

Windows Patchday

KIS Phoenix - JBoss Patch

24.11.2023

Management Summary

blabla

Inhaltsverzeichnis

Abkürzungen	3
1 Einführung	1
2 Generell	4
3 Testsystem	6
4 Produktivsystem	7
Abbildungsverzeichnis	8
Tabellenverzeichnis	9
Literatur	10
Glossar	11
Anhang	12
0.1 #811534244# - KIS PROD Down - SKS0021 Deffekt	13

Abkürzungen

ICT	information and communications technology
KSGR	Kantonsspital Graubünden
KIS	Klinisches Informationssystem
JBoss	JavaBean Open Source Software Application Sever
JVM	Java Virtual Machine

1 Einführung

Im Rahmen des von Microsoft indizierten Patchdays, bei dem Microsoft alle 2 Monate einen Patch veröffentlicht, müssen auch die JBoss-Server des KIS Phoenix angefasst werden. Die Server werden dabei von Ivanti mit den Patches betankt, so das man nur noch einen Reboot durchführen und nicht noch erst die Patches herunterladen muss..

Das KIS Phoenix verfügt dabei über 4 JBoss-Server, 2 für die Testsysteme resp. das Demosystem und 2 für das produktive System.

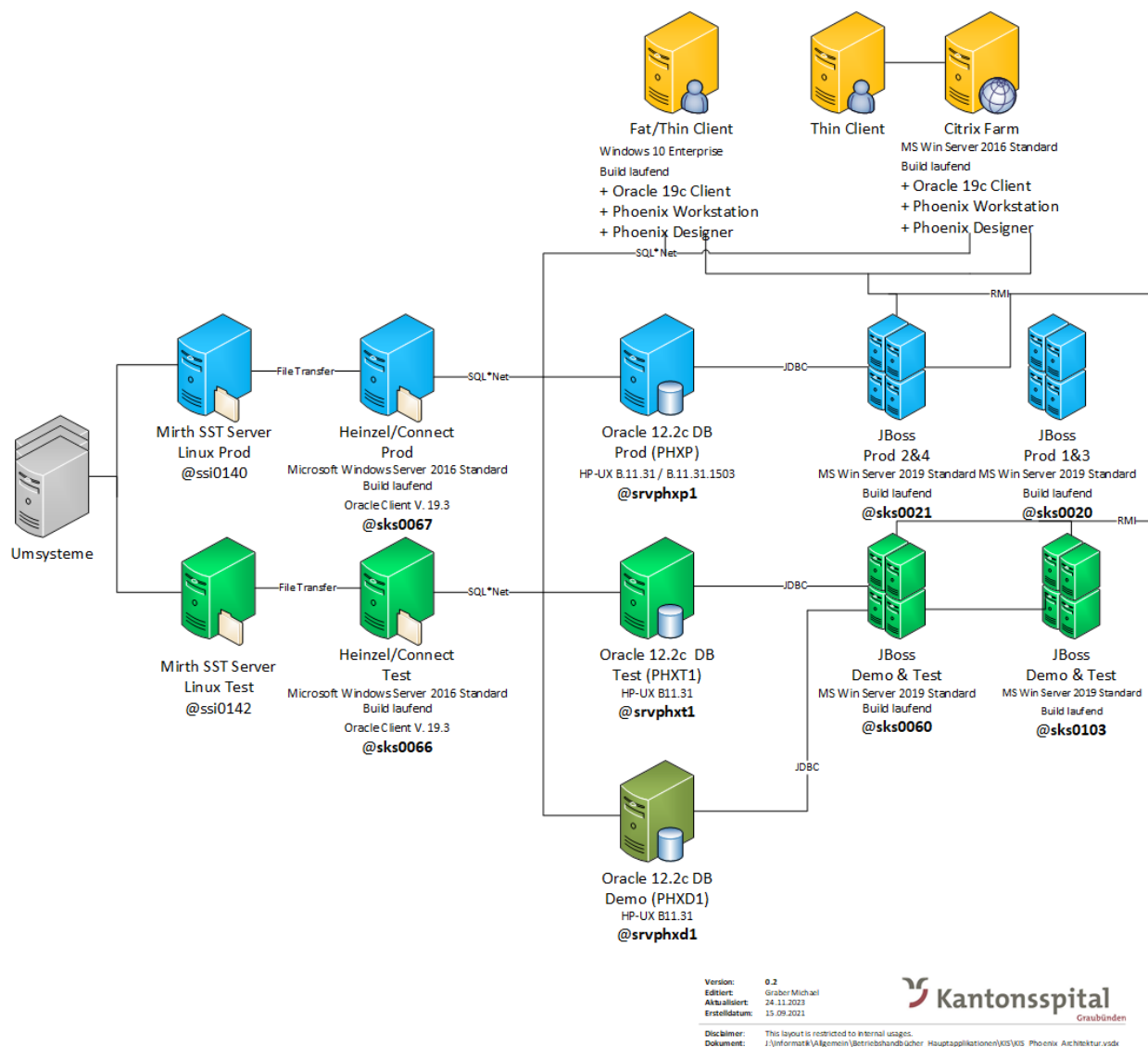


Abbildung 1.1: Architektur KIS Phoenix[4]

Das Test1 und Demosystem hat je einen JBoss-Node auf einer Seite, das Test2 hat nur einen Node auf einem Server. Das Produktivsystem hat je Site zwei Nodes, also insgesamt vier Nodes.

Die Verzeichnisstruktur sieht dabei wie folgt aus:

```
/
├── laufwerk
│   ├── phoenix-server-<system>-<version>
│   │   ├── <node>
│   │   │   ├── standalone
│   │   │   │   ├── log
│   │   │   │   │   └── server.log
```

Meistens existieren pro Node die Verzeichnisse mehrerer Versionen, üblicherweise immer die der aktuellen und vorangehenden Version.

