zu gewährleisten.

Der DPC ist ein Produkt der AIS, welches dazu dient versionsunabhängig Microsoft SQL und Oracle Datenbanken zu aktualisieren.

Beim Update des 1. Containers wird dabei durch den DPC die eigentliche DDL erzeugt, umso schneller die Kundensysteme versionsunabhängig auf den neusten Stand zu bekommen.

Geschwindigkeit während des Updates spielt eine sehr große Rolle um die Verfügbarkeit unseres Systems zu maximieren.

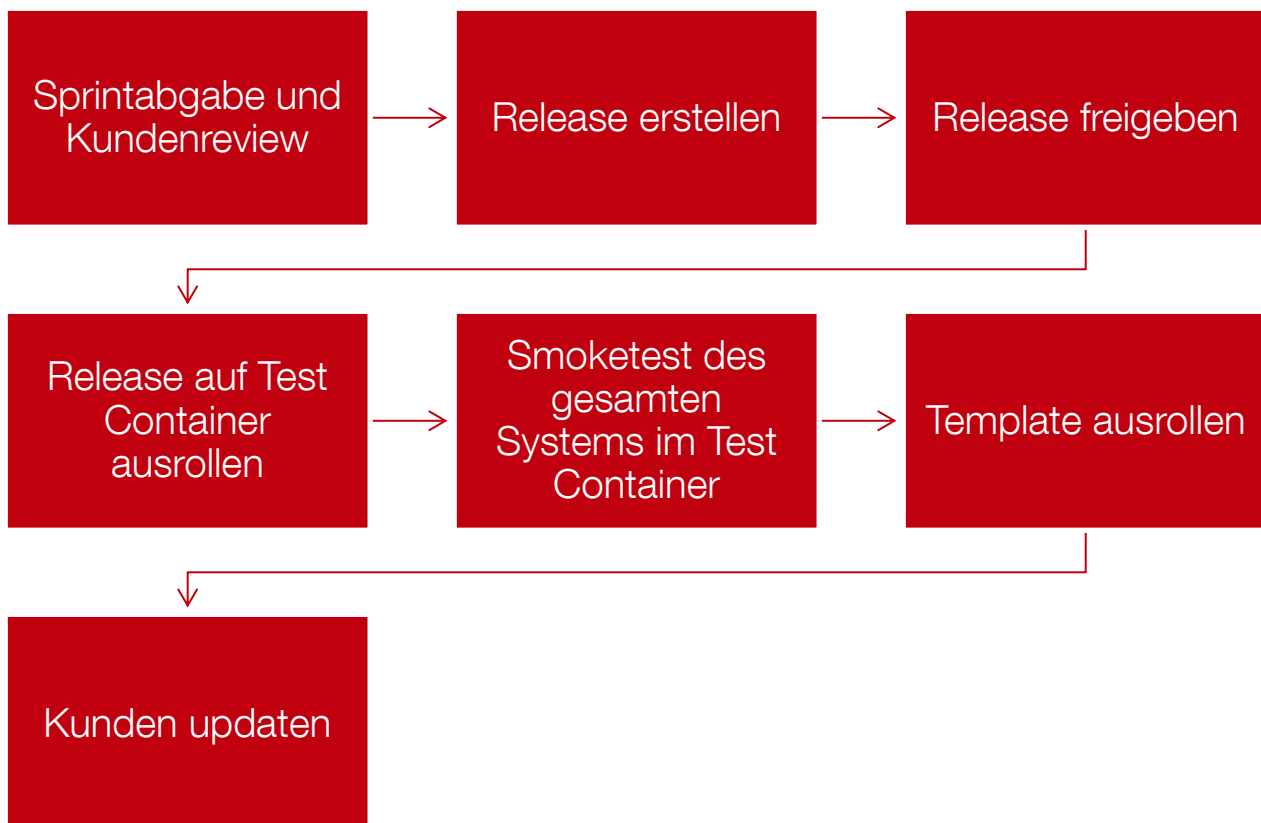
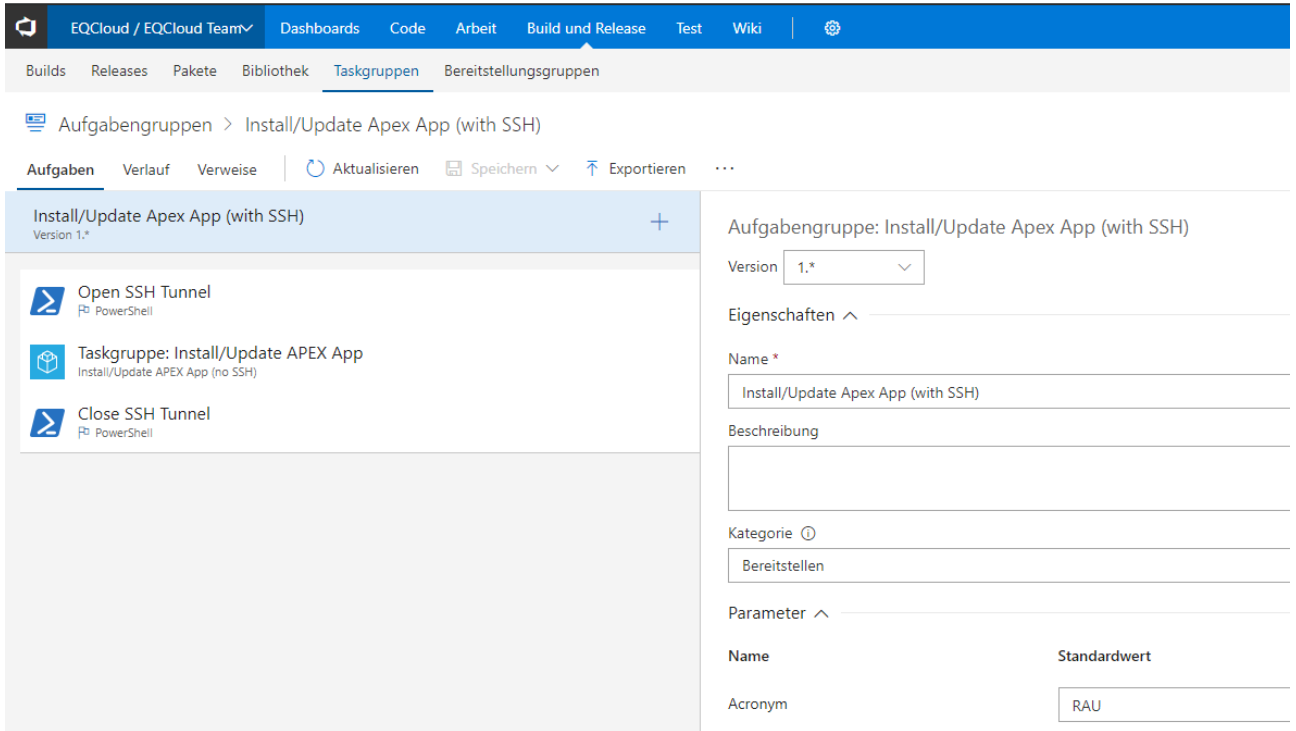


Abbildung 6 – Rolloutprozess



EQCloud / EQCloud Team | Dashboards | Code | Arbeit | **Build und Release** | Test | Wiki

Builds | Releases | Pakete | Bibliothek | **Taskgruppen** | Bereitstellungsgruppen

Aufgabengruppen > Install/Update Apex App (with SSH)

Aufgaben | Verlauf | Verweise | Aktualisieren | Speichern | Exportieren

Install/Update Apex App (with SSH)
Version 1.*

- Open SSH Tunnel (PowerShell)
- Taskgruppe: Install/Update APEX App (Install/Update APEX App (no SSH))
- Close SSH Tunnel (PowerShell)

Aufgabengruppe: Install/Update Apex App (with SSH)

Version: 1.*

Eigenschaften

Name: Install/Update Apex App (with SSH)

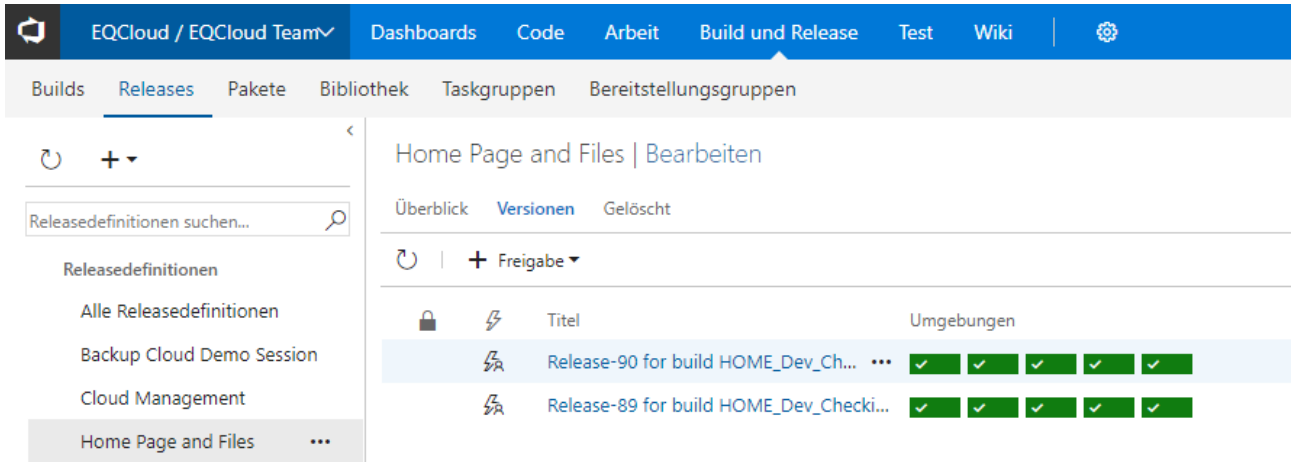
Beschreibung:

Kategorie: Bereitstellen

Parameter

Name	Standardwert
Acronym	RAU

Abbildung 7 - Release Definition



EQCloud / EQCloud Team | Dashboards | Code | Arbeit | **Build und Release** | Test | Wiki

Builds | Releases | Pakete | Bibliothek | **Taskgruppen** | Bereitstellungsgruppen

Releasedefinitionen suchen...

Releasedefinitionen

- Alle Releasedefinitionen
- Backup Cloud Demo Session
- Cloud Management
- Home Page and Files

Home Page and Files | Bearbeiten

Überblick | **Versionen** | Gelöscht

+ Freigabe

Titel	Umgebungen
Release-90 for build HOME_Dev_Ch...	✓ ✓ ✓ ✓ ✓
Release-89 for build HOME_Dev_Checki...	✓ ✓ ✓ ✓ ✓

Abbildung 8 - Release

6 Fazit und Ausblick

Unsere Test- und Rolloutmechanismen führten dazu, dass wir bisher keinerlei Ausfälle unserer EquipmentCloud® nach den Updates hatten und das die Funktionalität der Cloud stetig gewährleistet ist. Deshalb konnten wir unseren Entwicklungs- und Rolloutprozess auch dieses Jahr erfolgreich durch ISO 27001 zertifizieren lassen.

Zukünftig geht es vor allem darum bei den Updates noch schneller zu werden. Dafür setzen wir auf neue Technologien wie Applikationscontainer in der Oracle Datenbank.

7 Kontaktadresse

Ronny Weiß

AIS Automation Dresden GmbH

Otto-Mohr-Straße 6

D-01237 Dresden

Telefon: +49 (0) 351-2166 1388

E-Mail: <mailto:Ronny.Weiß@ais-automation.com>

Internet: <http://www.ais-automation.com>

EquipmentCloud®: <https://eqcloud.ais-automation.com>