Daher aufpassen in welchem Verzeichnis man ist!

Auf jedem Server wurde zudem *baretail.exe* installiert, ein Logging-Tool welches eine Echtzeitüberwachung der Logs ermöglicht. Leider ist es nicht auf jedem Server gleich installiert sondern individuell.

Die genaue Konfiguration sieht wie folgt aus:

Typ	Applikationsserver										Schnittstellenserver	
Systeme	Produktion					Test- und Demosystem Test- und Demosystem					Produktion	Test
Rolle	Produktion					Test- und Demosystem Test- und Demosystem					Produktion	Test
Physikalischer Hostname	sks0020					sks0060					sks0103	sks0067
IP Adresse phy. Host	10.0.22.43					10.0.22.52					10.0.22.15	10.0.22.54
IP Submask						255.255.255.0						10.0.22.53
Gateway						10.0.22.1						
DNS 1						10.0.16.163						
DNS 2						10.0.16.163						
Timeserver						10.10.146.196						
JBoss Nodes	prod1	prod3	prod2	prod4	demo11	test11	test21	demo12	test12		-	
Windows Services	PhoenixJBossEAP_prod_1	PhoenixJBossEAP_prod_3	PhoenixJBossEAP_prod_2	PhoenixJBossEAP_prod_4	PhoenixJBossEAP_demo1_1	PhoenixJBossEAP_test1_1	PhoenixJBossEAP_test2_1	PhoenixJBossEAP_demo1_2	PhoenixJBossEAP_test1_2		-	

Tabelle 1.1: Spezifikationen KIS Phoenix Applikations- und Schnittstellenserver

2 Generell

Grundsätzlich verläuft der Patch auf den Test- und Produktivservern gleich ab. Die Services auf dem zweiten Server dürfen erst gestoppt werden, wenn der JBoss-Service auf dem ersten Server vollständig einsatzbereit ist und verbindungen annimmt.

Es gibt leider keine Garantie dafür, das die JBoss-Services laufen, wenn der Windows-Service läuft. Um zu Prüfen, ob der JBoss-Service an sich lauffähig ist, gibt es mehrere Indikatoren.

Zum einen gibt es im standalone-Verzeichnis das Subverzeichnis deployments in welchem gewisse Files auskunft über die Gesundheit des Nodes geben. Diese sind wie folgt zu finden:

```
/
├─ laufwerk
│   └─ phoenix-server-<system>-<version>
│       └─ <node>
│           └─ standalone
│               └─ deployments
```

Bei einem geglückten Deployment eines Nodes muss ein *phoenix.ear*- und *phoenix.ear.deployed*-File vorhanden sein.

Bei einem Fehler wiederum wird i.d.R. *phoenix.ear.failed*-File erzeugt.




 phoenix.ear	15.09.2021 02:59	EAR-Datei	1 KB
 phoenix.ear.deployed	15.09.2021 02:59	DEPLOYED-Datei	1 KB
 phoenix.ear.failed	15.09.2021 02:59	FAILED-Datei	1 KB

Abbildung 2.1: Deployment-Status

Das entsprechende Mail ist im Anhang in folgendem Kapitel zu finden: [Abschnitt 0.1](#)

Allerdings ist dies nicht immer 100% verlässlich. Die sicherste Methode, um die Funktionsfähigkeit eines Nodes resp. Servers zu testen besteht darin, die *Workstation.exe* nur noch auf den entsprechenden Node resp. Server verbinden zu lassen. Die Verbindung wird ins *Workstation.ini* geschrieben, welches wie üblich hier zu finden ist:

```
/
├─ C:\
│   └─ Program Files (x86)
│       └─ <Phoenix7-Verzeichnis>
│           └─ Workstation.ini
```


Im Ini stehen die JBoss-Server und Nodes im Eintrag *-Dphoenix.server.nodes*. Hier das Beispiel des Produktiven Ini-Files:

```
1  -clean
2  -data
3  @noDefault
4  -configuration c:/temp/phoenix/prod_7_23_1/eclipse/configuration
5  -nl
6  de_CH
7  -vmargs
8  -Dorg.eclipse.update.reconcile=false
9  -Dphoenix.db=Phoenix
10 -Dphoenix.application.workspace.createdir=true
11 -Dphoenix.application.workspace=\\\\phoenix\\workspace\\prod_7_23_1
12 -Dphoenix.login.usernamefile=%APPDATA%/Phoenix/login_username_prod
13 -Dphoenix.server.nodes=sks0020:8443,sks0021:8443,sks0020:8543,sks0021:8543
14 -Djavax.net.ssl.trustStore=C:\\Program Files (x86)\\Phoenix7\\truststore.jks
15 -Dphoenix.login.enableLDAP=true
16 -Dphoenix.application.winlogin=true
17 -XX:-CreateMinidumpOnCrash
18
```

Listing 2.1: Workstation.ini PROD Beispiel

Am Einfachsten ist, man erstellt sich drei Ini-Files, eines für beide Server zusammen im Cluster und eines pro Server. Dann braucht man nur die Files umzubennen und kann rasch Testen.

Vorsicht!

Es kann vorkommen, dass jeder Server für sich alleine Lauffähig ist aber im Cluster-Verbund keine Anmeldung möglich ist!

Daher sollte nach einem Reboot nebst der Lauffähigkeit der einzelnen Nodes auch immer der Cluster getestet werden!

3 Testsystem

Beim

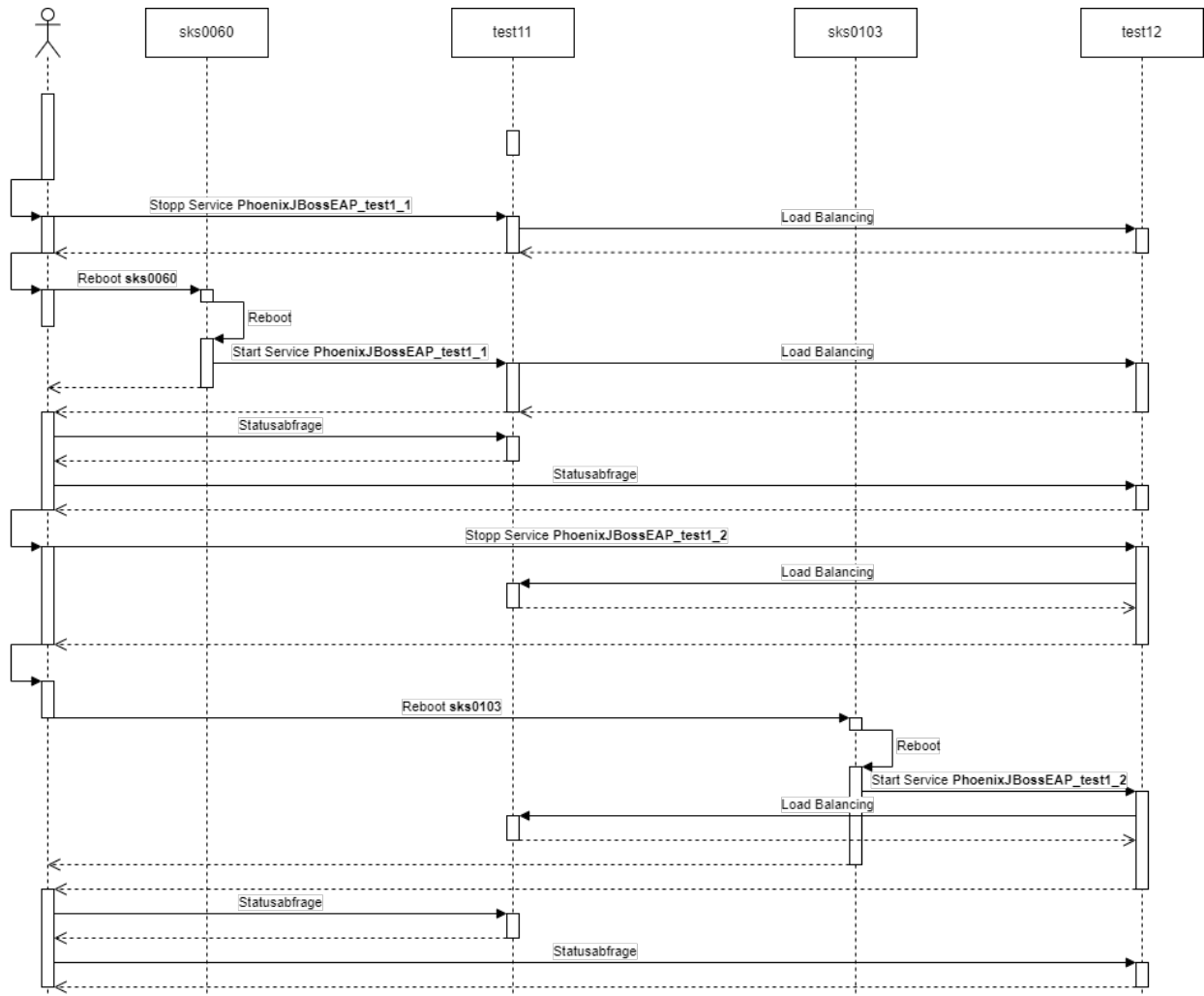


Abbildung 3.1: TEST-Sequenzdiagramm

4 Produktivsystem

Beim Produktivsystem

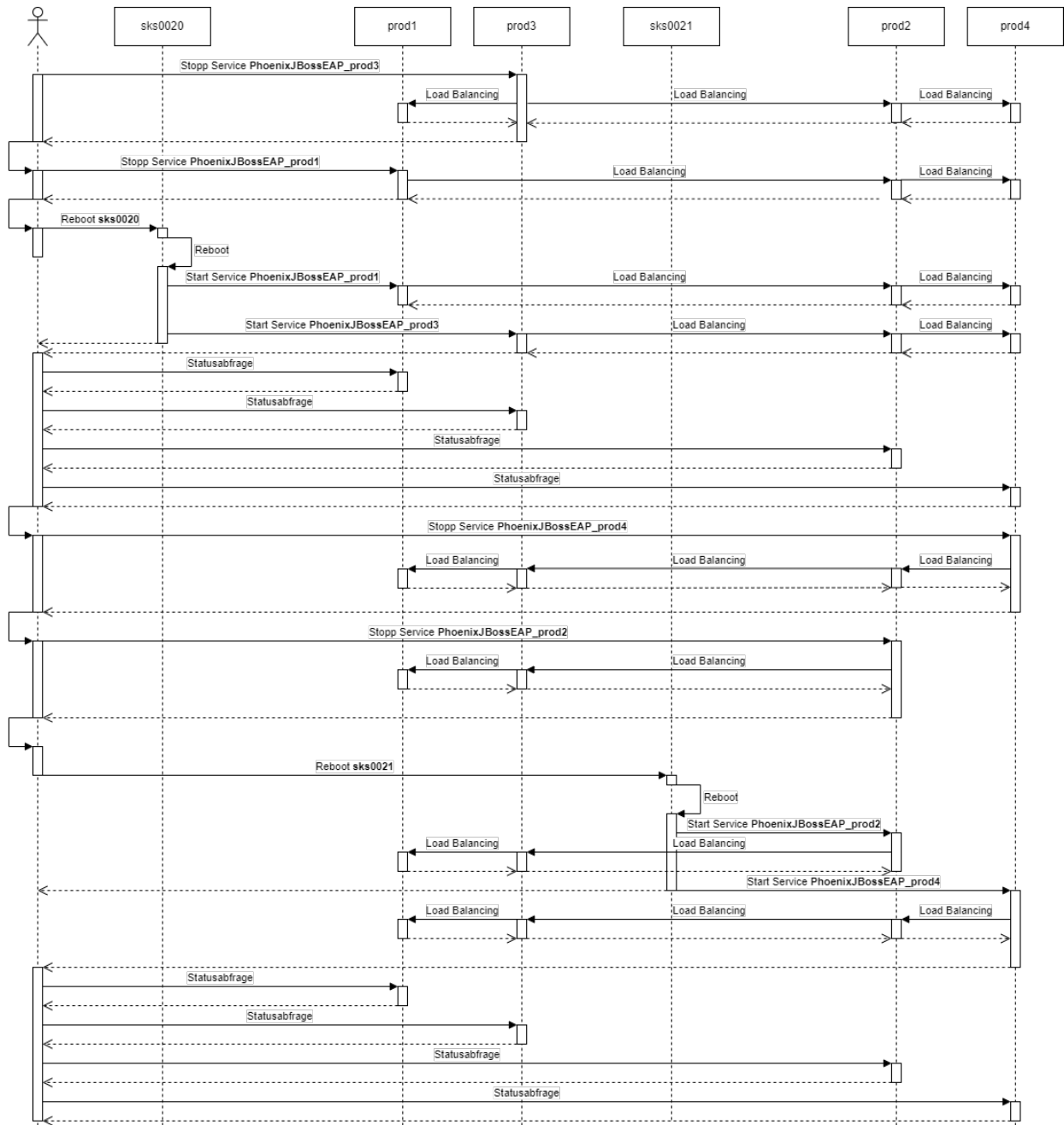


Abbildung 4.1: PROD-Sequenzdiagramm

Abbildungsverzeichnis

1.1	Architektur KIS Phoenix[4]	1
2.1	Deployment-Status	4
3.1	TEST-Sequenzdiagramm	6
4.1	PROD-Sequenzdiagramm	7
.1	#811534244# - KIS PROD Down - SKS0021 Defekt	13

Tabellenverzeichnis

1.1	Spezifikationen KIS Phoenix Applikations- und Schnittstellenserver	3
-----	--	---

Literatur

- [1] Stephan Augsten. *Was ist JBoss bzw. WildFly?* <https://www.dev-insider.de/was-ist-jboss-bzw-wildfly/> 2018.
- [2] Roger Cavegn und KSGR. *Ivanti_AppSense_AM.pdf*. [https://webbi.dokuwiki/doku.php?id=know-how:ivanti_application_control&s\[\]=ivanti](https://webbi.dokuwiki/doku.php?id=know-how:ivanti_application_control&s[]=ivanti).
- [3] Oracle Corporation. *Was ist Java, und wozu brauche ich es?* https://www.java.com/de/download/help/whatis_java.html. 2023.
- [4] Michael Graber u. a. *KIS-Betriebshandbuch_V0_25.docx*. 2021.
- [5] Inc Ivanti. *AppSense - Ivanti*. <https://www.ivanti.com/de/company/history/appsense>.

Glossar

Ivanti Das KSGR nutzt die Ivanti AppSense[5, 2] für die Verteilung von Software und dem Installieren von Patches.. 1

Java Java ist eine Hochsprache die 1995 von Sun Microsystems veröffentlicht wurde. Dank der JVM läuft Java auf sehr vielen Plattformen[3].. 3, 11

JBoss JBoss ist ein Applikationsserver für Java Anwendungen, die von Red Hat aufgekauft wurde. JBoss gilt als Standardserver für Java EE Anwendungen[1].. 3, 1, 4, 5

0.1 #811534244# - KIS PROD Down - SKS0021 Defekt

Graber Michael

Von: Eschler, Ramon <ramon.eschler@cgm.com>
Gesendet: Mittwoch, 15. September 2021 15:26
An: Graber Michael; SUPPORT@HIS-CH.CGM.COM
Cc: Bechmann Susann; Braunschweiler Patrick; Derungs Andreas; Donnicola Michele; Molenaar Pieter; Stein Matthias
Betreff: #811534244# - KIS PROD Down - SKS0021 Defekt

Hallo Michael

Nach eurem Anruf habe ich im auf dem SKS0021 bemerkt das beide Nodes nicht korrekt gestartet wurden. Ob dies mit dem zurücksetzen zu tun hat kann ich leider nicht beurteilen. Ebenfalls schrieb der Node1 im Serverlog das die Clusterview nicht korrekt ist, was vermuten lässt das er die beiden Nodes zwar gesehen hat aber nicht mit ihnen kommunizieren konnte. Dadurch hing der Node1 wohl ebenfalls fest weil er sich versuchte mit den beiden Nodes auf dem SKS0021 zu verbinden.

Um das Problem zu beheben habe ich alle Nodes heruntergefahren und danach einen nach dem anderen wieder hochgefahren. Dabei habe ich zuerst Node 1&3 auf dem SKS0020 neu gestartet. Anschliessend Node 2&4 auf dem SKS0021.

Im Normalfall habt ihr im Verzeichnis vom JBoss: \standalone\deployments Files vorhanden welche einen gewissen Status haben müssen. So ist es wichtig das die phoenix.ear ein File mit dem Namen phoenix.ear.deployed hat, ansonsten kannst du davon ausgehen das der Start noch nicht erfolgt ist oder nicht geklappt hat. Während der JBoss startet müsste das File phoenix.ear.doDeploy oder ähnlich heissen. Heisst das File zu einem Zeitpunkt phoenix.ear.failed - ist der Start missglückt. Hier würde ich in erster Linie versuchen den Service neu zu starten und die "tmp" & "data" Verzeichnisse im \standalone\ Ordner zu löschen. Sollte danach der JBoss nicht hochkommen müsstet ihr den Support kontaktieren.

phoenix.ear	15.09.2021 02:59	EAR-Datei	1 KB
phoenix.ear.deployed	15.09.2021 02:59	DEPLOYED-Datei	1 KB
phoenix.ear.failed	15.09.2021 02:59	FAILED-Datei	1 KB

Liebe Grüsse
Ramon

Ramon Eschler
Application Supporter / Service-Desk

Find a better way – every day



CompuGroup Medical Schweiz AG
Gartenstrasse 9, CH-3007 Bern
Telefon: +41 (31) 924 21 77
ramon.eschler@cgm.com / www.cgm.com/ch

SYNCHRONIZING HEALTHCARE

Before printing, think about the environment.

